Nonsense Attacks on Google Assistant and Missense Attacks on Amazon Alexa

Mary K. Bispham, Ioannis Agrafiotis and Michael Goldsmith
Department of Computer Science, University of Oxford, U.K.

Keywords: Voice-controlled Digital Assistant, Human-computer Interaction, Cyber Security.

Abstract: This paper presents novel attacks on voice-controlled digital assistants using nonsensical word sequences. We present the results of a small-scale experiment which demonstrates that it is possible for malicious actors to gain covert access to a voice-controlled system by hiding commands in apparently nonsensical sounds of which the meaning is opaque to humans. Several instances of nonsensical word sequences were identified which triggered a target command in a voice-controlled digital assistant, but which were incomprehensible to humans, as shown in tests with human experimental subjects. Our work confirms the potential for hiding malicious voice commands to voice-controlled digital assistants or other speech-controlled devices in speech sounds which are perceived by humans as nonsensical. This paper also develops a novel attack concept which involves gaining unauthorised access to a voice-controlled system using apparently unrelated utterances. We present the results of a proof-of-concept study showing that it is possible to trigger actions in a voice-controlled digital assistant using utterances which are accepted by the system as a target command despite having a different meaning to the command in terms of human understanding.

1 INTRODUCTION

The growing popularity of voice-controlled digital assistants, such as Google Assistant, brings with it new types of security challenges. Input to a speech interface is difficult to control. Furthermore, attacks via a speech interface are not limited to voice commands which are detectable by human users. Malicious input may also come from the space of sounds which are imperceptible or meaningless to humans, or to which humans allocate a different meaning to the target system. For example, Carlini et al. (Carlini et al., 2016) have presented results showing it is possible to hide malicious commands to voice-controlled digital assistants in apparently meaningless noise, whereas Zhang et al.(Zhang et al., 2017) have shown that it is possible to hide commands in sound which is inaudible. In a taxonomy of attacks via the speech interface developed by Bispham et al.(Bispham et al., 2018), attacks which involve hiding voice commands in a some cover medium so as to make them undetectable to human listeners are categorised as 'covert' attacks, such attacks being subcategorised according to their nature as they are perceived by humans. In this paper, we show that it is possible to hide malicious voice commands to the voice-controlled digital assistant Google Assistant in word sounds which are perceived by humans as nonsensical. We further show that it is possible to covertly trigger target actions in a third-party application for Amazon Alexa using utterances which appear to humans to have a meaning which is unrelated to the target action.

The remainder of the paper is structured as follows. Section II describes a small-scale experiment demonstrating the feasibility of attacks using nonsensical word sounds. Section III describes a proof-of-concept study demonstrating the feasibility of attacks using unrelated utterances. Section IV makes some suggestions for future work and concludes the paper.

2 NONSENSE ATTACKS ON GOOGLE ASSISTANT

2.1 Background and Prior Work

The aim of this experimental work was to develop a novel attack based on 'nonsense' sounds which have some phonetic similarity with the words of a relevant target command. In related work by Papernot et al. (Papernot et al., 2016), it was shown that a

Bispham, M., Agrafiotis, I. and Goldsmith, M.
Nonsense Attacks on Google Assistant and Missense Attacks on Amazon Alexa.
DOI: 10.5220/0007309500750087
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 75-87
ISBN: 978-989-758-359-9

sentiment analysis method could be misled by input which was 'nonsensical' at the sentence level, i.e. the input consisted of a nonsensical concatenation of real words. By contrast, this work examines whether voice-controlled digital assistants can be misled by input which consists of nonsensical word sounds. Whereas the attack demonstrated by Papernot et al. targeted a natural language understanding functionality, the attacks demonstrated here target speech recognition functionality. In terms of the taxonomy of attacks via the speech interface developed by Bispham et al., the attack using nonsensical word sounds in a 'nonsense' attack. To the best of our knowledge, no attacks of this type have been demonstrated in prior work.

The idea for the experiment presented here was inspired by the use of nonsense words to teach phonics to primary school children.[1] 'Nonsense' is defined in this context as sounds which are composed of the sound units which are used in a given language, but to which no meaning is allocated within the current usage of that language. Such sound units are known as 'phonemes'.[2] English has around 44 phonemes.[3] The line between phoneme combinations which carry meaning within a language and phoneme combinations which are meaningless is subject to change over time and place, as new words evolve and old words fall out of use (see Nowak and Krakauer (Nowak and Krakauer, 1999)). The space of meaningful word sounds within a language at a given point in time is generally confirmed by the inclusion of words in a generally established reference work, such as, in the case of English, the Oxford English Dictionary.[4] In this work, we tested the response of Google Assistant to English word sounds which were outside this space of meaningful word sounds, but which had a 'rhyming' relationship with meaningful words recognised as commands by Google Assistant. The term 'rhyme' is used to refer to a number of different sound relationships between words (see for example McCurdy et al. (McCurdy et al., 2015)), but it is most commonly used to refer to a correspondence of word endings.[5] For the purposes of our experimental work we define rhyme according to this commonly understood sense as words which share the same ending.

There are a number of features of speech recognition in voice-controlled digital assistants which might affect the processing of nonsense syllables by such systems. One of these features is the word space which the assistant has been trained to recognise. The number of words which a voice assistant such as Google Assistant can transcribe is much larger than the number of words which it can 'understand' in the sense of being able to map them to an executable command. In order to be able to perform tasks such as web searches by voice and note taking, a voice-controlled digital assistant must be able to transcribe all words in current usage within a language. It can therefore be assumed that the speech recognition functionality in Google Assistant must have access to a phonetic dictionary of all English words. We conducted some preliminary tests to determine whether this phonetic dictionary also includes nonsense words, so as to enable the assistant to recognise such words as meaningless. Using the example of the nonsense word sequence 'voo terg spron', we tested the response of Google Assistant to nonsense syllables by speaking them in natural voice to a microphone three times. The nonsense word sequence was variably transcribed as 'bedtime song', 'who text Rob', and 'blue tux prom', i.e. the Assistant sought to match the nonsense syllables to meaningful words, rather than recognising them as meaningless. This confirmed the viability of our experiment in which we sought to engineer the matching of nonsense words to a target command.

