



Project Report - INSE 6120 - Cryptographic Protocols and Network Security
Winter 2019
Evaluating Skill Squatting on Amazon Alexa

Submitted To:
Dr. Mohammad Mannan

Submitted By:
Jaskaran Singh - 40071053
Ivan Acevedo - 40075812
Maninderjeet Singh - 40057916

1. Abstract

The project work on “Evaluating Skill Squatting Attack on Amazon Alexa” deals with the analysis of Amazon Alexa in the way it behaves differently on similar sounding voice commands. The research work on the topic is included in the 27th USENIX Security Symposium conducted by University of Illinois. The research work primarily focuses on the empirical analysis of interpretation errors made by Amazon Alexa’s speech recognition system which is the powerhouse of all the Amazon Echo devices.

The evaluation of the attacks done in the scope of this project is limited to the audio samples taken from five individuals and we maintain the ethical considerations of the same. In the first part of the project work we verify the results concluded in the research paper. We performed the skill squatting on existing skills in the amazon skill store and building pair of skills with the names that are frequently confused by Alexa. We experienced some deviations in the results as compared to that in the research paper. We thus concluded that Alexa speech recognition is still ambiguous and unpredictable.

We further the work on Skill Squatting by contributing to real world case scenario by performing skill squatting on American Express skills. The findings on the same shows the security implications related to Amazon Alexa. The possibility of the attack is discussed by establishing two threat models which summaries our novel conclusion.

2. Introduction

With no limitation to the advancements in technology and with already popular IOT devices, there has been a seen a trend of smart speakers which interacts with voice as the main interface. They possess the ability to perform a variety of tasks on a single voice command as input.

But with the advancements, there are always some limitations that go hand-in-hand. And in the case of smart speakers, they have been paid less attention to. The speech recognition system is not mature enough to understand the different accents. In our case study, we analysed the already done efforts on misinterpretation in speech recognition system by Alexa as mentioned in the research paper, but we also extended it by performing few of our tests on them and came up with some results and findings of our own.

According to the research paper and our extended study, we found that Alexa has the ability to perform an error when it comes to distinguishing between two similar sounding words. This weakness can be exploited as vulnerability by the attacker to perform malicious attacks such as squatting attack.

3. Background - Amazon Alexa

All the voice enabled devices of the Amazon family, has the intelligence to perform a task based on the voice given as an input to the devices. The mind working behind these devices to act smartly on your command is Alexa. This cloud based service acts as an intelligent assistant by using the speech recognition, machine learning and natural language recognition.

The Alexa is packed with the arsenal of abilities that are called skills. These skills let Alexa interact with the user, from playing music to know the weather. This is done with a voice command coupled with an executable task. Alexa comes with some predefined skills but also provides freedom to third party developers to develop their own skill with their custom defined audio inputs and tasks.

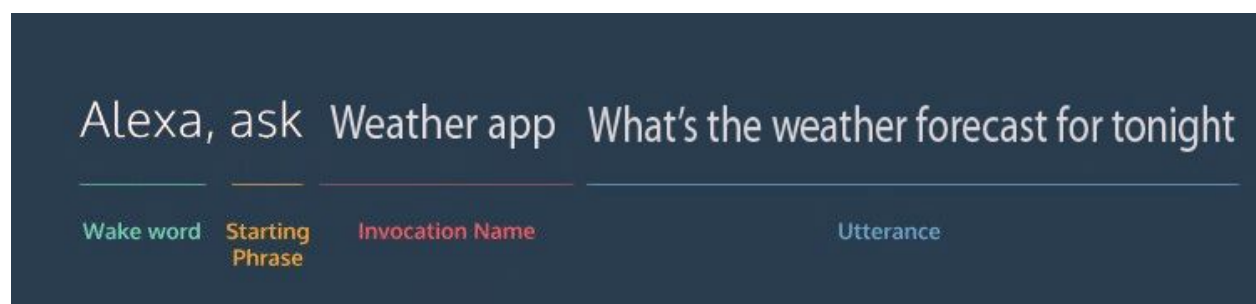
3.1. How to Build an Alexa Skill

These skills are built on same pattern as of any web applications or mobile applications are built. A skill is composed of two parts:

- **Interaction Model:** it is the front end of the application that the user sees. In other words just like a Graphical User Interface works, Alexa has a Voice user interface that holds the functionalities, actions and all the coding on how to act, to be used by the skill on the command.
- **Hosted Service:** It acts as the backend of the application. It comprises of the coding part of the skill that defines the response that should be given to the input received from the user.

3.2. How to interact with the skill

- **Wake Word:** Alexa requires a word to wake up and get ready for the task to perform. The word “Alexa” is the default word used to wake up the device and interact with Alexa.
- **Start Phase:** Followed by the word “Alexa” is the start phase. It indicates the type of the request that one uses.
- **Invocation Name:** The job that an invocation name does is to help Alexa direct to invoke a particular skill. Every skill, either built-in or custom made have unique Invocation Names assigned to them. For example:



3.4. Defining the Intents and the Utterances

- **Intents:** It decide the appropriate function or the behavior of the skill. A skill can have more than one intent defined for it. And assigned to every intent is a specific behavior which acts when it is called. It is very similar to how a click on different buttons works on a website in different ways. For eg: Hello is an intent.
- **Utterances:** These are the phrases that Alexa might respond to, with the appropriate response. For example how are you, Is it cold outside, is the ride here yet, etc?

3.5. Alexa Developer Console

To make a skill and define its behavior and responses, it is initiated by designing a Voice User Interface. Then we define everything in it. And this is done inside the Alexa Developer console.

