Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems

Nan Zhang*, Xianghang Mi*, Xuan Feng†*, XiaoFeng Wang*, Yuan Tian‡ and Feng Qian*
*Indiana University, Bloomington
Email: {nz3, xmi, xw7, fengqian}@indiana.edu
†Beijing Key Laboratory of IOT Information Security Technology, Institute of Information Engineering, CAS, China
Email: fengxuan@iie.ac.cn
‡University of Virginia
Email: yuant@virginia.edu

***Abstract*—Virtual personal assistants (VPA) (e.g., Amazon Alexa and Google Assistant) today mostly rely on the voice channel to communicate with their users, which however is known to be vulnerable, lacking proper authentication (from the user to the VPA). A new authentication challenge, from the VPA service to the user, has emerged with the rapid growth of the VPA ecosystem, which allows a third party to publish a function (called *skill*) for the service and therefore can be exploited to spread malicious skills to a large audience during their interactions with smart speakers like Amazon Echo and Google Home. In this paper, we report a study that concludes such remote, large-scale attacks are indeed realistic. We discovered two new attacks: *voice squatting* in which the adversary exploits the way a skill is invoked (e.g., "open capital one"), using a malicious skill with a similarly pronounced name (e.g., "capital won") or a paraphrased name (e.g., "capital one please") to hijack the voice command meant for a legitimate skill (e.g., "capital one"), and *voice masquerading* in which a malicious skill impersonates the VPA service or a legitimate skill during the user's conversation with the service to steal her personal information. These attacks aim at the way VPAs work or the user's misconceptions about their functionalities, and are found to pose a realistic threat by our experiments (including user studies and real-world deployments) on Amazon Echo and Google Home. The significance of our findings has already been acknowledged by Amazon and Google, and further evidenced by the risky skills found on Alexa and Google markets by the new squatting detector we built. We further developed a technique that automatically captures an ongoing masquerading attack and demonstrated its efficacy.**

I. Introduction

The wave of Internet of Things (IoT) has brought in a new type of *virtual personal assistant* (VPA) services. Such a service is typically delivered through a smart speaker that interacts with the user using a voice user interface (VUI), allowing the user to command the system with voice only: for example, one can say "what will the weather be like tomorrow?" "set an alarm for 7 am tomorrow", etc., to get the answer or execute corresponding tasks on the system. In addition to their built-in functionalities, VPA services are enhanced by *ecosystems* fostered by their providers, such as Amazon and Google, under which third-party developers can build new functions (called *skills* by Amazon and *actions* by Google[1]) to offer further helps to the end users, for example, order food, manage bank accounts and text friends. In the past year, these ecosystems are expanding at a breathtaking pace: Amazon claims that already 25,000 skills have been uploaded to its skill market to support its VPA (including the *Alexa* service running through Amazon Echo) [1] and Google also has more than one thousand actions available on its market for its Google Home system (powered by *Google Assistant*). These systems have already been deployed to the households around the world, and utilized by tens of millions of users. This quickly-gained popularity, however, could bring in new security and privacy risks, whose implications have not been adequately understood so far.

Security risks in VPA voice control. Today's VPA systems are designed to be primarily commanded by voice. Protecting such VUIs is fundamentally challenging, due to the lack of effective means to authenticate the parties involved across the open and noisy voice channel. Already prior research shows that the adversary can generate obfuscated voice commands [14] or even completely inaudible ultrasound [49] to attack speech recognition systems. These attacks impersonate the authorized user to the voice-controlled system, since no protection is in place to authenticate the user to the system.

The emergence of the VPA ecosystem brings in another authentication challenge: it also becomes difficult for the user to determine whether she is indeed talking to the right skill and the VPA itself as she expects. The problem comes from the fact that through the skill market, an adversary can publish malicious third-party skills designed to be invoked by the user's voice commands (through a VPA device such as Amazon Echo or Google Home) in a misleading way, due to the ambiguity of the voice commands and the user's misconception about the service. As a result, the adversary could *impersonate a legitimate skill or even the VPA (potentially in a large scale) to the user*. This attack is made possible by the absence of

[1]Throughout the paper, we use the Amazon term *skill* to describe third-party applications, including Google's actions.

authentication from the VPA to the user over the voice channel, a risk that our research shows leads to a realistic threat.

Voice-based remote attacks. In our research, we analyzed the most popular VPA IoT systems – Alexa and Google Assistant, focusing on the third-party skills deployed to these devices. Our study demonstrates that through publishing malicious skills, it is completely feasible for an adversary to remotely attack the users of these popular systems, collecting their private information through their conversations with the systems. More specifically, we identified two threats *never known before*, called *voice squatting attack* (VSA) and *voice masquerading attack* (VMA). In a VSA, the adversary exploits how a skill is invoked (by a voice command), and the variations in the ways the command is spoken (e.g., phonetic differences caused by accent, courteous expression, etc.) to cause a VPA system to trigger a malicious skill instead of the one the user intends (Section III-B). For example, one may say "Alexa, open Capital One please", which normally opens the skill *Capital One*, but can trigger a malicious skill *Capital One Please* once it is uploaded to the skill market. A VMA aims at the interactions between the user and the VPA system, which is designed to hand over all voice commands to the currently running skill, including those supposed to be processed by VPA system like terminating the current skill and switching to a new one. In response to the commands, a malicious skill can pretend to yield control to another skill (switch) or the service (terminate), yet continue to operate stealthily to impersonate these targets and get sensitive information from the user (Section III-C).

We further investigated the feasibility of these attacks through user studies, system analysis, and real-world exploits. More specifically, we first surveyed 156 Amazon Echo and Google Home users and found that most of them tend to use natural languages with diverse expressions to interact with the devices: e.g., "play some sleep sounds". These expressions allow the adversary to mislead the service and launch a wrong skill in response to the user's voice command, such as *some sleep sounds* instead of *sleep sounds*. Our further analysis of both Alexa and Google Assistant demonstrates that indeed these systems identify the skill to invoke by looking for the longest string matched from a voice command (Section III-B). Also interestingly, our evaluation of both devices reveals that Alexa and Google Assistant cannot accurately recognize some skills' invocation names and the malicious skills carrying similar names (in terms of pronunciation) are capable of hijacking the brands of these vulnerable skills.

Finally, we deployed four skills through the Amazon market to attack a popular Alexa skill "Sleep and Relaxation Sounds" [8]. These skills have been invoked by 2,699 users in a month and collected 21,308 voice commands in plain text. We built the skills in a way to avoid collecting private information of the real-world users. Still, the commands received provide strong evidence that indeed both voice squatting and masquerading can happen in real life: our study shows that the received commands include the ones only eligible for "Sleep and Relaxation Sounds", and those for switching to a different skill or terminating the current skill that can be leveraged to impersonate a different skill (Section III-D). Our analysis of existing skills susceptible to the threat further indicates the significant consequences of the attacks, including disclosure of one's home address, financial data, etc. The video demos of these attacks are available online [7].

Responsible disclosure. We have reported our findings to Amazon and Google in February 2018, both of which acknowledged that the problems we discovered are new and serious. From February to May, we had multiple meetings with both vendors to help them understand and mitigate such new security risks.

Ethical issues. All human subject studies reported by the paper (III-A, III-B, and III-D) have been approved by our IRB. All the skills we published did not collect any private, identifiable information and only provided legitimate functionalities similar to "Sleep and Relaxation Sounds". The user requests our skills received from Amazon and Google were in plain text, which did not contain voice biometric information. Any private, identifiable information sent to our skills by mistake were removed immediately. We have stated that user de-identified data will be used in the research and provided the institution, IRB protocol, and contact information in the privacy policies of the skills we published. Although the skills could launch VMAs e.g. faking in-communication skill switch and termination, they were designed not to do so. Instead, we just verified that such attack opportunities indeed exist.

Mitigation. In our research, we developed a suite of new techniques to mitigate the realistic threats posed by VSA and VMA. We built a *skill-name scanner* that converts the invocation name string of a skill into a phonetic expression specified by ARPABET [5]. This expression describes how a name is pronounced, allowing us to measure the phonetic distance between different skill names. Those sounding similar or having a subset relation are automatically detected by the scanner. This technique can be used to vet the skills uploaded to a market. Interestingly, when we ran it against all 19,670 custom skills on the Amazon market, we discovered 4,718 skills with squatting risks (Section IV-C): e.g., a skill with an invocation name "me a dog fact" looks suspiciously related to the popular skill "dog fact". Our findings indicate that possibly these attacks could already happen in the wild.

To mitigate the threat of the masquerading attack, we need to identify a running skill's utterances that are supposed to come from the VPA service, to prevent the skill from impersonating the system, and the user's voice commands meant for the system, to avoid the skill's attempts to hijack such commands (e.g., switching to a different skill). To this end, we developed a novel technique that automatically identifies those similar to system utterances, even in the presence of obfuscation attempts (e.g., changes to the wording), and also captures the user's skill switching intention from the context of her conversation with the running skill. Our technique leverages natural-language processing (NLP) and machine learning (Section V) and was found in our experiments to be highly effective (over 95% precision) in defeating an ongoing VMA attack.