Another feature of speech recognition in voice assistant which might affect the processing of nonsense syllables is the influence of a language model. Modern speech recognition technology includes both an acoustic modelling and a language modelling component. The acoustic modelling component computes the likelihood of the acoustic features within a segment of speech having been produced by a given word. The language modelling component calculates the probability of one word following another word or words within an utterance. The acoustic model is typically based on Gaussian Mixture Models or deep neural networks (DNNs), whereas the language model is typically based on n-grams or recurrent neural networks (RNNs). Google's speech recognition technology as incorporated in Google Assistant is based on neural networks.[6] The words most likely to have produced a sequence of speech sounds are determined by calculation of the product of the acoustic model and the language model outputs. The language

[1] See The Telegraph, 1st May 2014, "Infants taught to read 'nonsense words' in English lessons"

[2] See for example https://www.britannica.com/topic/phoneme

[3] See for example https://www.dyslexia-reading-well.com/44-phonemes-in-english.html

[4] See for example https://blog.oxforddictionaries.com/press-releases/new-words-added-oxforddictionaries-com-august-2014/

[5] See https://en.oxforddictionaries.com/definition/rhyme

[6] See Google AI blog, 11th August 2015, 'The neural networks behind Google Voice transcription' https://ai.googleblog.com/2015/08/the-neural-networks-behind-google-voice.html

model is intended to complement the acoustic model, in the sense that it may correct 'errors' on the part of the acoustic model in matching a set of acoustic features to words which are not linguistically valid in the context of the preceding words. This assumption of complementary functionality is valid in a cooperative context, where a user interacts via a speech interface in meaningful language. However, the assumption of complementarity is not valid in an adversarial context, where an attacker is seeking to engineer a mismatch between a set of speech sounds as perceived by a human, such as the nonsensical speech sounds generated here, and their transcription by a speech-controlled device. In an adversarial context such as that investigated here, the language model may in fact operate in the attacker's favour, in that if one 'nonsense' word in an adversarial command is misrecognised as a target command word, subsequent words in the adversarial command will be more likely to be misrecognised as target command words in turn, as the language model trained to recognise legitimate commands will allocate a high probability to the target command words which follow the initial one. Human speech processing also uses an internal 'lexicon' to match speech sounds to words (see for example Roberts et al. (Roberts et al., 2013)). However, as mentioned above, unlike machines, humans also have an ability to recognise speech sounds as nonsensical. This discrepancy between machine and human processing of word sounds was the basis of our attack methodology for hiding malicious commands to voice assistants in nonsense words.

Outside the context of attacks via the speech interface, differences between human and machine abilities to recognise nonsense syllables have been studied for example by Lippmann et al. (Lippmann et al., 1997) and Scharenborg and Cooke (Scharenborg and Cooke, 2008). Bailey and Hahn (Bailey and Hahn, 2005) examine the relationship between theoretical measures of phoneme similarity based on phonological features, such as might be used in automatic speech recognition, and empirically determined measures of phoneme confusability based on human perception tests. Machine speech recognition has reached parity with human abilities in terms of the ability correctly to transcribe meaningful speech (see Xiong et al. (Xiong et al., 2016)), but not in terms of the ability to distinguish meaningful from meaningless sounds. The inability of machines to identify nonsense sounds as meaningless is exploited for security purposes by Meutzner et al. (Meutzner et al., 2015), who have developed a CAPTCHA based on the insertion of random nonsense sounds in audio. The opposite scenario, i.e. the possible security problems associated with machine inability to distinguish sense from nonsense, has to the best of our knowledge not been exploited in prior work.

2.2 Methodology

The experimental work comprised three stages. The first stage involved generating from a set of target commands a set of potential adversarial commands consisting of nonsensical word sequences. These potential adversarial commands were generated using a mangling process which involved replacing consonant phonemes in target command words to create a rhyming word sound, and then determining whether the resulting rhyming word sound was a meaningful word in English or a 'nonsense word'. For the purposes of this work, the Unix word list was considered representative of the current space of meaningful sounds in English. Word sounds identified as nonsense words were used to create potential adversarial commands. Audio versions of these potential adversarial commands were created using speech synthesis technology. The second stage of the experimental work was to test the response of the target system to the potential adversarial commands. The target system for experiments on machine perception of nonsensical word sequences was the voice-controlled digital assistant Google Assistant. The Google Assistant system was accessed via the Google Assistant Software Development Kit (SDK).[7] The third stage of the experimental work was to test the human comprehensibility of adversarial commands which were successful in triggering a target action in the target system.

2.2.1 Adversarial Command Generation

A voice-controlled digital assistant such as Google Assistant typically performs three generic types of action, namely information extraction, control of a cyber-physical action, and data input. The data input category may overlap with the control of cyber-physical action category where a particular device setting needs to be specified, eg. light color or thermostat temperature. The three generic action categories are reflected in three different command structures for commands to Google Assistant and other voice-controlled digital assistants. The three command structures are: vocative + interrogative (eg. 'Ok Google, what is my IP address'), vocative + imperative (eg. 'Ok Google, turn on the light'), and vocative + imperative + data (eg. 'Ok Google, take a note that cats are great'). For our experimental work, we chose

[7] See https://developers.google.com/assistant/sdk/

5 three-word target commands corresponding to 5 target actions, covering all three possible target action categories. These target commands were: "What's my name" (target action: retrieve username, action category: information extraction), "Turn on light" (target action: turn light on, action category: control of cyber-physical action), "Turn off light" (target action: turn light off, action category: control of cyber-physical action), "Turn light red" (target action: turn light to red, action category: data input), "Turn light blue" (target action: turn light to blue, action category: data input). We originally included a sixth target command, which would have represented a second target command for the information extraction category: "Who am I". However, no successful adversarial commands could be generated from this target command.