In order to have access to the Alexa Developer Console, first we need to make an account in Amazon Developer Account. And creating an account is free of cost.

To access the Alexa Developer Console, the user can go to the Amazon Developer console (developer.amazon.com), login to their account or by creating a new account and click on “Skills” by hovering over ‘Your Alexa Consoles’.

4. Alexa Limitations

4.1. Phonemes

The phonemes are generally the way to represent a word that tells how that word will be pronounced. Every word has its own way of representation in phonemes. Alexa generally gets puzzled when it comes to two similar sounding words. And as a result it sometimes picks up the wrong word in place of the intended word and therefore opens the doors for squatting attack. For example, here we have the way “Coal” and “Call” are represented:

coal

call

K OW L

K AO L

Even though both the words have different phonetic representations, Alexa gets puzzled while recognizing the correct word.

4.2. Speech to Text Test Harness.

Alexa by default does not have the ability to supply the transcription of speech to text directly therefore it makes it available to the third party skill developer as a developer API feature. In the research paper provided, they tried to record all their speech to text transcriptions and save them all as a log in their server by developing a client that directs the files from Alexa files to their custom server. They first begin

by triggering the skill and thereby saving all the future requests to their custom skill server. **The research done, reflects a major flaw in the alexa speech recognition system i.e, Alexa is not able to differentiate between similar sounding words and it even interprets the same word differently when queried again.**

5. Dataset

In order to analyse the misinterpretation attacks by Alexa, the authors used two speech databases as mentioned in the research paper. They were Nationwide Speech Project (NSP) and Forvo. The NSP is provided by the Ohio State University that gives speech data of wide variety of speakers from United States. But in our approach, we did not use the NSP because they were charging fees for using it. So we came up with five random speakers, namely:

Name	Gender	Origin
Maninderjeet Singh	Male	India
Jaskaran Singh	Male	India
Ivan Acevedo	Male	Ecuador
Jose Sandlas	Female	Arab
Monica Sebi	Female	Arab

6. Test and Implementation

6.1. Creating a skill in Alexa Developer Console

6.1.1 Configuring the skill

In order to create a skill in Alexa, the user needs to register an account at Alexa Skills Kit (ASK). This platform allows users to create skills using Amazon services. The process is quite simple, and the user can use the same Amazon personal account if they have one. Once he login into the ASK platform, the user selects the **Create skill** option and an interface like the one shown below will be displayed. Then he needs to assign a name to the skill and select **Alexa-Hosted** if he wants to host the skill in Amazon servers.



Create a new skill

Skill name

4/50 characters

Default language

More languages can be added to your skill after creation

Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

Provision your own

Provision your own endpoint and backend resources for a skill. With this option, you will not gain access to the console's code editor.

Alexa-Hosted (Beta)

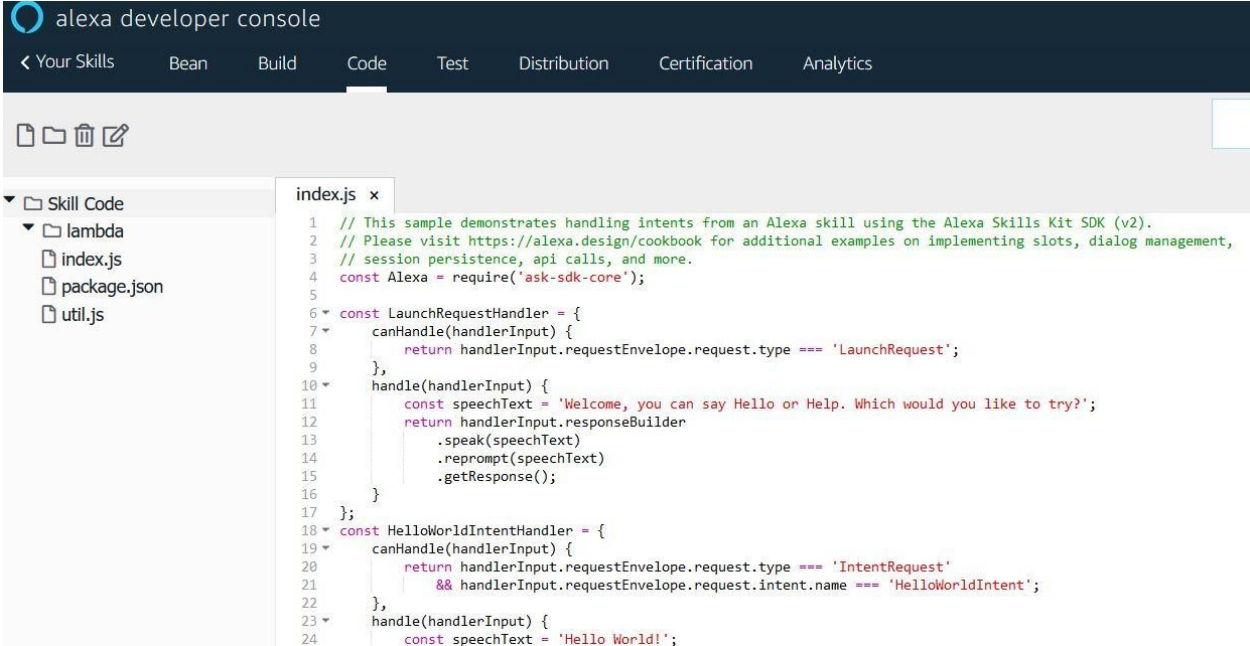
SELECTED

Alexa will host skills in your account up to the AWS Free Tier limits. You will gain access to an AWS Lambda endpoint, 5 GB of media storage with 15 GB of monthly data transfer, and a table for session persistence. [Learn more](#)

6.1.2 Coding Alexa Skill

One of the easiest ways to develop a skill is to use AWS Lambda. This is a service offered by Amazon as part of its Amazon Web Services (AWS) platform. AWS Lambda supports code written in Node.js, Java, Python, C#, or Go. For this project we decided to use this scheme.