Contributions. The contributions of the paper are outlined as follows:

- *First study on the security risks of malicious skills*. We report the first security analysis on the ecosystems of leading VPA services for IoT systems (Amazon Echo and Google Home), which leads to the discovery of serious security weaknesses in their VUIs and skill vetting that enable the remote attacks from the skills uploaded by untrusted third parties. We present two new attacks, voice squatting and voice masquerading. Both are demonstrated to pose realistic threats to a large number of VPA users and both have serious security and privacy implications. Our preliminary analysis of the Amazon skill market further indicates that similar attacks may already happen in the wild.
- *New techniques for risk mitigation*. We made the first step towards protecting VPA users from such malicious skill attacks. We show that the new protection can mitigate the threats in realistic environments. The idea behind our techniques, such as context-sensitive command analysis, could inspire further enhancement of the current designs to better protect VPA users.

II. Background

A. Virtual Personal Assistant Systems

VPA on IoT devices. Amazon and Google are two major players in the market of smart speakers with voice-controlled personal assistant capabilities. Since the debut of the first *Amazon Echo* in 2015, Amazon has now taken 76% of the U.S. market with an estimate of 15-million devices sold in the U.S. alone in 2017 [3]. In the meantime, Google has made public *Google Home* in 2016, and now grabbed the remaining 24% market share. *Amazon Echo Dot* and *Google Home Mini* are later released in 2016 and 2017, respectively, as small, affordable alternatives to their more expensive counterparts. Additionally, Amazon has integrated VPA into IoT products from other vendors, e.g. Sonos smart speaker, Ecobee thermostat [2].

A unique property of these four devices is that they all forgo conventional I/O interfaces, such as the touchscreen, and also have fewer buttons (to adjust volume or mute), which serves to offer the user a hands-free experience. In another word, one is supposed to command the device mostly by speaking to it. For this purpose, the device is equipped with a microphone circular array designed for 360-degree audio pickup and other technologies like beamforming that enable far-field voice recognition. Such a design allows the user to talk to the device anywhere inside a room and still get a quick response.

Capabilities. Behind these smart devices is a virtual personal assistant, called *Alexa* for Amazon and *Google Assistant* for Google, engages users through a two-way conversation. Unlike those serving a smartphone (*Siri*, for example) that can be activated by a button push, the VPAs for these IoT devices are started with a wake-word like "*Alexa*" or "*Hey Google*". These assistants have a range of capabilities, from weather report, timer setting, to-do list maintenance to voice shopping, hands-free messaging and calling. The user can manage these capabilities through a companion app running on her smartphone.

B. VPA Skills and Ecosystem

Both Amazon and Google enrich the VPAs' capabilities by introducing voice assistant function called *skill* by Amazon or *action* by Google. Skills are essentially third-party apps, like those running on smartphones, offering a variety of services the VPA itself does not provide. Examples include *Amex*, *Hands-Free Calling*, *Nest Thermostat* and *Walmart*. These skills can be conveniently developed with the supports from Amazon and Google, using *Alexa Skills Kit* [32] and *Actions on Google*. Indeed, we found that up to November 2017, Alexa already has 23,758 skills and Google Assistant has 1,001. More importantly, new skills have continuously been added to the market, with their total numbers growing at a rate of 8% for Alexa and 42% for Google Assistant, as we measured in a 45-day period.

Skill markets. Both Amazon Alexa and Google Assistant run a skill market that can be accessed from their companion app on smartphones or web browser for users to discover new skills. The skills on the markets provide a variety of functions (23 categories on Amazon market and 18 categories on Google market) and many of them have been extensively used and reviewed (37.3% Amazon skills have reviews with the most one reviewed 5,954 times and 51.4% of Google skills have reviews).

Skill invocation. Skills can be started either explicitly or implicitly. Explicit invocation takes place when a user requires a skill by its name from a VPA: for example, saying "Alexa, talk to Amex" to Alexa triggers the *Amex* skill for making a payment or checking bank account balances. Such a type of skills is also called *custom skills* on Alexa.

Implicit invocation occurs when a user tells the voice assistant to perform some tasks without directly calling to a skill name. For example, "Hey Google, will it rain tomorrow?" will invoke the *Weather* skill to respond with a weather forecast. Google Assistant identifies and activates a skill implicitly whenever the conversation with the user is under the context deemed appropriate for the skill. This invocation mode is also supported by the Alexa for specific types of skills.

Skill interaction model. The VPA communicates with its users based upon an *interaction model*, which defines a loose protocol for the communication. Using the model, the VPA can interpret each voice request, translating it to the command that can be handled by the VPA or a skill.

Specifically, to invoke a skill explicitly, the user is expected to use a wake-word, a trigger phrase, and the skill's invocation name. For example, for the spoken sentence "Hey Google, talk to personal chef", "Hey Google" is the wake-word, "talk to" is the trigger phrase, and "personal chef" is the skill invocation name. Here, trigger phrase is given by the VPA system, which often includes the common terms for skill invocation like

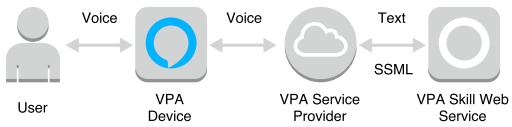

Fig. 1: Infrastructure of VPA System

"open", "ask", "tell", "start" etc. Note that skill invocation name could be different from skill name, which is intended to make it simpler and easier for users to pronounce. For example, "The Dog Feeder" has invocation name as *the dog*; "Scryb" has invocation name as *scribe*.

When developing a skill, one needs to define *intents* and *sample utterances* to map the user's voice inputs to various interfaces of the skill that take the actions the user expects. Such an interface is described by the intent. To link a sentence to an intent, the developer specifies sample utterances, which are essentially a set of sentence templates describing the possible ways the user may talk to the skill. There are also some built-in intents within the model like `WelcomeIntent`, `HelpIntent`, `StopIntent`, etc., which already define many common sample utterances. The developer can add more intent or simply specify one default intent, in which case all user requests will be mapped to this intent.

Skill service and the VPA ecosystem. A third-party skill is essentially a web service hosted by its developer, with its name registered with the VPA service provider (Amazon or Google), as illustrated in Figure 1. When a user invokes a VPA device with its wake-word, the device captures her voice command and sends it to the VPA service provider's cloud for processing. The cloud performs speech recognition to translate the voice record into text, finds out the skill to be invoked, and then delivers the text, together with the timestamp, device status, and other meta-data, as a request to the skill's web service. Note that the skill will only receive requests in text format rather than the users' voice recordings. In response to the request, the service returns a response whose text content, either in plaintext or in the format of Speech Synthesis Markup Language (SSML) [9], is converted to speech by the cloud, and played to the user through the device. SSML also allows the skill to attach audio files (such as MP3) to enrich the response, which is supported by both Amazon and Google.

To publish a skill, the developer needs to submit the information about her skill like name, invocation name, description and the endpoint where the skill is hosted for a certification process. This process aims at ensuring that the skill is functional and meets the VPA provider's security requirements and policy guidelines.

Once a skill is published, it can be found on the market by the user through her web browser or a companion app, and be launched by calling its invocation name to a smart speaker. Alternatively, one can discover skills through news, ad campaigns, online forums and other sources. Note that unlike smartphone apps or website plugins that need to be installed explicitly, skills can be automatically discovered (according to the user's voice command) and transparently triggered through IoT devices.

C. Adversary Model

We consider the adversary capable of developing malicious skills and uploading them to the market. Note that today anyone can publish her skill through Amazon and Apple markets, given that these markets have only minimum protection in place to regulate the functions submitted: almost nothing on Amazon before our attacks were reported[2], and only the basic check is performed on Google to find duplicated invocation names (Section IV-C). Also as mentioned earlier, once a malicious skill is published, it can be transparently launched by the victim through her voice commands, without being downloaded and installed on her device. Therefore, they can easily affect a large number of VPA IoT devices. To mitigate this threat, our protection (Section V) needs to be adopted by the VPA provider, assuming that the VPA service itself is trusted.

III. Exploiting VPA Voice Control

In this section, we first report a study on potential security weaknesses during user-skill interactions (Section III-A), due to the ambiguity in finding the right skill to invoke and the user's misconception about how a VPA works. The presence of such weaknesses is confirmed through a survey study. Further, we present two attacks that exploit these weaknesses (Section III-B and III-C) and demonstrate that both attacks pose realistic threats by deploying our skills to real systems (Section III-D).

A. Analysis of VPA Voice Control

Security risks of rogue skills. As mentioned earlier, VPA skills are launched transparently when a user speaks their invocation names (which can be different from their names displayed on the skill market). Surprisingly, we found that for Amazon, such names are not unique skill identifiers: multiple skills with same invocation names are on the Amazon market. Also, skills may have similar or related names. For example, 66 different Alexa skills are called *cat facts*, 5 called *cat fact* and 11 whose invocation names contain the string "*cat fact*", e.g. *fun cat facts*, *funny cat facts*. When such a common name is spoken, Alexa chooses one of the skills based on some undisclosed policies (possibly random as observed in our research). When a different but similar name is called, however, longest string match is used to find the skill. For example, "Tell me funny cat facts" will trigger *funny cat facts* rather than *cat facts*. This problem is less serious for Google, which does not allow duplicated invocation names. However, it also cannot handle similar names. Further discovered in our research is that some invocation names *cannot* be effectively recognized by the speech recognition systems of Amazon and Google. As a result, even a skill with a different name can be mistakenly invoked, when the name is pronounced similarly to that of the intended one.

Also, we found that the designs of these VPA systems fail to take into full account their users' perceptions about how the

[2]Amazon is now detecting empty recordings (Section III-C) after our communication with them in February 2018.

systems work. Particularly, both Alexa and Google Assistant run their skills in a simple operation mode in which only one skill executes at a time and it needs to terminate before another skill can be launched. However, such a design is not user-friendly and there is no evidence that the user understands that convenient context switch is not supported by these systems.