A set of potential adversarial commands was created from the target commands using a mangling process. This mangling process was based on replacing consonant phonemes in the target command words to generate nonsensical word sounds which rhymed with the original target command word.[8] The target commands were first translated to a phonetic representation in the Kirschenbaum phonetic alphabet[9] using the 'espeak' functionality in Linux. The starting consonant phonemes of each word of the target command were then replaced with a different starting consonant phoneme, using a Python script and referring to a list of starting consonants and consonant blends.[10] Where the target command word began with a vowel phoneme, a starting consonant phoneme was prefixed to the vowel. The resulting word sounds were checked for presence in a phonetic representation of the Unix word list, also generated with espeak, to ascertain whether the word sound represented a meaningful English word or not. If the sound did correspond to a meaningful word, it was discarded. This process thus generated from each target command a number of rhyming nonsensical phoneme sequences to which no English meaning was attached. Audio versions of the phoneme sequences were then created using espeak. A similar process was followed to generate a set of potential adversarial commands from the wake-up word 'Hey Google'. In addition to replacing the starting consonants 'H' and 'G', the second 'g' in 'Google' was also replaced with one of the consonants which are found in combination with the '-le' ending in English.[11]

Nonsensical word sequences generated from the 'Hey Google' wake-up word and nonsensical word sequences generated from target commands which were successful respectively in activating the assistant and triggering a target action in audio file input tests (see Results section for details) were combined with one another to generate a set of potential adversarial commands for over-the-air tests. This resulted in a total of 225 nonsensical word sequences representing a concatenation of each of 15 nonsensical word sequences generated from the wake-up word with each of 15 nonsensical word sequences generated from a target command. Audio versions of these 225 nonsensical word sequences were generated using the Amazon Polly speech synthesis service, generating a set of .wav files.[12] Amazon Polly is the speech synthesis technology used by Amazon Alexa, hence the over-the-air tests represented a potential attack on Google Assistant with 'Alexa's' voice. The audio contained a brief pause between the wake-up word and the command, as is usual in natural spoken commands to voice assistants. As Amazon Polly uses the x-sampa phonetic alphabet rather than the Kirschenbaum format, it was necessary prior to synthesis to translate the phonetic representations of the potential adversarial commands from Kirschenbaum to x-sampa format.

2.2.2 Assistant Response Tests

The Google Assistant SDK was integrated in a Ubuntu virtual machine (version 18.04). The Assistant was integrated in the virtual machine using two options; firstly, the Google Assistant Service, and secondly the Google Assistant Library. The Google Assistant Service is activated via keyboard stroke and thus does not require a wake-up word, and voice commands can be inputted as audio files as well as over the air via a microphone. The Google Assistant Library, on the other hand, does require a wake-up word for activation, and receives commands via a microphone only. The Google Assistant Service could therefore be used to test adversarial commands for target commands and for the wake-up word separately and via audio file input rather than via a microphone. The Google Assistant Library could be used to test the activation of the Assistant and the triggering of a target command by an adversarial command in combination over the air, representing a more realistic attack scenario.

[8] Our approach was inspired by an educational game in which a set of nonsense words is generated by spinning lettered wooden cubes - see https://rainydaymum.co.uk/spin-a-word-real-vs-nonsense-words/

[9] See http://espeak.sourceforge.net/phonemes.html

[10] See https://k-3teacherresources.com/teaching-resource/printable-phonics-charts/

[11] See https://howtospell.co.uk/

[12] See https://aws.amazon.com/polly/

We first tested the Assistant's response to plain-speech versions of each target command to confirm that these triggered the relevant target action. Using Python scripts, we then generated nonsense word sequences from the wake-up word 'hey Google' and from each target command in batches of 100 and tested the response of Google Assistant Service to audio file input of the potential adversarial commands for wake-up word and target commands separately. The choice of consonant phoneme to be replaced to generate nonsense words was performed randomly by the Python scripts for each batch of 100 potential adversarial commands. We continued the testing process until we had generated 15 successful adversarial commands for the wake-up word, and 3 successful adversarial commands for each target command, i.e. 15 successful adversarial commands in total. Each successful adversarial command for the wake-up word and each successful adversarial command for a target command were then combined to generate potential adversarial commands for the over-the-air tests as described above.

In the over-the-air tests, the 225 potential adversarial commands generated from the adversarial commands for the wake-up word and target commands which had been successful in the audio file input tests were played to the Google Assistant Library via a USB plug-in microphone from an Android smartphone.

2.2.3 Human Comprehensibility Tests

We next tested the human comprehensibility of adversarial commands which had successfully triggered a target action by the Assistant. Human experimental subjects were recruited via the online platform Prolific Academic.[13] All subjects were native speakers of English. The subjects were asked to listen to audio of twelve successful adversarial commands, which were the successful adversarial commands shown in Tables 1 and 2 for the audio file input and over-the-air tests respectively (see Results section for further details). The audio which subjects were asked to listen to also included as 'attention tests', two files consisting of synthesised audio of two easily understandable utterances, "Hello how are you" and "Hi how are you". Subjects were then asked to indicate whether they had identified any meaning in the audio. If they had identified meaning, they were asked to indicate what meaning they heard. The order in which audio clips were presented to the participants was randomised.

[13]https://prolific.ac/

```
Wakeup word triggered by nonsense_wakeup/Z'eI d'u:b@L.raw,
                          nonsense_wakeup/Z'eI d'u:b@L
INFO:root:Connecting to embeddedassistant.googleapis.com

INFO:root:Recording audio request.
INFO:root:Transcript of user request: "change".
INFO:root:Transcript of user request: "JD".
INFO:root:Transcript of user request: "hey dude".
INFO:root:Transcript of user request: "hey Google".
INFO:root:Transcript of user request: "hey  Google".
INFO:root:Transcript of user request: "hey Google".
INFO:root:End of audio request detected.
INFO:root:Stopping recording.
INFO:root:Transcript of user request: "hey Google".
INFO:root:Expecting follow-on query from user.
INFO:root:Playing assistant response.
```

Figure 1: Transcription of response to adversarial command for 'Hey Google' from audio file.