All the skills we created simply included the following welcome text "**Welcome, you can say Hello or Help. Which would you like to try?**". Every time we invoked one of our skills that was the dialogue we displayed.



```

1 // This sample demonstrates handling intents from an Alexa skill using the Alexa Skills Kit SDK (v2).
2 // Please visit https://alexa.design/cookbook for additional examples on implementing slots, dialog management,
3 // session persistence, api calls, and more.
4 const Alexa = require('ask-sdk-core');
5
6 const LaunchRequestHandler = {
7   canHandle(handlerInput) {
8     return handlerInput.requestEnvelope.request.type === 'LaunchRequest';
9   },
10  handle(handlerInput) {
11    const speechText = 'Welcome, you can say Hello or Help. Which would you like to try?';
12    return handlerInput.responseBuilder
13      .speak(speechText)
14      .reprompt(speechText)
15      .getResponse();
16  }
17 };
18
19 const HelloWorldIntentHandler = {
20   canHandle(handlerInput) {
21     return handlerInput.requestEnvelope.request.type === 'IntentRequest'
22       && handlerInput.requestEnvelope.request.intent.name === 'HelloWorldIntent';
23   },
24   handle(handlerInput) {
25     const speechText = 'Hello World!';
  
```

6.1.3. Testing an Alexa Skill

Once a skill has been created, the user can access the **Test** section to check that it works correctly. In the Amazon developer console under the Test section an Alexa simulator is provided which works exactly the same as Alexa device. It is mainly for the testing purpose and links appropriately with the Alexa app. Any voice command queried on the Alexa simulator is comprehended with a card invocation on the Alexa app.

For this example, the skill **Bean** has been created. In the image it can be seen that Alexa actually interpreted the command **“Open bean”** (first line) and then it displayed the welcome message of the skill (second line). For the rest of this paper, every time we show an image like this one, it can be assumed that the grey dialogue is what Alexa interpreted and the blue dialogue is what Alexa showed as result of the voice command.

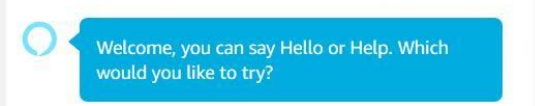
< Your Skills
Bean
Build
Code
Test
Distribution
Certification
Analytics

Skill testing is enabled in: Development
✓ Skill I/O
✓ Device Display
☐ Device Log

Alexa Simulator
Manual JSON
Voice & Tone

English (US)
Type or click and hold the mic

open bean



Welcome, you can say Hello or Help. Which would you like to try?

Skill I/O

JSON Input

```

1 {
2   "version": "1.0",
3   "session": {
4     "new": true,
5     "sessionId": "amzn1.echo-api.session.acfab9d2-222e-4f12-964.
6     "application": {
7       "applicationId": "amzn1.ask.skill.a48b8751-09aa-4912-8e
8     },
9     "user": {
10      "userId": "amzn1.ask.account.AE4RB4RSKY4DF5AJZ2HMLZ6YL3
11    }
12  },
  
```

6.2. Testing skills with similar phonemes

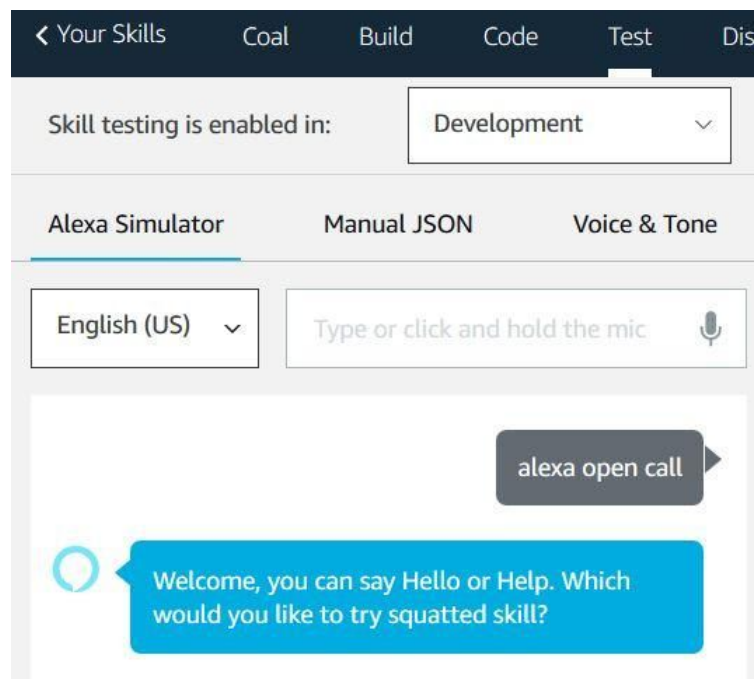
From the study conducted in **Skill Squatting Attacks on Amazon Alexa**, they were able to determine that there are certain words that Alexa is prone to confuse when the user sends a voice command. We have selected 10 skills from their list and we proceeded to replicate this test to compare if the results we obtained are similar to the ones that they showed in this study. The words chosen from the study and their respective success rate by invoking the squatted skill instead of the target skill are detailed below.