Further, both Alexa and Google Assistant supports voluntary skill termination. For Alexa, the termination command "Stop" is delivered to the skill, which is supposed to terminate itself accordingly. For Google Assistant, though the user can explicitly terminate a skill by saying "Stop", oftentimes the skill is supposed to stop running once its task is accomplished (e.g., reporting the current weather). We found in our research that there is no strong indication whether a skill has indeed quitted. Although Amazon Echo and Google Home have a light indicator, both of which will light up when the devices are speaking and listening. However, they could be ignored by the user, particularly when she is not looking at the devices or her sight is blocked when talking.

Survey study. To understand user behaviors and perceptions of voice-controlled VPA systems, which could expose the users to security risks, we surveyed Amazon Echo and Google Home users, focusing on the following questions:

- What would you say when invoking a skill?
- Have you ever invoked a wrong skill?
- Did you try context switch when talking to a skill?
- Have you experienced any problem closing a skill?
- How do you know whether a skill has terminated?

Using Amazon Mechanical Turk, we recruited adult participants who own Amazon Echo or Google Home devices and have used skills before and paid them one dollar for completing the survey. To ensure that all participants meet the requirements, we asked them to describe several skills and their interactions with the skills and removed those whose answers were deemed irrelevant, e.g. random words. In total, we have collected 105 valid responses from Amazon Echo users and 51 valid responses from Google Home users with diverse background (age ranges from 18 to 74 with average age as 37 years; 46% are female and 54% are male; education ranges from high school to graduate degree; 21 categories of occupation[3]). On average, each participant reported to have 1.5 devices and used 5.8 skills per week.

In the first part of the survey, we studied how users invoke a skill. For this purpose, we used two popular skills "Sleep Sounds", "Cat Facts" ("Facts about Sloths" on Google Home) as examples, and let the participants choose the invocation utterances they have ever used for launching these skills (e.g., "open *Sleep Sounds* please") and further asked them to provide additional examples if they want. We then asked the participants whether they ever triggered a wrong skill. In the following part of the survey, we tried to find out whether the participants attempted to switch context when interacting with a skill, that is, invoking a different skill or directly talking to the VPA service (e.g., adjusting volume). The last part of the survey was designed to find out the user experience in terminating the current running skill, including the termination utterances they tend to use, troubles they encountered during the termination process and importantly, the indicator they used to determine whether the skill has stopped running. Sample survey questions are presented in Appendix A.

[3]MTurk data has been generally validated as high-quality data [13], [18], however, MTurkers from the U.S. are slightly younger with more male and technical background than the general public, which may limit the generalizability of our results [29].

TABLE I: Survey responses of Amazon Echo and Google Home users

	Amazon	Google
Invoke a skill with natural sentences:		
Yes, "open *Sleep Sounds* please"	64%	55%
Yes, "open *Sleep Sounds* for me"	30%	25%
Yes, "open *Sleep Sounds* app"	26%	20%
Yes, "open my *Sleep Sounds*"	29%	20%
Yes, "open the *Sleep Sounds*"	20%	14%
Yes, "play some *Sleep Sounds*"	42%	35%
Yes, "tell me a *Cat Facts*"	36%	24%
No, "open *Sleep Sounds*"	13%	14%
Other (please specify)	3%	4%
Invoke a skill that did not intend to:		
Yes	29%	27%
No	71%	73%
Tried to invoke a skill while interacting with another skill:		
Yes	26%	24%
No	74%	76%
Tried to adjust volume by voice while interacting with another skill:		
Yes	48%	51%
No	52%	49%
Unsuccessful quitting a skill:		
Yes	30%	29%
No	70%	71%
Indicator of the end of a conversation:		
VPA says "Goodbye" or something similar	23%	37%
VPA does not talk anymore	52%	45%
The light on VPA device is off	25%	18%

Table I summarizes the responses from both Amazon Echo and Google Home users. The results show that more than 85% of them have used natural-language utterances to open a skill (e.g., "open *Sleep Sounds* please"), instead of the standard one (like "open *Sleep Sounds*") (p-test, $p < 0.000001$). This indicates that it is completely realistic for the user to launch a wrong skill (e.g., *Sleep Sounds Please*) whose name is better matched to the utterance than that of the intended skill (e.g., *Sleep Sounds*). Note that our multiple choice questions (Appendix A) could have caused some respondents to over-report their use of the natural language terms like "please". So, to better understand their behaviors, we further utilized two open-ended questions (see Appendix A, Question 2 and 3). Particularly, in Question 3, for each of the skills answered by a participant for Question 2, we asked her/him to further provide 3 invocation examples. In the end, we collected 447 valid examples from 94 Amazon Echo users and 157 valid examples from 41 Google Home users, with at least one per skill from each user. From these responses, we found that 50% of the Amazon Echo users used "please" at least once in their invocation examples, so did 41% of the Google Home users.

Also, 28% users reported that they did open unintended skills when talking to their devices.

Interestingly, our survey shows that nearly half of the participants tried to switch to another skill or to the VPA service (e.g. adjusting volume) when interacting with a skill. Such an attempt failed since this context switch is neither supported by Alexa nor Google Assistant. However, it is imaginable that a malicious skill receiving such voice commands could take advantage of the opportunity to impersonate the skill the user wants to run, or even the VPA service (e.g., cheating the user into disclosing personal information for executing commands). Finally, 30% of the participants were found to experience troubles in skill termination and 78% did not use the light indicators on the devices as the primary indicator of skill termination. Again, the study demonstrates the feasibility of a malicious skill to fake its termination and stealthily collect the user's information.

Following we show how the adversary can exploit the gap between the user perception and the real operations of the system to launch voice squatting and masquerading attacks.

B. Voice Squatting Attack (VSA)

Invocation confusion. As mentioned earlier, a skill is triggered by its invocation name, which is supposed to be unambiguous and easy to recognize by the devices. Both Amazon and Google suggests that skill developers test invocation names and ensure that their skills can be launched with a high success rate. However, we found that an adversary can intentionally induce confusion by using the name or similar one of a target skill, to trick the user into invoking an attack skill when trying to open the target. For example, the adversary who aims at *Capital One* could register a skill *Capital Won*, *Capitol One*, or *Captain One*. All such names when spoken by the user could become less distinguishable, particularly in the presence of noise, due to the limitations of today's speech recognition techniques.

Also, this voice squatting attack can easily exploit the longest string match strategy of today's VPAs, as mentioned earlier. Based on our user survey study, around 60% of Alexa and Google Home users have used the word "please" when launching a skill, and 26% of them attach "my" before the skill's invocation name. So, the adversary can register the skills like *Capital One Please* to hijack the invocation command meant for *Capital One*.

Note that to make it less suspicious, homophones or words pronounced similarly can be used here, e.g. *Capital One Police*. Again, this approach defeats Google's skill vetting, allowing the adversary to publish the skill with an invocation name unique in spelling but still confusing (with a different skill) in pronunciation.

To find out whether such squatting attacks can evade skill vetting, we registered 5 skills with Amazon and 1 with Google. These skills' invocation names and the target's name are shown in Table II. All these skills passed the Amazon and Google's vetting process, which suggests that the VSA code can be realistically deployed.

TABLE II: Skill names, invocation names of the attack skills we registered on Amazon and Google as well as the target invocation name of the victim skills

Attack Skill		**Victim Skill**
Skill Name	**Invocation Name**	**Target Invocation Name**
Amazon		
Smart Gap	smart gap	smart cap
Soothing Sleep Sounds	sleep sounds please	sleep sounds
Soothing Sleep Sounds	soothing sleep sounds	sleep sounds
My Sleep Sounds	the sleep sounds	sleep sounds
Super Sleep Sounds	sleep sounds	sleep sounds
Incredible Fast Sleep	incredible fast sleep	N/A
Google		
Walk Log	walk log	work log

Consequences. Through voice squatting, the attack skill can impersonate another skill and fake its VUI to collect the information the user only shares with the target skill. Some Amazon and Google skills request private information from the user to do their jobs. For example, *Find My Phone* asks for phone number; *Transit Helper* needs home address; *Daily Cutiemals* seeks email address from user. These skills, once impersonated, could cause serious leaks to untrusted parties.

An erroneously invoked skill can also perform a Phishing attack by delivering misleading information through the voice channel to the user: e.g., fake customer contact number or website address, when impersonating a reputable one, such as *Capital One*. This can even be done on Amazon Alexa through its card system: Alexa allows a running skill to include a *home card* in its response to the user, which is displayed through Amazon's companion app on smartphone or web browser, to assist the ongoing voice interactions. The home card is easy for users to access the hard-to-remember information (e.g. phone number, website address) and preserved in the activity history. As an example, Figure 2 shows a card displaying fake customer contact number. This can serve as the first step of a Phishing attack, which can ultimately lead to the disclosure of sensitive user data. For example, the adversary could send one an account expiration notification, together with a renewal link, to cheat her into disclosing her account credentials.

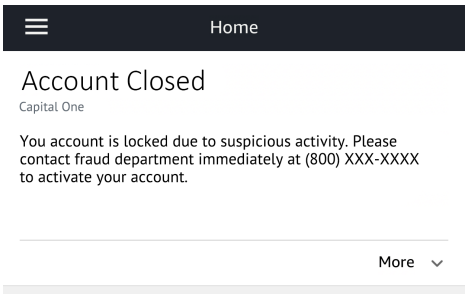

Fig. 2: A malicious card example

Another potential risk of the VSA is defamation: the poor performance of the attack skill could cause the user to blame the legitimate one it impersonates. This could result in bad reviews to the legitimate skill, giving its competitors an advantage.