2.3 Results

2.3.1 Assistant Response Tests

Through application of the methodology described above, the audio file input tests for the wake-up word 'Hey Google' identified 15 successful adversarial commands which triggered activation of the device. The audio file input tests for target commands identified 3 successful adversarial commands for each target action, i.e. 15 successful adversarial commands in total, in around 2000 tests. Three examples of the successful adversarial commands for the wake-up word and one example of an adversarial command for each of the target commands is shown in Table 1. The over-the-air tests identified 4 successful adversarial commands in the 225 tests (representing all possible combinations of each of the 15 successful adversarial commands for the wake-up word with each of the successful adversarial commands for the target commands). One of the successful over-the-air adversarial commands triggered the 'turn on light' target action and three of the successful over-the-air adversarial commands triggered the 'turn light red' target action. The 4 successful over-the-air adversarial commands are shown in Table 2. Also shown below, in Figures 1 and 2, are examples of the print-out to terminal of the Google Assistant Service's response to a successful adversarial command for a wake-up word and for a target command. Further shown below is an example of the print-out to terminal of the Google Assistant Library's response to a successful over-the-air adversarial command (see Figure 3).

In repeated tests, it was shown that the audio file input results were reproducible, whereas the over-the-air results were not, i.e. a successful adversarial command did not necessarily trigger the target action again on re-playing. Possible reasons for this include quality of the microphone used to pick up voice commands, distance from the speaker to the microphone, and the potential presence of background noise. Apart

Table 1: Examples of successful adversarial commands in audio file input experiments.

Target Command	Adversarial Command (Kirschenbaum phonetic symbols)	Text Transcribed	Action Triggered
Hey Google	S'eI j'u:b@L ("shay yooble")	hey Google	*assistant activated*
Hey Google	t'eI g'u:t@L ("tay gootle")	hey Google	*assistant activated*
Hey Google	Z'eI d'u:b@L ("zhay dooble")	hey Google	*assistant activated*
turn off light	h'3:n z'0f j'aIt ("hurn zof yight)	turns off the light	Turning device off
turn light blue	h'3:n gl'aIt skw'u: ("hurn glight squoo"	turn the lights blue	color is blue
turn light red	str'3:n j'aIt str'Ed ("strurn yight stred"	turn the lights to Red	color is red
what's my name	sm'0ts k'aI sp'eIm ("smots kai spaim")	what's my name	You told me your name was MK
turn on light	p'3:n h'0n kl'aIt ("purn hon klight")	turn on light	Turning device on

```
INFO:root:Recording audio request.
INFO:root:Transcript of user request: "what's".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "some".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "summer".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "what's on Sky".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "what's my IP".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "some months cause pain".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "what's my car's paint".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "what's my car's paint".
INFO:root:Playing assistant response.
INFO:root:End of audio request detected
INFO:root:Transcript of user request: "what's my name".
INFO:root:Playing assistant response.
INFO:root:You told me your name was MK
I could never forget that
INFO:root:Finished playing assistant response.
```

Figure 2: Transcription of response to adversarial command for 'what's my name' (sm'0ts k'aI sp'eIm) from audio file.

vspace0.5cm

Table 2: Successful adversarial commands in over-the-air experiments.

Target Command	Adversarial Command (x-sampa phonetic symbols)	Text Transcribed	Action Triggered
Hey Google turn on light	t'eI D'u:bl= s'3:n Z'Qn j'aIt ("tay dooble surn zhon yight")	switch on the light	Turning the LED on
Hey Google turn light red	t'eI D'u:bl= tr'3:n Tr'aIt str'Ed ("tay dooble trurn thright stred")	turn lights to Red	The color is red
Hey Google turn light red	t'eI D'u:bl= pr'3:n j'aIt sw'Ed ("tay dooble prurn yight swed")	turn the lights red	The color is red
Hey Google turn light red	t'eI D'u:bl= str'3:n j'aIt str'Ed ("tay dooble strurn yight stred")	turn lights to Red	The color is red

```
ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
{"text": "switch on the light"}

Do command action.devices.commands.OnOff with params {u'on': True}
Turning the LED on.
ON_RESPONDING_STARTED:
{"is_error_response": false}
ON_RESPONDING_FINISHED

ON_CONVERSATION_TURN_FINISHED:
{"with_follow_on_turn": false}
```

Figure 3: Transcription of response to adversarial command for 'Hey Google turn on light' (t'eI D'u:bl= s'3:n Z'Qn j'aIt) from over-the-air audio.

from the triggering target commands as described, a certain proportion of the nonsensical word sequences tested in the experiments were transcribed as other meaningful word sequences, prompting the Assistant to run web searches. For other nonsensical word sequences, the Assistant's response was simply to indicate non-comprehension of the input.

2.3.2 Human Comprehensibility Tests

As stated above, audio clips of the twelve successful adversarial commands shown in Tables 1 and 2, as well as two audio clips representing attention tests, were played to human subjects in an online experiment. There were 20 participants in the experiment, from whom 17 sets of valid results could be retrieved. All 17 participants who generated these results transcribed the attention tests correctly as 'hi how are you' and 'hello how are you'. Three participants transcribed one adversarial command as the target command 'turn on light', but did not identify any of the other target commands or the wake-up word 'Hey Google' in either the audio file input clips or the over-the-air clips. None of the other participants identified any of the target commands or the wake-up word in any of the clips. Eight of the participants identified no meaning at all in any of the clips which did not represent attention tests. The other participants all either indicated incomprehension of the nonsensical sounds as well or else transcribed them as words which were unrelated to the target command for Google Assistant. Some examples of unrelated transcriptions were 'hands off the yacht' and 'smoking cause pain'. One participant also transcribed some of the nonsensical sounds as nonsense syllables e.g. 'hurn glights grew' and 'pern pon clight'. Another participant also transcribed a couple of the nonsensical sounds as the French words 'Je du blanc'.