Target Skill	Squatted Skill	Success Rate
Coal	Call	100%
Rip	Rap	88.80%
Lull	Lol	81.90%
Calm	Com	67.90%
Dime	Time	65.20%
Sweeten	Sweeden	57.40%
Outshine	Outshyne	21.20%
Bean	Been	17.80%
Main	Maine	3.10%
Boil	Boyle	0%

As an example, we have selected as target skill **coal** and its corresponding squatted skill **call**. First, we proceed to create both skills as explained in previous paragraphs (Creating a skill in Alexa Developer Console).

	Call View Skill ID	English (US)	Custom	2019-04-14	● In Development
	Coal View Skill ID	English (US)	Custom	2019-04-14	● In Development

Once both skills have been created we proceeded to test them in the **Test** interface. As can be seen in the image for the skill **coal**, when we sent the voice command “Alexa open coal”, she wrongly activated the squatted skill **call**.



This process was carried out with the 10 selected skills and the results we obtained are the following:

Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	Success Rate
Coal	Call	Coal	Coal	Coal	Coal	Coal	Coal	12.50%
Rap	Rip	Rip	Rip	Rip	Rip	Rip	Rip	12.50%
Lol	Lol	Lull	Lull	Lull	Lull	Lol	Lull	75.00%
Com	Com	Call	Com	Com	Com	Call	Com	75.00%
Dime	Dime	Dime	Time	Time	Time	Dime	Time	50.00%
Sweden	Sweeten	Sweeten	Sweeten	Sweeten	Sweeten	Sweeten	Sweden	12.50%
Outshyne	Outshyne	Outshyne	Outshyne	Outshyne	Outshyne	Outshyne	Outshyne	100.00%
Bean	Bean	Bean	Bean	Been	Bean	Bean	Been	12.50%
Main	Main	Main	Main	Main	Main	Main	Main	0.00%
Boyle	Boyle	Boyle	Boy	Boyle	Boyle	Boyle	Boy	75.00%

In our tests we obtained different results in some of the selected skills. It is difficult to conclude why this happened, as we do not really know what arguments Alexa uses to select its response to the user's request. However, we noticed that some of factors are related to:

- Intonation.
- Pronunciation.
- User dialect.
- Speed while sending the voice command.

6.3. Testing an existing skill and creating a squatted one

The second test we performed to assess squatting attack on Alexa was to select as target skill, one that is currently available on the Alexa Skills site.

6.3.1. Selecting existing skills

In the following image it can be seen the target skills we selected for this test.



Comic Con Dates

by Tiago

☆☆☆☆☆ 0

Free to Enable

"Alexa, ask comic con dates
what is happening on august
fourth"



boil an egg

by charrawrz

☆☆☆☆☆ 0

Free to Enable

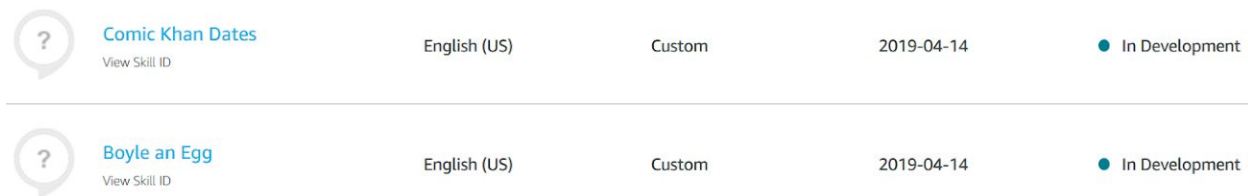
"Alexa, Open boil an egg"

The following table illustrates the target skill and squatted skill we have chosen for this test.

Target Skill	Squatted Skill
Boil and Egg	Boyle and Egg
Comic Con Dates	Comic Khan Dates

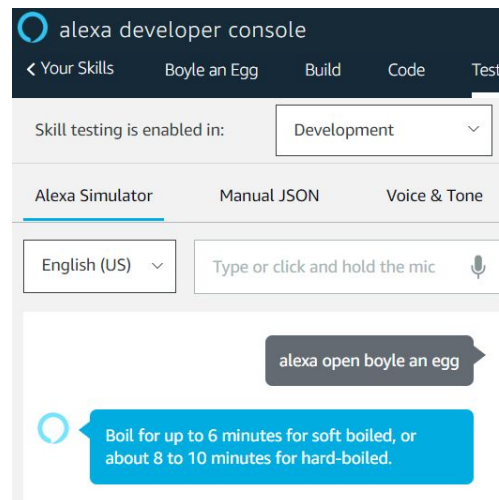
6.3.2. Creating squatted skills

The process of creating a new skill is the one that we have described in previous paragraphs. The following image shows the squatted skills we created for this test.