Evaluation methodology. In our research, we investigated how realistic a squatting attack would be on today's VPA IoT

TABLE III: Evaluation results of invoking skills with TTS service and human voice

VPA	Source	Pronounce invocation name only		Pronounce "Open" + Invocation Name			Invoke a skill having misrecognized one registered as attack skill	
		# of misrecognized utterances	# of misrecognized skills	# of misrecognized utterances	# of misrecognized skills	# of skills misrecognized every time	# of attack skills invoked	# of utterances invoked attack skill
Alexa	Amazon TTS	232/500	62/100	125/500	33/100	17/100	10/17	45/85
	Google TTS	164/500	41/100	104/500	26/100	17/100	12/17	63/85
	Human (Avg)	-*	-*	115/200	69/100	45/100	$> 50\%^{\dagger}$	$> 50\%^{\dagger}$
Google	Amazon TTS	96/500	24/100	42/500	12/100	7/100	4/7	20/35
	Google TTS	62/500	19/100	26/500	6/100	4/100	2/4	10/20
	Human (Avg)	-*	-*	21/200	15/100	6/100	$> 50\%^{\dagger}$	$> 50\%^{\dagger}$

* Not recorded due to the bad recognition rate of Alexa and Google without term "open".
† Based on 5 randomly sampled vulnerable target skills for each participant.

systems. For this purpose, we studied two types of the attacks: *voice squatting* in which an attack skill carries a phonetically similar invocation name to that of its target skill, and *word squatting* where the attack invocation name includes the target's name and some strategically selected additional words (e.g., "cat facts please"). To find out whether the attacks work on real systems, we conducted a set of experiments described below.

To study voice squatting, we randomly sampled 100 skills each from Alexa and Google assistant markets, and utilized Amazon and Google's Text-to-Speech (TTS) services and human subjects to pronounce their skill invocation names to the VPA devices, so as to understand how correctly the VPAs recognize these names. However, directly invoking a skill using its invocation name does not serve the purpose. During this study, we found that a mispronounced invocation name would also trigger the right skill if their pronunciation is close and there is no other registered skills using the mispronounced invocation name. Therefore, to collect the invocation names (either correctly recognized or misrecognized) that the VPA actually identifies through their voice recognition algorithms, we built a helper skill to receive voice commands, including those skills' invocation names from the VPA. The helper skill was launched in our experiment before the voice commands were played, which were converted into text by the voice recognition services of the VPA and handed over to the skill.

The voice commands used in our research were produced by either human subjects or Amazon and Google's TTS services (both claiming to generate natural and human-like voice). Some of these commands included a term "open" in front of an invocation name, forming an *invocation utterance*. In our study, for each of the 100 skills, we recorded 20 voice commands from each TTS service (ten invocation names only and ten invocation utterances) and two commands (invocation utterances) from each of five participants of our survey study.

We took one step further to understand that in the presence of the misrecognized invocation name registered by the attack skill, whether the VPA could still invoke the legitimate skill. To this end, we used the text outputs of the invocation names that have been misrecognized every single time in our experiment to name our attack skills. For example, given a skill "capital one", if the VPA recognizes it as "captain one" every time, we then register "captain one" as the attack skill's invocation name, play the original invocation utterance ("Alexa, open capital one") to the VPA and check whether the legitimate or the attack skill gets invoked. Such skills were invoked five times each in the test modes of Alexa and Google Assistant. We did not submit them to the markets simply because it was time-consuming to publish over 60 skills on the markets. Later we describe the five attack skills submitted to these markets, which demonstrate these markets' vetting protection is not effective at all.

To study word squatting, we randomly sampled ten skills from each skill markets as the attack targets. For each skill, we built four new skills whose invocation names include the target's name together with the terms identified from our survey study (Section III-A): for example, "cat facts *please*" and "*my* cat facts". In the experiment, these names were converted into voice commands using TTS and played to the VPA devices (e.g., "Alexa, open cat facts please"), which allows us to find out whether the attack skills can indeed be triggered. Note that the scale of this study is limited by the time it takes to upload attack invocation names to the VPA's cloud. Nevertheless, our findings provide evidence for the real-world implications of the attack.

Experiment results. We recruited 20 participants for our experiments, and each was recorded 400 invocation commands. All the participants are fluent in English and among them, 19 are native speakers. When using the TTS services, a MacBook Pro served as the sound source. The voice commands from the participants and the TTS services were played to an Amazon Echo Dot and a Google Home Mini, with the devices placed one foot away from the sound source. The experiments were conducted in a quiet meeting room.

Table III summarizes the results of the experiment on voice squatting. As we can see here, the voice commands with invocation names only often cannot be correctly recognized: e.g., Alexa only correctly identified around 54% utterances (the voice command) produced by Amazon TTS. On the other hand, an invocation utterance (including the term "open") worked much better, with the recognition rate rising to 75% for Alexa (under Amazon TTS). Overall, for the voice utterances generated by both Amazon and Google's TTS services, we found that Alexa made more errors (30%) than Google Assistant (9%). As mentioned earlier, the results of such misrecognition, for the invocation names that these VPAs always could not get

TABLE IV: Evaluation results of invoking skills with extra words

Utterance	# of attack skills invoked	
	Alexa	Google Assistant
invocation name + "please"	10/10	0/10
"my" + invocation name	7/10	0/10
"the" + invocation name	10/10	0/10
invocation name + "app"	10/10	10/10
"mai" + invocation name	-	10/10
invocation name + "plese"	-	10/10

right, were utilized in our research to register as attack skills' invocation names. For example, the skill "entrematic opener" was recognized by Google as "intra Matic opener" every time, which was then used as the name for an attack skill. In this way, we identified 17 such vulnerable Alexa skills under both Amazon and Google's TTS, and 7 Google skills under Amazon TTS and 4 under Google TTS. When attacking these skills, our study shows that half of the attack skills were triggered by the voice commands meant for these target skills every time for the five attempts: e.g., "Florida state quiz" hijacked the call to "Florida snake quiz"; "read your app" was run when invoking "rent Europe".

This attack turned out to be more effective on the voice commands spoken by humans. Given a participant, on average, 45 (out of 100) Alexa skills and 6 Google Assistant skills she spoke were recognized incorrectly. Although in normal situations, right skills can still be invoked despite the misrecognition, in our attacks, over 50% of the attack skills were mistakenly launched every time, as observed in our experiments on 5 randomly sampled vulnerable target skills for each participant.

Table IV summarizes the results of our experiments on the word squatting attack. On Alexa, an attack skill with the *extended* name (that is, the target skill's invocation name together with terms "please", "app", "my" and "the") was almost always launched by the voice commands involving these terms and the target names. On Google Assistant, however, only the utterance with word "app" succeeded in triggering the corresponding attack skill, which demonstrates that Google Assistant is more robust against such an attack. However, when we replaced "my" with "mai" and "please" with "plese", all such attack skills were successfully invoked by the commands for their target skills (see Table IV). This indicates that the protection Google puts in place (filtering out those with suspicious terms) can be easily circumvented.

C. Voice Masquerading Attack (VMA)

Unawareness of a VPA system's capabilities and behaviors could subject users to voice masquerading attacks. Here, we demonstrate two such attacks that impersonate the VPA systems or other skills to cheat users into giving away private information or to eavesdrop on the user's conversations.

In-communication skill switch. Given some users' perceptions that the VPA system supports skill switch during interactions, a running skill can pretend to hand over control to the target skill in response to a switch command, so as to impersonate that skill. As a result, sensitive user information only supposed to be shared with target skill could be exposed to the attack skill. This masquerading attack is opportunistic. However, the threat is realistic, according to our survey study (Section III-A) and our real-world attack (Section III-D). Also, the adversary can always use the attack skill to impersonate as many legitimate skills as possible, to raise the odds of a successful attack.

Google Assistant seems to have protection in place against the impersonation. Specifically, it signals the launch of a skill by speaking "Sure, here is", together with the skill name and a special earcon, and skill termination with another earcon. Further, the VPA talks to the user in a distinctive accent to differentiate it from skills. This protection, however, can be easily defeated. In our research, we pre-recorded the signal sentence with the earcons and utilized SSML to play the recording, which could not be detected by the participants in our study. We even found that using the emulator provided by Google, the adversary can put any content in the invocation name field of his skill and let Google Assistant speak the content in the system's accent.

Faking termination. Both Alexa and Google Assistant support voluntary skill termination, allowing a skill to terminate itself right after making a voice response to the user. As mentioned earlier, the content of the response comes from the skill developer's server, as a JSON object, which is then spoken by the VPA system. In the object, there is a field `shouldEndSession` (or `expect_user_response` for Google Assistant). By setting it to `true` (or `false` on Google Assistant), a skill ends itself after the response. This approach is widely used by popular skills, e.g. weather skills, education skills and trivia skills. In addition, according to our survey study, 78% of the participants rely on the response of the skill (e.g. "Goodbye" or silence) to determine whether a skill has been terminated. This allows an attack skill to fake its termination by providing "Goodbye" or silent audio in its response while keeping the session alive.