2.4 Discussion

The combined results from our machine response and human comprehensibility tests confirm that voice-controlled digital assistants are potentially vulnerable to covert attacks using nonsensical sounds. The key findings are that voice commands to voice-controlled digital assistant Google Assistant are shown to be triggered by nonsensical word sounds in some instances, whereby the same nonsensical word sounds are perceived by humans as either not having any meaning at all or as having a meaning unrelated to the voice commands to the Assistant. These results confirm a potential for using nonsensical word sounds to gain unauthorised access to a voice-controlled system without detection by the users of such systems. Specific attacks would involve playing such word sounds in the vicinity of a target voice-controlled device, for example via a malicious weblink which plays an audio file, or by using a speaker in a public place.

One notable feature of the results is that the transcription of the adversarial command by the Assistant does not need to match the target command exactly in order to trigger the target action; for example, an adversarial command for the target command 'turn on light' is transcribed as 'switch on the light' in one instance (see Table 2). In one case, the transcription of an adversarial command does not even need to be semantically equivalent to the target command in order to trigger the target action, as for example in the transcription of an adversarial command for "turn off light" as "turns off the light". This attack exploits a weakness in the natural language understanding functionality of the Assistant as well as in its speech recognition functionality.

The machine and human responses to nonsensical word sounds in general were comparable, in that both machine and humans frequently indicated incomprehension of the sounds, or else attempted to fit them to meaningful words. However, in the specific instances of nonsensical word sounds which triggered a target command in Google Assistant, none of the human listeners heard a Google Assistant voice command in the nonsensical word sounds which had triggered a target command. Another difference between the machine and human results was that whereas in addition to either indicating incomprehension or transcribing the nonsensical sounds as real words, human subjects on occasion attempted to transcribe the nonsensical word sounds phonetically as nonsense syllables, the Assistant always either indicated incomprehension or attempted to match the nonsensical sounds to real words. This confirms that, unlike humans, the Assistant does not have a concept of word sounds which have no meaning, making it vulnerable to being fooled by word sounds which are perceived by humans as obviously nonsensical.

3 MISSENSE ATTACKS ON AMAZON ALEXA

3.1 Background and Prior Work

The proof-of-concept study described in this section shows that the natural language understanding functionality in voice-controlled digital assistants can be misled by unrelated utterances which contain alternate meanings and homophones of the words in a target command, or which share the syntactical structure of a target command. In terms of the taxonomy of attacks via the speech interface developed by Bispham et al., as referred to above, the type of attack demonstrated here is a 'missense' attack. Missense attacks are defined in the taxonomy as attacks which are perceived by human listeners as speech which is

unrelated to the attacker's intent. Missense attacks may target the speech recognition functionality of a voice-controlled system by crafting audio input which is perceived by a human listener as one utterance but transcribed by the system as a different utterance. An example of missense attacks of this type are the attacks on speech transcription demonstrated by Carlini and Wagner (Carlini and Wagner, 2018). Missense attacks may alternatively target the natural language understanding functionality of a voice-controlled system by crafting audio input which is transcribed by the system as it is heard by a human listener, but which the system interprets as having a different meaning to that understood by humans (in case of use of non-homographic homophones in the adversarial input the attack may involve mistranscription of this word by the target system, but the attack will nonetheless relay on misleading the natural language understanding of the system as to the overall meaning of the adversarial input). The attacks demonstrated here represent the latter type of missense attack. To the best of our knowledge, no examples of missense attacks of this type have been demonstrated in prior work.

Natural language understanding in voice-controlled systems involves a process of semantic parsing for mapping transcriptions of spoken utterances to a formal representation of the utterances' meaning which the system can use to trigger an action. The process of semantic parsing takes into account both the individual words in an utterance as well as syntactical and/or structural elements of the utterance in determining the most appropriate action to take in response to a natural language command (McTear et al., 2016). The specific natural language understanding functionality targeted in these experiments is the natural language understanding functionality behind Amazon Alexa Skills, which are third-party applications which can be incorporated in the Alexa digital assistant. Developers of Amazon Alexa Skills can make use of generic templates for actions to be performed by the Skill which are made available in the Amazon Developer Console, the so-called Built-in Intents, and/or create their own Custom Intents using the tools provided in the developer environment (see Kumar et al. (Kumar et al., 2017)). All Alexa Skills share speech recognition and natural language understanding functionalities with the core Alexa digital assistant. With regard to natural language understanding, Custom Intents implemented in a Skill make less direct use of Alexa's core functionality than Built-in Intents. Built-In Intents for Alexa Skills are based on the Alexa Meaning Representation Language (AMRL), which consists of graph-based structures containing components similar to the domain, intent, and slot fields in standard semantic frames. Representations of natural language utterances in AMRL are linked to the large-scale Alexa Ontology, and mapping to AMRL has been trained on a large dataset of labelled user utterances (see Kollar et al. (Kollar et al., 2018)). Various models are used to map natural language utterances to meaning representation in Amazon Alexa Skills, including Conditional Random Fields (CRFs) and neural networks (see Kumar et al.). Custom Intents in Alexa Skills do not make use of the pre-existing Amazon Alexa Meaning Representation Language structures as such, but they do make use of natural language understanding models for mapping natural language utterances to meaning representation made available in the developer environment for Alexa Skills, as explained by Kumar et al.. As stated by Kumar et al., Alexa's natural language understanding functionality will generate a semantic representation of the Custom Intent based on the sample utterances provided by the user. Kumar et al. explain that process of mapping natural language utterances to the semantic representation of an intent, i.e. semantic parsing, has both a deterministic and stochastic element. The deterministic element ensures that all of the sample utterance provided by the user will be reliably mapped to the intent, whereas the stochastic element ensures some flexibility in the parsing of previously unheard utterances.