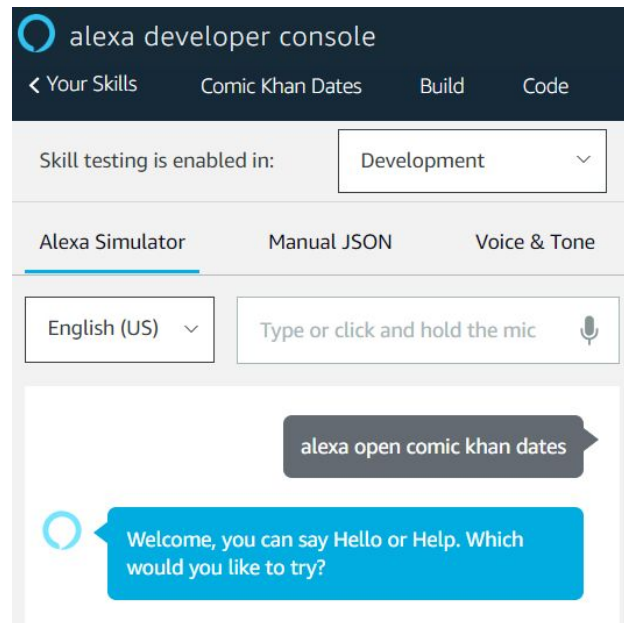


6.3.3. Our results

In the image shown below you can see that for the squatted skill **Boyle an egg**, even when Alexa understood that command the result it presented corresponds to the dialogue of the target skill. Of all the tests we performed we could never invoke squatted skill. Our success rate was 0%.



However, in the second image corresponding to the squatted skill **Comic Khan Dates**, when Alexa recognized the activation command of that skill, we were able to effectively invoke our malicious skill. The success rate in this case was 50%.



6.4. Replicating “Amex Echosquat”

The goal of this test is to create a squatted skill with the name **Am Express** and when a user wants to invoke the true **Amex** skill, available on the Amazon Skills download site, Alexa presents the squatted skill instead.

6.4.1. Creating the squatted skill

First, the **Am Express** skill is created in Amazon Developer Console.

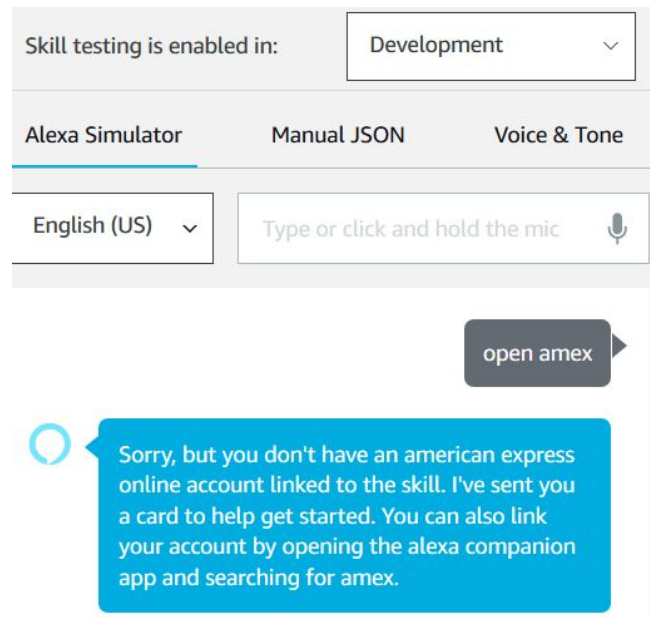
The skills in the amazon developer console can be made compatible with multiple languages for example English(CA), English(US), English(UK) and many more, which usually depends on the support Alexa wants to provide to its user. The skills owner decides in what all languages it wants to make the skill compatible with.

Alexa Skills

SKILL NAME	LANGUAGE	TYPE	MODIFIED	STATUS
 Am Express View Skill ID	English (UK), English (US), English (CA)	Custom	2019-04-14	● In Development

6.4.2. Testing the squatted skill

In the Test interface we proceeded to invoke the squatted skill with the command "Alexa, open amex". According to the original attack of Amex Echosquat, when the user performed this process the skill that was activated was the squatted one. However, in our test that was not the result. Therefore, we can assume that Amazon has corrected this error. In the following images it can be seen that the skill invoked correspond to the official American Express one.



It is certainly comprehended with genuine American Express card on the Alexa app.

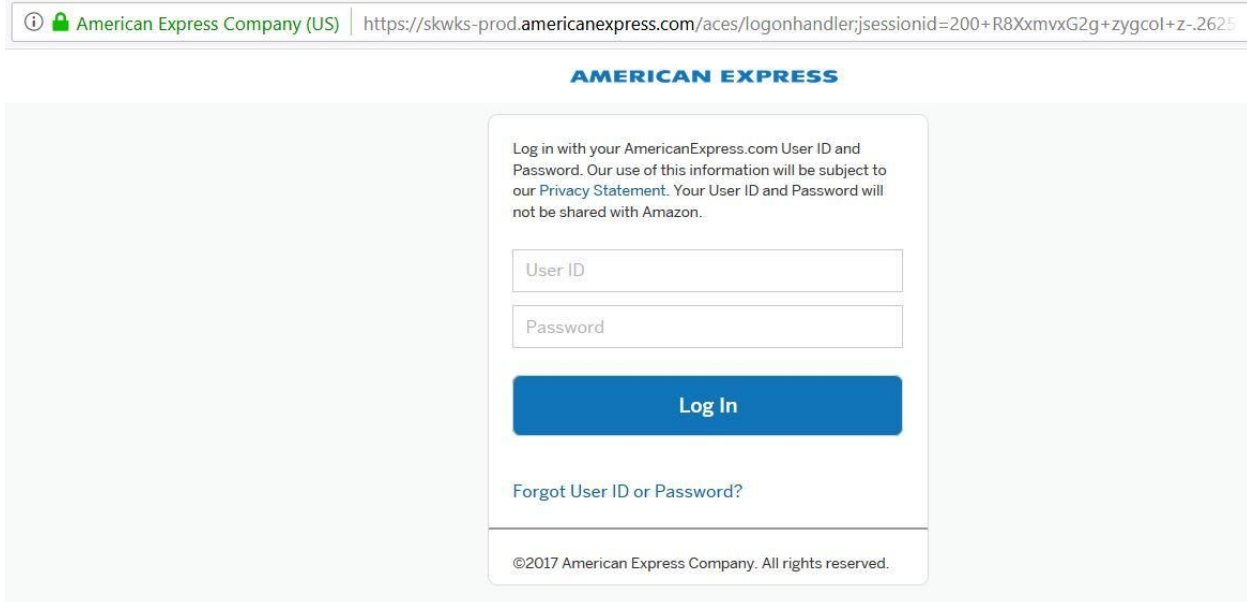
Amex - Account Setup

Amex

To get the most out of this Skill please link your account.