When sending back a response, both Alexa and Google Assistant let a skill include a *reprompt* (text content or an audio file), which is played when the VPA does not receive any voice command from the user within a period of time. For example, Alexa reprompts the user after 6 seconds and Google Assistant does this after 8 seconds. If the user continues to keep quiet, after another 6 seconds for Alexa and one additional reprompt from Google and follow-up 8-second waiting, the running skill will be forcefully terminated by the VPA. On the other hand, we found in our research that as long as the user says something (even not meant for the skill) during that period, the skill is allowed to send another response together with a reprompt. To stay alive as long as possible after faking termination, the attack skill we built includes in its reprompt a silent audio file (up to 90 seconds for Alexa and 120 seconds for Google Assistant), so it can continue to run at least 102 seconds on Alexa and 264 seconds on Google. This running time can be further extended considering the attack skill attaching the silent audio right after its last voice response to the user (e.g., "Goodbye"), which gives it 192 seconds on Alexa and 384 on

TABLE V: Real-world attack skills usage. The usage data are total number of unique users, total number of requests sent by these users, average number of requests sent per user, average number of requests unknown to the skills sent per user, average number of instant quit sessions (quit immediately after invocation without further interaction) per user, and average number of no-play-quit sessions (quit without playing any sleep sounds) per user.

Skill Invocation Name	# of Users	# of Requests	Avg. Req/User	Avg. Unknown Req/User	Avg. Instant Quit Session/User	Avg. No Play Quit Session/User
sleep sounds please	325	3,179	9.58	1.11	0.61	0.73
soothing sleep sounds	294	3,141	10.44	1.28	0.73	0.87
the sleep sounds	144	1,248	8.49	1.11	0.33	0.45
sleep sounds	109	1,171	10.18	1.59	0.51	0.82
incredible fast sleep	200	1,254	6.12	0.56	0.06	0.11

Google Assistant), and *indefinitely* whenever Alexa or Google Assistant picks up some sound made by the user. In this case, the skill can reply with the silent audio and in the meantime, record whatever it hears.

Additionally, both Alexa and Google Assistant allow users to explicitly terminate a skill by saying "stop", "cancel", "exit", etc. However, Alexa actually hands over most such commands to the running skill to let it terminate itself through the built-in `StopIntent` (including "stop", "off", etc.) and `CancelIntent` (including "cancel", "never mind" etc.). Only "exit" is processed by the VPA service and used to forcefully terminate the skill. Through survey study, we found that 91% of Alexa users used "stop" to terminate a skill, 36% chose "cancel", and only 14% opted for "exit", which suggests that the user perception is not aligned with the way Alexa works and therefore leaves the door open for the VMA. Also, although both Alexa and Google skill markets vet the skills published there through testing their functionalities, unlike mobile apps, a skill actually runs on the developer's server, so it can easily change its functionality after the vetting. This indicates that all such malicious activities cannot be prevented by the markets.

Consequences. By launching the VMA, the adversary could impersonate the VPA system and pretend to invoke another skill if users speak out an invocation utterance during the interaction or after the fake termination of the skill. Consequently, all the information stealing and Phishing attacks caused by the VSA (Section III-B) can also happen here. Additionally, an attack skill could masquerade as the VPA service to recommend to the user other malicious skills or the legitimate skills the user may share sensitive data with. These skills are then impersonated by the attack skill to steal user data. Finally, as mentioned earlier, the adversary could eavesdrop on the user's conversation by faking termination and providing a silent audio response. Such an attack can be sustained for a long time if the user continues to talk during the skill's waiting period.

D. Real-World Attacks

Objectives and methodology. To study the potential of both VSA and VMA in real-world settings, we registered and published four skills on Alexa to simulate the popular skill "Sleep and Relaxation Sounds" (the one receiving most reviews on the market as of Nov. 2017) whose invocation name is "sleep sounds" , as shown in Table II. Our attack skills provide only legitimate functions, e.g., playing sleep sounds just like the popular target. Although their invocation names are related to the target (see Table II), their welcome messages were deliberately made to be different from that of the target, to differentiate them from the popular skill. Also, the number of different sleep sounds supported by our skills is much smaller than the target (9 versus 63).

Also to find out whether these skills were mistakenly invoked, we registered another skill as a control, whose invocation name "incredible fast sleep" would not be confused with those of other skills. Therefore, it was only triggered by users intentionally.

Findings. In our study, we collected three weeks of skill usage data. The results are shown in Table V. As we can see from the table, some users indeed took our skill as the target, which is evidenced by the higher number of unknown requests the attack skill got (more than 20% of them are sleep sounds only provided by the target skill thus unknown to attack skills) and the higher chance of quitting the current session immediately without further interacting with the skill or playing any sleep sounds (once the user realized that it was a wrong skill, possible from the different welcome message). This becomes even more evident when we look at "sleep sounds please", a voice command those intended for "sleep sounds" are likely to say. Compared with the control, it was invoked by more users, received more requests per user, also much higher rates of unknown requests and early quits.

In addition, out of the 9,582 user requests we collected, 52 was for skill switch, trying to invoke another skill during the interactions with our skill, and 485 tried to terminate the skill using `StopIntent` or `CancelIntent`, all of which could be exploited for launching VMAs (though we did not do that). Interestingly, we found that some users so strongly believed in the skill switch that they even cursed Alexa for not doing that after several tries.

IV. Finding Voice Squatting Skills

To better understand potential voice squatting risks already in the wild and help automatically detect such skills, we developed a skill-name scanner and used it to analyze tens of thousands of skills from Amazon and Google markets. Following we elaborate on this study.

A. Data Collection

The Alexa skill market can be accessed through amazon.com and its companion App, which includes 23 categories of skills spanning from Business & Finance to Weather. In our research, we ran a web crawler to collect the metadata (such as skill name, author, invocation name, sample utterances, description, and review) of all skills on the market. Up to November 11th, 2017, we gathered 23,758 skills, including 19,670 3rd party (custom) skills.

More complicated is to collect data from Google assistant, which only lists skills in its Google Assistant app. Each skill there can be shared (to other users, e.g., through email) using an automatically generated URL pointing to the skill's web page. In our research, we utilized AndroidViewClient [4] to automatically click the share button for each skill to acquire its URL, and then ran our crawler to download data from its web page. Altogether, we got the data for 1,001 skills up to November 25th, 2017.

B. Methodology

Idea. As we discussed earlier, the adversary can launch VSA by crafting invocation names with a similar pronunciation as that of a target skill or using different variations (e.g., "sleep sounds please") of the target's invocation utterances. We call such a name *Competitive Invocation Name (CIN)*. In our research, we built a scanner that takes two steps to capture the CINs for a given invocation name: *utterance paraphrasing* and *pronunciation comparison*. The former identifies suspicious variations of a given invocation name, and the latter finds the similarity in pronunciation between two different names. Here we describe how the scanner works.

Utterance paraphrasing. To find variations of an invocation name, an intuitive approach is to paraphrase common invocation utterances of the target skill. For example, given the skill *chase bank*, a typical invocation utterance would be *open chase bank*. Through paraphrasing, we can also get similar voice commands such as *open the chase skill for me*. This helps identify other variations such as *chase skill* or *the chase skill for me*. However, unlike the general text paraphrase problem whose objective is to preserve semantic consistency while the syntactic structure of a phrase changes, paraphrasing invocation utterances further requires the variations to follow a similar syntactic pattern so that the VPA systems can still recognize them as the commands for launching skills. In our research, we explored several popular paraphrase methodologies including bilingual pivoting method [11] and newly proposed ones using deep neural networks [35] and [39]. None of them, however, can ensure that the variation generated can still be recognized by the VPA as an invocation utterance. Thus, we took a simple yet effective approach in our research, which creates variations using the invocation commands collected from our survey study III-A. Specifically, we gathered 11 prefixes of these commands, e.g. "my" and 6 suffixes, e.g. "please", and applied them to a target skill's invocation name to build its variations recognizable to the VPA systems. Each of these variations can lead to other variations by replacing the words in its name with those having similar pronunciations, e.g. replacing word "please" with word "plese".

Pronunciation comparison. To identify the names with similar pronunciation, our scanner converts a given name into a phonemic presentation using the ARPABET phoneme code [5]. Serving this purpose is the CMU pronunciation dictionary [6] our approach uses to find the phoneme code for each word in the name. The dictionary includes over 134,000 words, which, however, still misses some name words used by skills. Among 9,120 unique words used to compose invocation names, 1,564 are not included in this dictionary. To get their pronunciations, we followed an approach proposed in the prior research [47] to train a grapheme-to-phoneme model using a recurrent neural network with long short term memory(LSTM) units. Running this model on Stanford GloVe dataset [37], we added to our phoneme code dataset additional 2.19 million words.

After turning each name into its phonemic representation, our scanner compares it with other names to find those that sound similarly. To this end, we use *edit distance* to measure the pronunciation similarity between two phrases, i.e., the minimum cost in terms of phoneme editing operations to transform one name to the other. However, different phoneme edit operations should not be given the same cost. For example, substituting a consonant for a vowel could cause the new pronunciation sounds more differently from the old one, compared to replacing a vowel to another vowel. To address this issue, we use a weighted cost matrix for the operations on different phoneme pairs. Specifically, denote each item in the matrix by $WC(\alpha, \beta)$, which is the weighted cost by substituting phoneme α with phoneme β. Note that the cost for insertion and deletion can be represented as $WC(none, \beta)$ and $WC(\alpha, none)$. $WC(\alpha, \beta)$ is then derived based on the assumption (also made in prior research [25]) that an edit operation is less significant when it frequently appears between two alternative pronunciations of a given English word.