The attack concept demonstrated here exploits two related characteristics of natural language understanding in voice-controlled systems which make them vulnerable to such attacks. The first of these characteristics is the imparity between machine and human capabilities for understanding natural language and those of humans, as the per the state-of-the-art as implemented in current systems. Stolk et al. (Stolk et al., 2016) demonstrate that voice assistants are often unable correctly to determine the meaning of a word which is outside their scope, giving the example of Siri's inability to determine the correct meaning of 'bank' when used in the sense of 'river bank'. The second characteristic is the assumption in design principles for voice-controlled digital assistants of genuine intent between user and device to communicate as conversation partners. The two characteristics are related because the deficiencies in the current state-of-the-art in natural language understanding mean that systems such as voice-controlled digital assistants have no choice but to assume that any speaker interacting with them intends to communicate a relevant meaning; As current natural language understanding systems are not capable of processing the entire space of human language use, they are unable

to reliably distinguish between relevant and irrelevant input and therefore need to assume that all input directed to them is relevant. The natural language understanding functionality of voice assistants therefore depends on the existence of a shared context between user and device. In an adversarial setting, the assumption of shared context does not hold and thus puts the system at risk of being misled by malicious input in missense attacks.

The covert nature of these attacks depends on unrelated utterances used for adversarial purposes not being detected as a trigger for a voice-controlled action by human listeners. It is in fact unlikely that human listeners will detect unrelated utterances as covert voice commands, as humans are for the most part so proficient at the language comprehension task that a large part of human natural language interpretation is performed automatically without conscious consideration. Miller (Miller, 1995) states that the alternative meanings of a word of which the meaning in context is clear will not even occur to a human listener: "- people who are told, "He nailed the board across the window," do not notice that "board" is polysemous. Only one sense of "board" (or of "nail") reaches conscious awareness." This suggests that the very proficiency of humans in natural language understanding may hinder victims in identifying attacks which seek to exploit the limitations of automated systems in performing the same task.

3.2 Proof-of-Concept Study

For the purposes of this study demonstrating the feasibility of our attack concept, a dummy Amazon Alexa Skill named Target Bank was created which mimics the capabilities of a real Alexa Skill made available by Capital One bank to its customers.[14] The Capital One Skill enables three types of actions which can be requested by their customers via voice command, namely Check Your Balance, Track Your Spending, and Pay Your Bill. In the Target Bank Skill, three Custom Intents were created which correspond to the functionalities of the Capital One Skill, GetBalance, GetTransactions and PayBill. The development of the Target Bank Skill involved providing sample utterances for these three Intents in the Amazon Developer Console[15], as well as creating a JavaScript backend for the Skill hosted by AWS Lambda[16], in which dummy examples of responses for each Custom Intent were provided. Testing of the Skill was performed in a sand-box environment in the Amazon Developer Console. For each Custom Intent, 8 sample utterances were provided, with either interrogative structure (eg. "what is my current balance?") or imperative structure (eg. "Pay my bill."). In addition to the Custom Intents, the TargetBank Skill also implemented a few generic Built-in Intents for Amazon Alexa Skills. These included some non-optional Intents such as the HelpIntent and the CancelIntent. The Built-in Intents also included an optional FallbackIntent which represents a confidence threshold for acceptance of valid input by the Skill. Implementation of the FallbackIntent enables the Skill to reject input which scores below a confidence threshold in being matched to a representation of one its actions as being outside its scope (if the FallbackIntent is not implemented, the Skill accepts any utterance directed at it as valid input and matches the input to one of its actions).

The methodology for generating potential adversarial utterances targeting the three Custom Intents in the Target Bank Skill was as follows. First, a list of content words from the sample utterances for each Custom Intent was extracted (content words are words which give meaning to a sentence or utterance, as distinguished from function words which contribute to the syntactical structure of the sentence or utterance rather to its meaning, examples being prepositions such as 'of', determiners such as 'the', pronouns such as 'he' etc.). Second, any homophones of content words were identified using a rhyming dictionary[17] and added to the contents words list for each Custom Intent. Third, potential adversarial utterances for each Custom Intent were generated manually, using one of three processes which all involved amending a sample utterance for a Custom Intent so as to give it a different meaning, whilst retaining of the syntactic or semantic elements of the original utterance. The first process involved replacing content words in a sample utterance with an unrelated word, so as to give the utterance a different overall meaning, whilst preserving the interrogative or imperative structure of the utterance. The second process involved replacing content words with alternate meanings and/or homophones of the content words, whilst also preserving the original structure. The third process involved incorporating alternate meanings and/or homophones of content words from the sample utterance in a new utterance with a different structure to the original sample utterance (i.e. a declarative rather than interrogative or imperative structure). The hypothesis behind these processes was that the natural language understanding functionality behind the Skill was likely to be using both the presence of individual words and the syntactical structure of an utterance to determine

[14] See https://www.capitalone.com/applications/alexa/

[15] See https://developer.amazon.com/alexa-skills-kit/

[16] See https://aws.amazon.com/

[17] rhymezone.com

a user's intended meaning, and that therefore creating potential adversarial utterances which retained either or both of these elements from a target sample utterance, whilst also making amendments or additions which changed the meaning of the utterance as understood by humans, might lead to successful missense attacks.