Link Account

As the user clicks on the “Link Account” button on the Amex - Account Setup card, the user lands on the Account set up web page where the user can enter its details and successfully add the American Express card details in the Alexa app.



6.5. Our Contribution to “Amex Echosquat”

In this project we wanted to investigate a little more about this attack and understand how Amazon Alexa works when presenting the results after receiving a voice request from the user.

6.5.1. Configuring the skill card and authentication server

First, we added the feature **Link Account** to our squatted skill **Am Express**. In the following image it can be seen how this card looks like.

Am Express - Account Setup

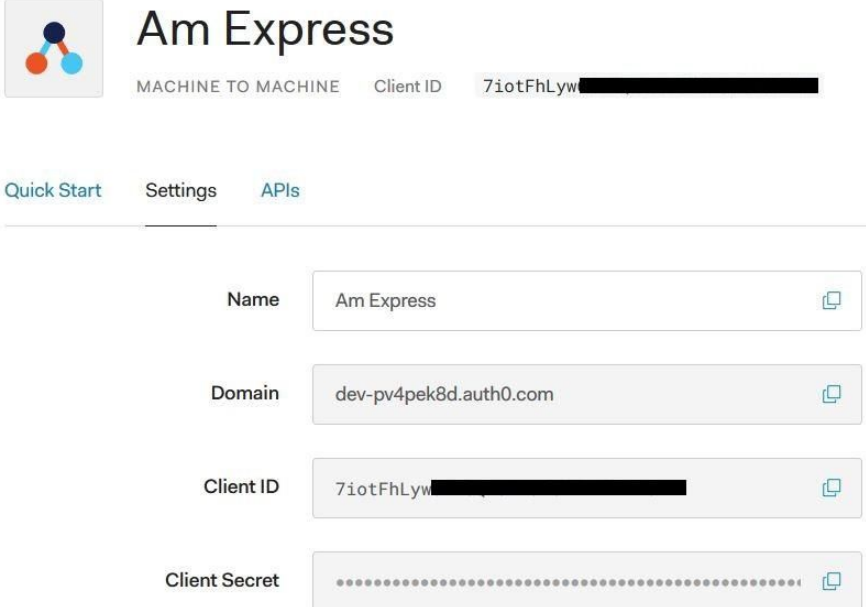
Am Express

To get the most out of this Skill please link your account.

Link Account

6.5.2. Hosting the login page and authentication service

When a skill requires to log a user into an account, Alexa Skill Kit requires a special configuration for validation and user authentication. For this project we have used the services of Auth0 (<https://auth0.com/>) to display the login interface of our squatted skill. The important parameters that must be added in Auth0 are the information referring to the Alexa skill that we are creating and additionally the HTML code for the login interface. The image below corresponds to the configuration on the Auth0 authentication server.



The screenshot shows the Auth0 console interface for a Machine to Machine application. At the top, there is a logo for 'Am Express' and the text 'MACHINE TO MACHINE Client ID 7iotFhLw[redacted]'. Below this, there are tabs for 'Quick Start', 'Settings', and 'APIs'. The 'Settings' tab is selected. The settings are displayed in a form with the following fields:

Field	Value
Name	Am Express
Domain	dev-pv4pek8d.auth0.com
Client ID	7iotFhLw[redacted]
Client Secret	[redacted]

Additionally, in the Alexa Development Console it is necessary to add the authentication server information that we are going to use for our skill. This information is provided by Auth0 after we configured the skill there. The following image details the parameters required to link both services (Alexa Developer Console and Auth0).

 Auth Code Grant

Authorization URI 

https://dev-pv4pek8d.auth0.com/authorize

Access Token URI 

https://dev-pv4pek8d.auth0.com/oauth/token

Account linked users will continue to use the previous URI until a user relinks their skill. [Learn more](#)

Client ID* 

7iotFhLw[REDACTED]

Client Secret* 

Client Authentication Scheme* 

HTTP Basic (Recommended) 

6.5.3. Testing the squatted skill

Summarizing what we've done until now. We have created a squatted skill with the name **Am Express**. We also created the corresponding card that will appear when the user access to the Alexa application after invoking our squatted skill. Additionally, we have configured the authentication service and login page for our skill that will have the following domain (<https://dev-pv4pek8d.auth0.com>) and that will look the same as the official American Express site.

Once all these steps were done correctly, we proceeded to test our squatted skill. From an Alexa device we send the invocation command "**Alexa, open Amex**".