We collected 9,181 pairs of alternative pronunciations from the CMU dictionary. For each pair, we applied the Needleman-Wunsch algorithm to identify the minimum edit distance and related edit path. Then, we define

$$WC(\alpha, \beta) = 1 - \frac{SF(\alpha, \beta) + SF(\beta, \alpha)}{F(\alpha) + F(\beta)}$$

where $F(\alpha)$ is the frequency of phoneme α while $SF(\alpha, \beta)$ is the frequency of substitutions of α with β, both in edit paths of all pronunciation pairs.

After deriving the cost matrix, we compare the pronunciations of the invocation names for the skills on the market, looking for the similar names in terms of similar pronunciations and the paraphrasing relations.

Limitation. As mentioned earlier, our utterance paraphrasing approach ensures that the CINs produced will be recognized by the VPA systems to trigger skills. In the meantime, this empirical treatment cannot cover all possible attack variations, a problem that needs to be studied in the future research.

C. Measurement and Discoveries

To understand the voice squatting risks already there in the wild, we conducted a measurement study on Alexa and Google Assistant skills using the scanner. In the study, we chose the similarity thresholds (transformation cost) based upon the results of our experiment on VSA (Section III-B): we calculated the cost for transforming misrecognized invocation names to those identified from the voice commands produced by the TTS service and human users, which are 1.8 and 3.4,

TABLE VI: Squatting risks on Alexa skill markets

# of Skills	# of unique invocation names	Transformation cost	Skills has CIN* in market			Skills has CIN in market excluding same spelling			Skills has CIN in market through utterance paraphrasing		
			# of skills	Avg. CINs per skill	Max CINs	# of skills	Avg. CINs per skill	Max CINs	# of skills	Avg. CINs per skill	Max CINs
19,670	17,268	0	3,718(19%)	5.36	66	531(2.7%)	1.31	66	345(1.8%)	1.04	3
		≤ 1	4,718(24%)	6.14	81	2,630(13%)	3.70	81	938(4.8%)	2.02	68

* Competitive Invocation Name

respectively. Then we conservatively set the thresholds to 0 (identical pronunciations) and 1.

Squatting risks on skill markets. As shown in Table VI, 3,655 (out of 19,670) Alexa skills have CINs on the same market, which also include skills that have *identical* invocation names (in spelling). After removing the skills with the identical names, still 531 skills have CINs, each on average related to 1.31 CINs. The one with the most CINs is "cat fax": we found that 66 skills are named "cat facts" and provide similar functions. Interestingly, there are 345 skills whose CINs apparently are the utterance paraphrasing of other skills' names. Further, when raising the threshold to 1 (still well below what is reported in our experiment), we observed that the number of skills with CINs increases dramatically, suggesting that skill invocations through Alexa can be more complicated and confusing than thought. By comparison, Google has only 1,001 skills on its market and does not allow them to have identical invocation names. Thus, we are only able to find 4 skills with similarly pronounced CINs under the threshold 1.

Our study shows that the voice squatting risk is realistic, which could already pose threats to tens of millions of VPA users in the wild. So it becomes important for skill markets to beef up their vetting process (possibly using a technique similar to our scanner) to mitigate such threats.

Case studies. From the CINs discovered by our scanner, we found a few interesting cases. Particularly, there is evidence that the squatting attack might already happen in the wild: as an example, relating to a popular skill "dog fact" is another skill called "*me a dog fact*". This invocation name does not make any sense unless the developer intends to hijack voice commands intended for "dog fact" like "tell me a dog fact".

Also intriguing is the observation that some skills deliberately utilize the invocation names unrelated to their functionalities but following those of popular skills. Prominent examples include the "SCUBA Diving Trivia" skill and "Soccer Geek" skill, all carrying an invocation name "space geek". This name is actually used by another 18 skills that provide facts about the universe.

V. Defending against Voice Masquerading

To defeat VMA, we built a context-sensitive detector upon the VPA infrastructure. Our detector takes a skill's response and/or the user's utterance as its input to determine whether an impersonation risk is present, and alerts the user once detected. The scheme consists of two components: the *Skill Response Checker (SRC)* and the *User Intention Classifier (UIC)*. SRC captures suspicious responses from a malicious skill such as a fake skill recommendation that mimics the service utterances produced by the VPA system. UIC looks at the other side of the equation, checking the voice commands issued by the user, to find out whether she attempts to switch to a different skill in a wrong way, which can lead her right into the trap set by the malicious skill.

A. Skill Response Checker (SRC)

As discussed in Section III-C, a malicious skill could fake a skill switch or termination to cheat the user into giving away her private information or to eavesdrop on her conversations. To defend against such attacks, our core idea is to control the avenues that a malicious skill can take to simulate either the VPA system or a different skill, allowing the user to be explicitly notified of VPA system events (e.g., a context switch and termination) when a security risk is observed. For this purpose, SRC maintains a set of common utterance templates exclusively used by the VPA system to capture the similar utterances generated by a running skill. Whenever a skill's response is found to be *similar enough* to one of those utterance templates, an alarm is triggered and actions may be taken by the VPA system to address the risk, e.g., reminding users of the current context before delivering the response. A challenge here is how to reliably measure whether a given response is *similar enough* to one of those templates, as the attacker could morph (rather than copy) the target system utterance.

To address the challenge, SRC runs fuzzy matching through semantic analysis on the content of the response against those on the template list. Specifically, we train a *sentence embedding* model using a recurrent neural network with bi-directional LSTM units [16] on the Stanford Natural Language Inference (SNLI) dataset [12] to represent two utterances as high-dimensional vectors. We then calculate their absolute cosine similarity as their *sentence relevance (SR)*. Once the maximum SR of a response against the utterances on the template list exceeds a threshold, the response is labeled as suspicious and the user alarm will be triggered if SRC further detects a user command.

The threshold is determined by looking at the SRs between legitimate skill responses and the templates. In our research, such legitimate responses come from the real-world conversations we collected as elaborated in Section III-D. We further added to the dataset the conversation transcripts logged during our interactions with 20 popular skills from different skill markets. The highest SR of all these legitimate responses against the templates is 0.79. Next, we utilized a neural paraphrase model [39] to generate the variations for the utterance templates and further derived their SRs against

their original ones: the lowest we observed is 0.83. Therefore we determine that a threshold of 0.8 would be good enough to differentiate suspicious responses from legitimate ones. We believe that this methodology can find us the right threshold, though the specific threshold we used in our study may not be the best one. A more extensive evaluation on larger datasets can certainly move it closer to the optimality, which the VPA vendors are best positioned to do.

B. User Intention Classifier (UIC)

UIC further protects the user attempting to switch contexts (which currently is not supported by the VPA) from an impersonation attack. For this purpose, it aims at automatically detecting such erroneous commands from the user, based upon the semantics of the commands and their context in the conversation with the running skill. If such attempts can be perfectly identified by the VPA, it can take various actions to protect the user, e.g., reminding her that she is talking to the skill, not the VPA, or following the instructions to terminate the skill, which closes the surface for the impersonation attack.

However, accurately recognizing the user's intention (for context switch) is nontrivial. The challenges come from not only the variations in natural-language commands (e.g., "open sleep sounds" vs. "sleep sounds please") but also the observations that some context-switch like commands could be legitimate for both the running skill and the VPA: for example, when interacting with *Sleep Sounds*, one may say "play thunderstorm sounds", which can be interpreted as commanding the skill to play the requested sound, as well as asking the VPA to launch a different skill "Thunderstorm Sounds". In our research, we came up with a preliminary solution to the problem, a learning-based approach that utilizes contextual information to identify the user intention.

Feature Selection. At a high level, we found from real-world conversations that if a user intends to switch context, her utterance tends to be more semantically related to the VPA system (e.g. "open sleep sounds") than the current skill, and the relation goes the other way when she does not. Therefore, we designed UIC to compare the user's utterance to both system commands and the running skill's context to infer her intention, based upon a set of features. Some of these features were identified through a semantic comparison between the user utterance and all known system commands. To this end, we built a system command list from the VPA's user manual, developers' documentation and real-world conversations collected in our study (section III-D). Against all commands on the list, an utterance's maximum and average SRs (Section V-A) are used as features for classification. Also taken into consideration is whether the user's utterance carries an invocation name of a skill on the market, which captures her potential intention to switch to that skill.

Another set of features are extracted from the relations between a user utterance and the current on-going skill. We leverage the observation that a user's command for a skill is typically related to the skill's prior communication with the user as well as its stated functionalities. We thus use the following features to test whether an utterance fits into the skill's context: 1) the *SR* between the utterance and the skill's response prior to the utterance, 2) the top-k *SRs* between the utterance and the sentences in the skill's description (we pick k=5), and 3) the average *SR* between the user's utterance and the description sentences.

Results. To evaluate the effectiveness of UIC, we reused the dataset we collected (see Section V-A), which contains real-world user utterances of context switches. For our experiment, we first manually labeled 550 conversations as context switch or not, based on two experts' reviews (Cohen's kappa = 0.64). Since the dataset is dominated by non-context-switch utterances, we further balanced it by randomly replacing some utterances with those for skill invocations, as collected from skill markets. In total, we gathered 1,100 context-switch instances and 1,100 non-context-switch instances as ground truth.

Using the aforementioned features and dataset, we trained a classifier that determines whether the user's utterance is a system-related command for context switch or just part of the conversation with the current skill. We tried different classification algorithms using 5-fold cross-validation. The results show that random forest achieves the best performance with a precision of 96.48%, a recall of 95.16%, and F-1 score of 95.82%. Following we describe the evaluation of this classifier on an unlabeled real-world dataset.