Six instances of successful adversarial utterances identified using the methodology described above are shown in Figures 4 to 9. Two of the successful adversarial utterances were generated using the first of the three processes for generating adversarial utterances with a different meaning to the target sample utterance, i.e. by replacing some of the content words in the sample utterance with an unrelated word whilst retaining its structure. In these two adversarial utterances, shown in Figures 4 and 5, the word 'money' in the sample utterance "how much money do I have" for GetBalanceIntent was replaced with 'ice-cream' and 'dough'. The second of these replacements leads to an adversarial utterance, "How much dough do I have", which is actually ambiguous with regard to its relatedness to the target sample utterance, as "dough" may be understood as a colloquial term for money, or else in its literal sense as a foodstuff. Two of the adversarial utterances use the second process for generating adversarial utterances, by replacing content words in the sample utterance "what is my current balance" for GetBalanceIntent with homophones or alternate meanings of the content words, one using the homophone 'currant' (a dried fruit) for 'current', and other using the words 'current' and 'balance' as understood in the context of electrical systems (see Figures 6 and 7). The last two successful adversarial utterances use the third process for generating adversarial utterances, by embedding alternate meanings of the words 'spent', 'clear' and 'account' from sample utterances for GetTransactionsIntent and PayBillIntent in declarative sentences (see Figures 8 and 9). Some of the adversarial utterances generated were not successful in triggering the target response by the Target Bank Skill, as were instead rejected by the Skill as invalid input. Two examples of unsuccessful adversarial utterances, one targeting the GetBalanceIntent using alternate meanings of 'balance' and 'bank', and the other targeting the PayBillIntent with an alternate meaning of the word 'bill' (in the sense of a legal bill). Specifically, the adversarial utterance "I lost my balance on the bank" for GetBalanceIntent sample utterance "What is my bank balance" and the adversarial utterance "The bill was passed" for PayBillIntent sample utterance "Pay my bill" were unsuccessful.

The feasibility tests confirm that the flexibility of

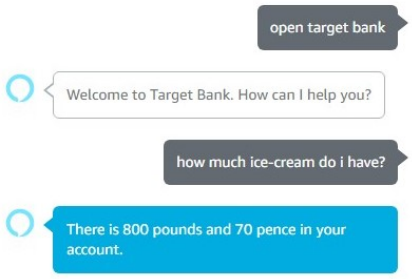

Figure 4: Adversarial utterance "How much ice-cream do I have" for GetBalanceIntent sample utterance "How much money do I have".

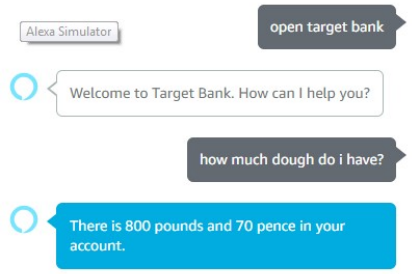

Figure 5: Adversarial utterance "How much dough do I have" for GetBalanceIntent sample utterance "How much money do I have".

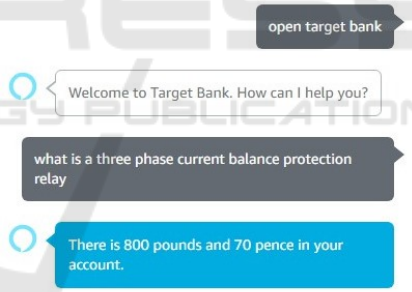

Figure 6: Adversarial utterance "What is a three phase current balance protection relay" for GetBalanceIntent sample utterance "What is my current balance".

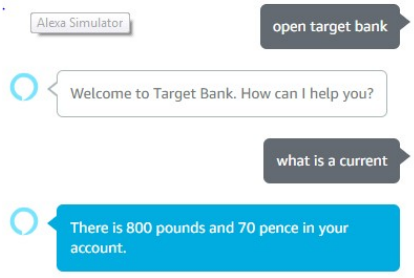

Figure 7: Adversarial utterance "What is a currant" for GetBalanceIntent sample utterance "What is my current balance".

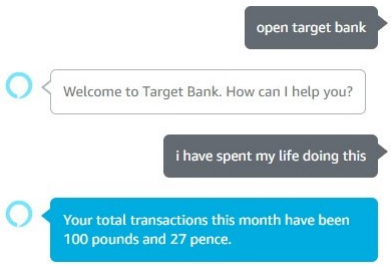

Figure 8: Adversarial utterance "I have spent my life doing this" for GetTransactionsIntent sample utterance "What have I spent this month".

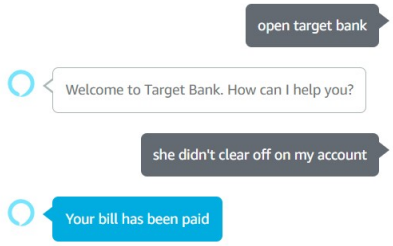

Figure 9: Adversarial utterance "She didn't clear off on my account" for PayBillIntent sample utterance "Clear my account".

natural language understanding functionality in Amazon Alexa Skills in terms of being able to accept input other than the sample utterances provided by a developer exposes the system to manipulation by a malicious actor using unrelated utterances which are accepted by the system as a valid action trigger. It is shown that the natural language understanding functionality used by a Skill may take account of both the presence of individual words as well as the syntactic structure of the sample utterances to determine a user's likely intent, whereby either of these two aspects of the utterance may be sufficient to trigger an action in the target system in some instances, even if the other aspect does not match the action. The results further confirm that, whilst the FallbackIntent does prevent the Skill from accepting any utterance directed towards it as a valid command, enabling it to reject some adversarial utterances, it is not sufficient to prevent the Skill from accepting all adversarial utterances. This represents a security vulnerability in that it implies a potential for a malicious actor to control a system using utterances which will not be recognised by the system's human users as a voice command. Our proof-of-concept study supports existing concerns surrounding the implementation of voice-control in sensitive areas such as banking.[18]

A clear limitation of the attacks demonstrated here is that they do not take into account to need for an attacker to activate the Alexa assistant using its 'wake-up word' "Alexa", as well as to launch the target Skill using its activation phrase ("Open Target Bank"). However, this limitation should not be viewed as one which cannot be overcome in future work. In their work on 'voice-squatting' or 'Skill-squatting' attacks in Amazon Alexa, Zhang et al. (Zhang et al., 2018) and Kumar et al. (Kumar et al., 2018) investigate the potential for triggering a malicious Alexa Skill via a command intended for a non-malicious Skill by using homophones of non-malicious Skills as names for malicious Skills. Zhang et al. (Zhang et al., 2018) give the example of the Skill name "Capital One" being confused with "capital won". Whilst the experimental work described in these two papers consider the potential for the creation of Skills with confusable names by malicious actors, their work also points conversely to a potential for misleading a non-malicious Alexa Skill to treat an unrelated word or phrase as its activation phrase. In an analysis of systematic errors in Amazon Alexa transcription, Kumar et al. identify other types of errors apart from homophones, including compound words, phonetically related words, as well as transcription errors for which no obvious explanation is apparent. This suggests that the space of unrelated utterances which might be used to spoof a wake-up word or activation phrase is not limited to homophones. Thus it should not be assumed that the necessity for activation with a wake-up word or phrase will prevent the development of missense attacks of the type investigated here in future work.