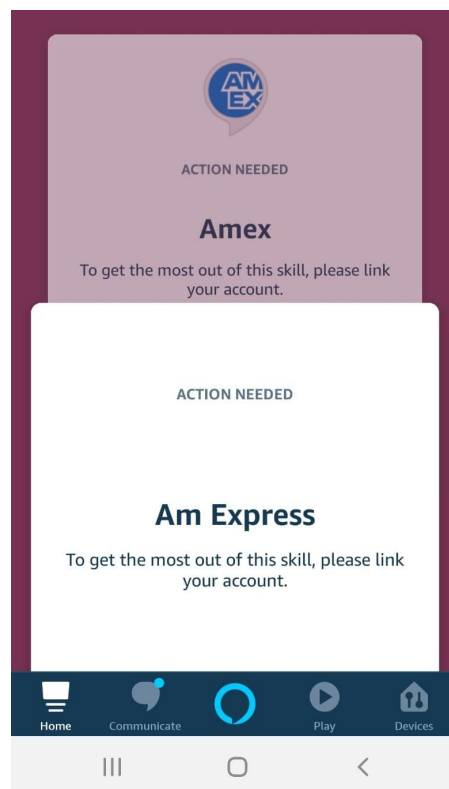
The squatted skill is compatible with multiple languages available and we tried to evaluate skill squatting attack on multiple languages. **We evaluated skill squatting attacks by changing the squatted skill language and the language on the Alexa device interchangeably.** We concluded the results with following scenario:

1. Squatted Skill Language - English(US), Alexa Device Language - English (US)
Result - Original American Express Skill was evoked
2. Squatted Skill Language - English(US), Alexa Device Language - English (CA)
Result - Squatted American Express Skill was evoked.
3. Squatted Skill Language - English(CA), Alexa Device Language - English (US)
Result - Original American Express Skill was evoked

4. Squatted Skill Language - English(CA), Alexa Device Language - English (CA)
Result - Squatted American Express Skill was evoked.

It can be clearly seen that the original american express skill works only when the Alexa device language is set to English(US) and the squatted skill works when English(CA) is used. The original american express skill is only available in English(US) and a user without knowing this can change the language setting and can fall prey to the squatted skill.

In the image below, it can be seen a comparison between the official skill card and the squatted skill card we have created, this is what the victim will see when he opens his Alexa app.



The image below shows the login page that the user will be redirected after accessing to the squatted skill from his Alexa app. From now on, it's up to the attacker to decide how he wants to continue this attack.

AMERICAN EXPRESS

Log in with your AmericanExpress.com User ID and Password. Our use of this information will be subject to our [Privacy Statement](#). Your User ID and Password will not be shared with Amazon.

Log In

[Forgot User ID or Password?](#)

©2017 American Express Company. All rights reserved.

What did we do differently from the “Amex Echosquat” attack explained in previous paragraphs?

Although Amazon Alexa has apparently fixed the vulnerability of Amex Echosquat, there is still a very serious problem, which is how Alexa decides what skill is triggered after a user sent a voice command. We have concluded that one of these arguments is related to the language in which the user's Alexa device is configured. For example, if a user has configured his Alexa speaker in English (US), he will obtain different results from the same voice command of another person who has his device in English (CA). Of course, the user does not know this information. Moreover, the user is not aware that there are some skills that only work in a specific language. This is a problem that an attacker can take advantage of to abuse Amazon Alexa skills.

In the following section we will explain two threat models that we have thought to take to exploit this flaw in the Alexa skill selection process.

7. Threat Models

7.1. Threat Model - 1

This threat model follows similar assumptions taken and environment implemented, to support the alternative threat models explained in the research paper. This particular threat model is totally different and original if compared to the threat models explained in the research paper.

This threat model explains the developments done in the section 6.5.3. in a novelistic manner. The threat model does not involve any victim but explains how an adversary can exploit a loophole to test this attack.

Notations:

1. Mimic Skill - A skill which mimics the exact same operation and functionality of the original skill, in this case we consider a mimic skill of the original American Express Skill. We will refer to it as "mimic skill"

Assumptions :

1. Alexa Developer Console and the Alexa app user account are configured with the same credentials
2. The original skill is not available in the target language.

Steps:

1. An adversary develops a mimic skill in English(CA)
2. The adversary changes the language to English(CA) in the Alexa Device app settings
3. The adversary triggers the mimic skill with the voice command "Alexa, open Amex"
4. The original skill is not available in the selected language, henceforth the mimic skill will react to the voice command.
5. The mimic skill will spawn a card in the Alexa app, clicking on which, will land the adversary on the fake American Express log-in page.

7.2. Threat Model - 2

We came up with another threat model which looks more realistic and which can be implemented in a real world case scenario.

Notations:

1. Victim : A victim owns an Alexa device and has his Alexa app installed in the phone. Both the entities will share the same amazon login credentials.
2. Mimic Skill - A skill which mimics the exact same operation and functionality of the original skill, in this case we consider a mimic skill of the original American Express Skill. We will refer to it as "mimic skill".

Assumptions:

1. The adversary somehow managed to get the amazon credentials of the victim.
2. The original skill is not available in the target language i.e. English(CA).

Steps:

1. Alexa device language settings can be configured or changed in the following manner:
 - Alexa device by default is set in English(CA)
 - Adversary can change the language setting of the Alexa device by logging in to the Alexa app (web / smart device)
 - The victim itself changes the language to English(CA)
2. The adversary opens up an Amazon developer account with the same compromised credentials of the victim.
3. The Amazon developer setup does not ask any verification procedure to validate the user.
4. The adversary build a mimic skill on the Amazon developer console.
5. The mimic skill's card can be spawned in the victim's Alexa app in the following manner:
 - The adversary triggers the card in the victim's alexa app through the Amazon developer console Alexa simulator with the voice command "Alexa, open Amex" (Since the Amazon developer console and the Alexa device / Alexa app are on the same account, the mimic skill when spawned through the Amazon developer console Alexa simulator, it spawns a card in the Alexa app)
 - The victim triggers the card with voice command "Alexa, open Amex" (The victim does not know if the original skill is available in the target language and hence ends up calling the mimic skill which looks exactly the same compared to the original skill).