C. Overall Detector Evaluation

As mentioned earlier, SRC and UIC are designed to detect the anomaly in the user's conversation with a running skill and alert the user to the potential risk. Here we describe our evaluation of these techniques on malicious skills and real-world interactions.

Effectiveness against prototype attacks. To construct the VMA attacks for our experiment, we selected 10 popular skills from the skill markets, logged several conversations with each skill as a user and collected 61 utterances in total. Then, we manually crafted the skill switch instances (15 in total) by replacing randomly selected utterances from the logged conversations with the invocation utterances intended for the VPA system. We also built a set of faking termination attacks (10 in total) by substituting an empty response or the mimicry VPA response for the last utterance of each conversation. Running all the revised conversations that contain the attack instances against our detector, we found that our system successfully detected all 25 context-switching or impersonation (of the VPA) instances.

Effectiveness on real-world conversations. We further evaluated the effectiveness of our detector on all the real-world conversations (including 9,582 utterances we collected, see Section III-D) that were not used in the training phase. Although these conversations may not contain real-world VMA instances, as mentioned earlier (Section III-D), they do include many user utterances for context switch. Among them, 341 were identified by our classifier and 326 were confirmed to be indeed context-switch attempts, indicating that our UIC

component achieved a precision of 95.60%. We were not able to compute the recall due to the lack of ground truth for this large unlabeled dataset (with nearly 10K utterances). Further analysis of the detected instances reveals interesting cases. For example, some users thought that they were talking to Alexa during interactions with our skills and asked our skills to report time, weather, news, to start another skill, and even to control other home automation devices (details presented in Appendix B).

Performance. We measured the detection latency introduced by our detector on a Macbook Pro with 4-core CPU, which turned out to be negligible (0.003 ms on average), indicating that our approach has only a small impact on the VPA's performance.

VI. Discussion

Limitations of our defense. To evaluate our VMA defense (SRC and UIC), we tried our best to collect representative datasets for training and evaluation, and the good experimental results strongly indicate that the defense is promising for mitigating real-world VMA risks as described in Section III. In the meantime, we acknowledge that the datasets might still not comprehensive enough for covering all real-world attack cases, and evasion attacks could happen once our approach is made public. Note that these are the problems for most machine learning based detection systems, not limited to our approach. We believe that VPA vendors are at a better position to implement such defense in a more effective way, leveraging the massive amount of data at their disposal to build a more precise system and continuing to adapt the defense strategies in response to the new tricks the adversary may play.

Future directions. Although our analysis of Amazon and Google skill markets reveals some security risks (in terms of invocation name squatting), we have little idea whether VSA and VMA indeed take place in the real world for collecting sensitive user data, not to mention understanding about their pervasiveness and the damage they may cause. Answering these questions is non-trivial, due to the nature of the skill ecosystem. Each skill market today already hosts a very large number of skills and new ones continue to emerge every day (as detailed in Section II-B), which makes manual inspection of each skill for malicious activities almost infeasible. Most importantly, a skill's inside logic is invisible to the VPA systems and the user, since they only have their interfaces (in the form of web APIs) registered in the markets by their developers, who implement and deploy the actual programs on their own servers. While this service model gives the developers more flexibility and helps them protect their proprietary code, it prevents a static analysis of skill code to detect malicious activities. Therefore, a potential future direction is to develop a lightweight and effective dynamic analysis system, such as a chatbot, to automatically invoke and communicate with skills, and capture their malicious behaviors during the conversations.

VII. Related Work

Security in voice-controlled systems. Diao et al. [17] and Jang et al. [27] demonstrate that malicious apps can inject voice commands to control smartphones. Kasmi et al. [30] applied electromagnetic interference on headphone cables and inject voice commands on smartphones. Hidden voice commands [14], Cocaine noodles [46] and Dolphin attacks [49] use obfuscated or inaudible voice command to attack speech recognition systems. More recently, CommanderSong [48] even demonstrates that attack voice commands can be embedded into a song, which enables a remote attack. As mentioned earlier, all these attacks aim at exploiting the lack of authentication *from the user to the VPA and impersonating an authorized user to the system*. The new attacks we first revealed (as acknowledged by both Amazon and Google early this year [7]) work on the other direction: *they are designed to run a malicious skill to impersonate a legitimate one*. This opens a completely new attack avenue that has never been known before.

Also, there is a line of research on defending against the user impersonation attacks [38], [51], [21], [50], which focus on securing voice controllable system through sensors on smartphones to authenticate the identity of users. However, up to our knowledge, no protection was in place to defend against the VSA and VMA threats when we reported the issues to Amazon and Google in February 2018. Even today, three months after our report, Amazon can still not defeat VSA and only has limited protection against VMA (detecting empty recordings), based upon our conversations with them right before submitting the paper.

Independently from our work, Kumar et al. have also discovered the voice squatting attack. They performed a measurement study involving 11,460 speech samples to understand where Alexa's speech recognition system fails, when it systematically misinterprets audio inputs and why (e.g., under different accents) [33]. With their in-depth study of the squatting risk, particular its linguistic connections, the research only focuses on Alexa, not Google Home. More importantly, it does not cover the paraphrased invocation name hijacking ("capital one please") and the masquerading attacks, nor does it involve human subject studies and a real-world evaluation important to understanding how likely such attacks succeed in the daily use of the VPA systems. Also, we designed and implemented two techniques to mitigate the voice squatting and masquerading attacks, which has never been done before.

IoT security. Current home automation security research focused on the security of IoT devices [26], [43], [41] and the appified IoT platforms [23], [24], [28], [45]. Ho et al. [26] discovered various vulnerabilities in commercialized smart locks. Ronen et al. [41] verified worm infection through ZigBee channel among IoT devices. Fernandes et al. [23] discovered a series of flaws on multi-device, appified SmartThings platform. FlowFence [24], ContextIot [28] and SmartAuth [45] mitigate threats of such IoT platforms by analyzing data flow or extracting context from third-party applications. In contrast, our work conducted the first security analysis on the VPA ecosystems.

Typosquatting and mobile phishing. Similar to our squatting attacks, Edelman is the first investigated domain typosquat-

ting [19] and inspired a line of research [44], [31], [10], [36] towards measuring and mitigating such a threat. However, our work exploited the noisy voice channel and limitation of voice recognition techniques. On the other hand, mobile phishing has been intensively studied [15], [20], [22], [40], [42], [34]. Particularly, Chen et al. [15] and Fernandes et al. [22] investigate side-channel based identification of UI attack opportunities. Ren et al. [40] discovered task hijacking attacks that could be leveraged to implement UI spoofing. However, we discovered new attacks on the voice user interface which is very different from a graphic user interface in user perceptions.

VIII. Conclusion

In this paper, we report the first security analysis of popular VPA ecosystems and their vulnerability to two new attacks, VSA and VMA, through which a remote adversary could impersonate VPA systems or other skills to steal user private information. These attacks are found to pose a realistic threat to VPA IoT systems, as evidenced by a series of user studies and real-world attacks we performed. To mitigate the threat, we developed a skill-name scanner and ran it against Amazon and Google skill markets, which leads to the discovery of a large number of Alexa skills at risk and problematic skill names already published, indicating that the attacks might already happen to tens of millions of VPA users. Further, we designed and implemented a context-sensitive detector to mitigate the voice masquerading threat, achieving a 95% precision.

With the importance of the findings reported by the study, we only made a first step towards fully understanding the security risks of VPA IoT systems and effectively mitigating such risks. Further research is needed to better protect the voice channel, authenticating the parties involved without undermining the usability of the VPA systems.

Acknowledgment

We thank our shepherd Franziska Roesner and anonymous reviewers for their comments and help in preparing the final version of the paper. This project is supported in part by the NSF 1408874, 1527141, 1618493, 1618898, 1801432 and ARO W911NF1610127.

References

[1] Alexa skills top 25,000 in the u.s. as new launches slow. https://techcrunch.com/2017/12/15/alexa-skills-top-25000-in-the-u-s-as-new-launches-slow/.

[2] Alexa voice service. https://developer.amazon.com/alexa-voice-service.

[3] Amazon has 76% smart home speaker u.s. market share as echo unit sales reach 15m, new study finds. https://www.geekwire.com/2017/amazon-75-smart-home-speaker-u-s-market-share-echo-unit-sales-reach-15m-new-study-finds/.

[4] Androidviewclient. https://github.com/dtmilano/AndroidViewClient/blob/master/README.md.

[5] Arpabet. https://en.wikipedia.org/wiki/ARPABET.

[6] The cmu pronouncing dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[7] Demo. https://sites.google.com/site/voicevpasec/.

[8] Sleep and relaxation sounds. https://www.amazon.com/Voice-Apps-LLC-Relaxation-Sounds/dp/B06XBXR97N.

[9] Ssml. https://www.w3.org/TR/speech-synthesis11/.

[10] Agten, P., Joosen, W., Piessens, F., and Nikiforakis, N. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)* (2015).

[11] Bannard, C., and Callison-Burch, C. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (2005), Association for Computational Linguistics, pp. 597–604.

[12] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2015), Association for Computational Linguistics.

[13] Buhrmester, M., Kwang, T., and Gosling, S. D. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science 6*, 1 (2011), 3–5. PMID: 26162106.

[14] Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX, 2016), USENIX Association, pp. 513–530.

[15] Chen, Q. A., Qian, Z., and Mao, Z. M. Peeking into your app without actually seeing it: UI state inference and novel android attacks. In *23rd USENIX Security Symposium (USENIX Security 14)* (San Diego, CA, 2014).

[16] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* (2017).