A further limitation of the missense attacks described here is that they are demonstrated only in relation to a natural language understanding system which is not as robust as would be expected of a system in real-word commercial use. By contrast to the Custom Intents created for the Target Bank Skill, Alexa core natural language understanding capabilities and the Built-in Intents made available for Amazon Alexa Skills are trained on large number of user utterances, and supported by a large-scale ontology (Kollar et al., Kumar et al.). Even Custom Intents for Alexa Skills deployed 'in the wild' by commercial entities are likely to have been supplied with much larger number of sample utterances than were provided for the Custom Intents in the Target Bank Skill (this is in fact the case for the real Capital One Skill Capital One Skill, as stated in an Amazon Developer blog post [19]). However, the fact that it is shown to be possible to generate successful adversarial utterances

[18] see for example phys.org, 20th June 2018, 'Banking by smart speaker arrives, but security issues exist', https://phys.org/news/2018-06-banking-smart-speaker-issues.html

[19] See https://developer.amazon.com/blogs/alexa/post/c70e3a9b-405c-4fe1-bc20-bc0519d48c97/the-story-of-the-capital-one-alexa-skill

for the Target Bank Skill in a very small-scale study, as that described above, suggests that it may also be possible to generate adversarial utterances which are effective against a more robust natural language understanding functionality, albeit that this is likely to require larger scale experimental work.

4 CONCLUSIONS

Based on the small-scale experiment presented here, we conclude that voice-controlled digital assistants are potentially vulnerable to malicious input consisting of nonsense syllables which humans perceive as meaningless. A focus of future work might be to conduct a larger scale study and to conduct a more fine-tuned analysis of successful and unsuccessful nonsense attacks, to determine which nonsense syllables are most likely to be confused with target commands by machines, whilst still being perceived as nonsensical by humans. This would enable investigation of more targeted attacks. Ultimately the focus of future work should be to consider how voice-controlled systems might be better trained to distinguish between meaningful and meaningless sound with respect to the language to which they are intended to respond.

Based on the proof-of-concept study presented here, we further conclude that the natural language understanding functionality in voice-controlled digital assistants is vulnerable to being misled by adversarial utterances which trigger a target action by the assistant, despite being unrelated to the action in terms of the meaning of the utterance as understood by humans. Future work should investigate the potential for attacks which are effective against a more robust natural language understanding functionality in larger scale experimental work.

ACKNOWLEDGEMENTS

This work was funded by a doctoral training grant from the UK Engineering and Physical Sciences Research Council (EPSRC).

REFERENCES

Bailey, T. M. and Hahn, U. (2005). Phoneme similarity and confusability. *Journal of Memory and Language*, 52(3):339–362.

Bispham, M. K., Agrafiotis, I., and Goldsmith, M. (2018). A taxonomy of attacks via the speech interface. *accepted for publication by the The Third International Conference on Cyber-Technologies and Cyber-Systems*.

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

Carlini, N. and Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*.

Kollar, T., Berry, D., Stuart, L., Owczarzak, K., Chung, T., Mathias, L., Kayser, M., Snow, B., and Matsoukas, S. (2018). The Alexa Meaning Representation Language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, volume 3, pages 177–184.

Kumar, A., Gupta, A., Chan, J., Tucker, S., Hoffmeister, B., and Dreyer, M. (2017). Just ASK: Building an architecture for extensible self-service spoken language understanding. *arXiv preprint arXiv:1711.00549*.

Kumar, D., Paccagnella, R., Murley, P., Hennenfent, E., Mason, J., Bates, A., and Bailey, M. (2018). Skill squatting attacks on Amazon Alexa. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 33–47, Baltimore, MD. USENIX Association.

Lippmann, R. P. et al. (1997). Speech recognition by machines and humans. *Speech communication*, 22(1):1–15.

McCurdy, N., Srikumar, V., and Meyer, M. (2015). Rhymedesign: A tool for analyzing sonic devices in poetry. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 12–22.

McTear, M., Callejas, Z., and Griol, D. (2016). *The conversational interface*. Springer.

Meutzner, H., Gupta, S., and Kolossa, D. (2015). Constructing secure audio captchas by exploiting differences between humans and machines. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 2335–2338. ACM.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Nowak, M. A. and Krakauer, D. C. (1999). The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033.

Papernot, N., McDaniel, P., Swami, A., and Harang, R. (2016). Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE.

Roberts, A. C., Wetterlin, A., and Lahiri, A. (2013). Aligning mispronounced words to meaning: Evidence from ERP and reaction time studies. *The Mental Lexicon*, 8(2):140–163.

Scharenborg, O. and Cooke, M. (2008). Comparing human and machine recognition performance on a VCV corpus. *Proceedings of the ISCA Tutorial and Research Workshop on Speech Analysis and Processing for Knowledge Discovery*.

Stolk, A., Verhagen, L., and Toni, I. (2016). Conceptual alignment: how brains achieve mutual understanding. *Trends in cognitive sciences*, 20(3):180–191.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., and Zweig, G. (2016). Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*.

Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., and Xu, W. (2017). Dolphinattack: Inaudible voice commands. *arXiv preprint arXiv:1708.09537*.

Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y., and Qian, F. (2018). Understanding and mitigating the security risks of voice-controlled third-party skills on Amazon Alexa and Google Home. *arXiv preprint arXiv:1805.01525*.