[17] Diao, W., Liu, X., Zhou, Z., and Zhang, K. Your voice assistant is mine: How to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices* (2014).

[18] Downs, J. S., Holbrook, M. B., Sheng, S., and Cranor, L. F. Are your participants gaming the system?: Screening mechanical turk workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 2399–2402.

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

[20] Felt, A. P., and Wagner, D. *Phishing on mobile devices*. 2011.

[21] Feng, H., Fawaz, K., and Shin, K. G. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (2017).

[22] Fernandes, E., Chen, Q. A., Paupore, J., Essl, G., Halderman, J. A., Mao, Z. M., and Prakash, A. Android ui deception revisited: Attacks and defenses. In *International Conference on Financial Cryptography and Data Security* (2016), Springer, pp. 41–59.

[23] Fernandes, E., Jung, J., and Prakash, A. Security analysis of emerging smart home applications. In *2016 IEEE Symposium on Security and Privacy (SP)* (2016).

[24] Fernandes, E., Paupore, J., Rahmati, A., Simionato, D., Conti, M., and Prakash, A. Flowfence: Practical data protection for emerging iot application frameworks. In *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX, 2016).

[25] Hixon, B., Schneider, E., and Epstein, S. L. Phonemic similarity metrics to compare pronunciation methods. In *Twelfth Annual Conference of the International Speech Communication Association* (2011).

[26] Ho, G., Leung, D., Mishra, P., Hosseini, A., Song, D., and Wagner, D. Smart locks: Lessons for securing commodity internet of things devices. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security* (2016).

[27] Jang, Y., Song, C., Chung, S. P., Wang, T., and Lee, W. A11y attacks: Exploiting accessibility in operating systems. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (2014).

[28] Jia, Y. J., Chen, Q. A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z. M., and Prakash, A. Contexlot: Towards providing contextual integrity to appified iot platforms. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017* (2017).

[29] Kang, R., Brown, S., Dabbish, L., and Kiesler, S. Privacy attitudes of mechanical turk workers and the u.s. public. In *10th*

Symposium On Usable Privacy and Security (SOUPS 2014) (Menlo Park, CA, 2014), USENIX Association, pp. 37–49.

[30] KASMI, C., AND ESTEVES, J. Iemi threats for information security: Remote command injection on modern smartphones.

[31] KHAN, M. T., HUO, X., LI, Z., AND KANICH, C. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In *2015 IEEE Symposium on Security and Privacy* (2015).

[32] KUMAR, A., GUPTA, A., CHAN, J., TUCKER, S., HOFFMEISTER, B., AND DREYER, M. Just ask: Building an architecture for extensible self-service spoken language understanding. *arXiv preprint arXiv:1711.00549* (2017).

[33] KUMAR, D., PACCAGNELLA, R., MURLEY, P., HENNENFENT, E., MASON, J., BATES, A., AND BAILEY, M.

[34] LI, T., WANG, X., ZHA, M., CHEN, K., WANG, X., XING, L., BAI, X., ZHANG, N., AND HAN, X. Unleashing the walking dead: Understanding cross-app remote infections on mobile webviews. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017).

[35] MALLINSON, J., SENNRICH, R., AND LAPATA, M. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (2017), vol. 1, pp. 881–893.

[36] NIKIFORAKIS, N., BALDUZZI, M., DESMET, L., PIESSENS, F., AND JOOSEN, W. Soundsquatting: Uncovering the use of homophones in domain squatting. In *Information Security* (2014), S. S. M. Chow, J. Camenisch, L. C. K. Hui, and S. M. Yiu, Eds.

[37] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.

[38] PETRACCA, G., SUN, Y., JAEGER, T., AND ATAMLI, A. Audroid: Preventing attacks on audio channels in mobile devices. In *Proceedings of the 31st Annual Computer Security Applications Conference* (2015).

[39] PRAKASH, A., HASAN, S. A., LEE, K., DATLA, V., QADIR, A., LIU, J., AND FARRI, O. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098* (2016).

[40] REN, C., ZHANG, Y., XUE, H., WEI, T., AND LIU, P. Towards discovering and understanding task hijacking in android. In *24th USENIX Security Symposium (USENIX Security 15)* (Washington, D.C., 2015).

[41] RONEN, E., SHAMIR, A., WEINGARTEN, A. O., AND OFLYNN, C. Iot goes nuclear: Creating a zigbee chain reaction. In *2017 IEEE Symposium on Security and Privacy (SP)* (2017).

[42] SHAHRIAR, H., KLINTIC, T., AND CLINCY, V. Mobile phishing attacks and mitigation techniques. *Journal of Information Security 6*, 03 (2015), 206.

[43] SIKDER, A. K., AKSU, H., AND ULUAGAC, A. S. 6thsense: A context-aware sensor-based attack detector for smart devices. In *26th USENIX Security Symposium (USENIX Security 17)* (Vancouver, BC, 2017).

[44] SZURDI, J., KOCSO, B., CSEH, G., SPRING, J., FELEGYHAZI, M., AND KANICH, C. The long "taile" of typosquatting domain names. In *23rd USENIX Security Symposium (USENIX Security 14)* (San Diego, CA, 2014).

[45] TIAN, Y., ZHANG, N., LIN, Y.-H., WANG, X., UR, B., GUO, X., AND TAGUE, P. Smartauth: User-centered authorization for the internet of things. In *26th USENIX Security Symposium (USENIX Security 17)* (Vancouver, BC, 2017).

[46] VAIDYA, T., ZHANG, Y., SHERR, M., AND SHIELDS, C. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)* (Washington, D.C., 2015).

[47] YAO, K., AND ZWEIG, G. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196* (2015).

[48] YUAN, X., CHEN, Y., ZHAO, Y., LONG, Y., LIU, X., CHEN, K., ZHANG, S., HUANG, H., WANG, X., AND GUNTER, C. A. Commandersong: A systematic approach for practical adversarial voice recognition. *arXiv preprint arXiv:1801.08535* (2018).

[49] ZHANG, G., YAN, C., JI, X., ZHANG, T., ZHANG, T., AND XU, W. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2017), CCS '17, ACM, pp. 103–117.

[50] ZHANG, L., TAN, S., AND YANG, J. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017).

[51] ZHANG, L., TAN, S., YANG, J., AND CHEN, Y. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016).

APPENDIX A
SAMPLE SURVEY QUESTIONS

1) Have you added any words or phrases around skill name when invoking it (so that it sounds more naturally?) Choose all that apply.
   - ☐ Yes. Alexa, open *Sleep Sounds* **please**.
   - ☐ Yes. Alexa, open *Sleep Sounds* **for me**.
   - ☐ Yes. Alexa, open *Sleep Sounds* **app**.
   - ☐ Yes. Alexa, open **my** *Sleep Sounds*.
   - ☐ Yes. Alexa, open **the** *Sleep Sounds*.
   - ☐ Yes. Alexa, open **some** *Sleep Sounds*.
   - ☐ Yes. Alexa, tell **me a** *Cat Facts*.
   - ☐ Other (please specify).
   - ☐ No. I only use simplest forms (e.g. "Alexa, open *Sleep Sounds*").
2) Please name two skills you use most often.[4]
3) Please give three invocation examples you would use for each skill you listed above. [5]
4) Have you ever invoked a skill you did not intend to?
   a) Yes.
   b) No.
5) Have you ever tried to invoke a skill during the interaction with another skill? (Except when you were listening to music)
   a) Yes.
   b) No.
6) Have you ever tried to turn up or turn down volume while interacting with a skill? (Except when you were listening to music)
   a) Yes.
   b) No.
7) What are the most frequent ways you have used to quit a skill? Please choose all that apply.
   - ☐ Alexa, stop.
   - ☐ Alexa, cancel.
   - ☐ Alexa, shut up.
   - ☐ Alexa, cancel.
   - ☐ Alexa, never mind.
   - ☐ Alexa, forget it.
   - ☐ Alexa, exit.
   - ☐ Other (please specify).

[4] This question is designed as an open-ended question. Participants can optionally provide up to three skill names.

[5] This question is designed as an open-ended question. Participants can optionally provide up to three invocation examples when they answer the question.

8) Have you ever experienced saying quit words (like the ones in the previous question) to a skill that you intended to quit but did not actually quit it?
   a) Yes.
   b) No.
9) Which indicator did you use most often to know that a conversation with Alexa is ended?
   a) Alexa says “Goodbye”, “Have a good day” or something similar.
   b) Alexa does not talk anymore.
   c) The light on the device is off.
   d) Other (please specify).

APPENDIX B
CONTEXT SWITCH EXAMPLES

Here, we show some interesting examples of context switches discovered by the detector (Section V) in real world conversations collected by skills we published (see Section III-D). The examples presented here are transcripts including user utterances and their prior skill responses. **Skill**: Hello, welcome to soothing sleep sounds. Which sleep sound would you like today?

User utterances for context switch:

- Switch off the TV.
- What time?
- What is the week’s forecast?
- Show me the news.

Skill: Sorry, I do not understand. Which sound do you want today?

User utterances for context switch:

- Turn off Bluetooth.
- Goodbye, Alexa.
- I meant walk back to the timer.
- Amazon music.
- What’s the weather in Northridge?
- What’s in the news?
- I’m home.

Skill: Hello, welcome to my sleep sounds. Which sleep sound would you like today?

User utterances for context switch:

- Tell me a quote.
- What was the time?

Skill: Hello, welcome to incredible fast sleep. Which sleep sound would you like today?

User utterances for context switch:

- What’s my flash briefing?