8. Conclusions

- The speech recognition system of Alexa is still ambiguous and has semantic errors.
- The voice commands understood by Alexa depends on the demographics and accent of the speakers
- There is no verification process when creating a new Amazon Developer Account. The information required to setup this new developer account is shared in the screenshots below.

it just demands basic address and contact information which can be faked by the attacker.

1. Profile Information2. App Distribution Agreement3. Payments

* indicates a required field.

Country/Region *	<div>United States</div>
First name *	<div></div>
Last name *	<div></div>
Email address *	<div></div>
Phone number * e.g. 212-555-1212, +44 0161 715 3369	<div></div>
Fax number	<div></div>
Developer name or company name * Displayed on your apps at Amazon.com	<div></div>
Developer description Maximum characters 4000, Remaining: 4000	<div></div>
Address 1 *	<div></div>
Address 2	<div></div>
City *	<div></div>
State *	<div>Please select</div>
Zip code/Postal code *	<div></div>
Customer support email address	<div></div>
Customer support phone	<div></div>
Customer support website	<div></div>

Cancel

Save and Continue

1. Profile Information 2. App Distribution Agreement 3. Payments

English | 中文 (Chinese)* | 日本語 (Japanese)*

- Structure of Agreement.** This agreement (the "**Agreement**") includes the body of the agreement below, all schedules to this agreement ("**Schedules**"), and all terms, rules and policies that we make available for participating in this program, including on our developer portal (together, the "**Program Policies**"). However, the terms in each Schedule only apply to you if you engage in the activity or use the Program Materials (defined in Section 3) to which the Schedule applies (for instance, the terms of the Distribution Schedule only apply to you if you submit a covered product to us to sell, distribute, or promote). Please carefully read the Agreement before clicking to accept it.
- Our Program.** Our program (the "**Program**") allows end users to purchase, download, and access software applications, games, and other digital products and services (for instance, the Amazon Alexa voice service (the "**Alexa Service**")) and allows developers to enable access to Amazon products and services in their Apps and Devices. "**Apps**" are software applications, games, and other digital products that you submit to us for sale, distribution, or promotion through the Program, or with which you use any Program Materials, together with their enhancements, upgrades, updates, bug fixes, new versions and other modifications and amendments. "**Devices**" are devices and device components that use any Program Materials. "**Content**" means your Apps, all content, ads, services, technology, data and other digital materials included in or made available through your Apps or Devices, and all Product Information (defined in Section 7).
- Program Materials.** We may make available certain software, software development kits, libraries, application programming interfaces, services, documentation, sample code, and related materials and information for use in connection with the Program (collectively, the "**Program Materials**"). If you use any Program Materials, you are subject to and agree to comply with our Program Materials License Agreement (the "**Program Materials License**"), located at <https://developer.amazon.com/support/legal/pml>. Your use of certain Program Materials is also subject to the additional terms in any Schedules that apply to those Program Materials (for instance, your use of the Program Materials that we make available for sale of In-App Products is subject to the terms of the Distribution Schedule and the In-App Products Schedule). You are solely responsible for ensuring your Content and Devices function properly with any Program Materials you use, including any future updated or modified versions of those Program Materials. To the extent there is any conflict between this Agreement and the Program Materials License, the Program Materials License will govern with respect to your use of the Program Materials.
- Compliance with Laws; Privacy and Security Obligations.** You, your Content, and your Devices must comply with all applicable laws, rules, regulations, orders, and other requirements of governmental agencies ("**Laws**"). In addition, if you (or any third-party plug-in or service provider you use) have access to any name, password, other login information, or personally identifiable information or personal data of any end user based on any use of or interaction with your Content or Devices, you will (i) provide legally adequate privacy notices to such end user, (ii) obtain any necessary consent from the end user for the collection, use, transfer, and storage of the information, (iii) use and authorize others to access and use the information only for the purposes permitted by the end user, and (iv) ensure the information is collected, used, transferred, and stored in accordance

Print Agreement

***Please note:** All non-English translations are provided for informational purposes only and are non-binding. By clicking "Accept and Continue" below, you agree to the English language version of the Mobile App Distribution Agreement.

Cancel

Accept and Continue

1. Profile Information 2. App Distribution Agreement 3. Payments

⚠ Second factor verification using Mobile SMS will soon be required to update payment information. Please update your mobile number within [My Account](#).

* Indicates a required field.

Do you plan to monetize your digital content, such as charging for apps or games or selling in-app items or in-game items, or by receiving cash rewards for your skills? *

☒ No ☐ Yes

Do you plan to monetize apps by displaying ads from the Amazon Mobile Ad Network ? *

☒ No ☐ Yes

Note: You may still monetize later if you select "No" by entering payment and tax information from the Settings menu.

Cancel

Save and Continue

- We tried to implement the threat model with different languages such as English(UK) but similar results were not seen. Null response from Alexa was noted.
- Skill Squatting on Existing Skills is still a threat as can be seen from our results.
- It is still not clear on what basis Alexa allows a skill to interact with the user. It can either be:
 - Language selection made in the Alexa device setup
 - Language or Location setup done in the Amazon account.
 - The actual location of the user or the country a user is in.

This argument is supported with the warning message, Alexa displays in the Alexa app, while attempting to change the language.

Change Alexa's Language

The selected language does not match your Amazon Account and is not fully supported in your country. If you make this change, some Alexa capabilities, skills, music and content may be unavailable.

CANCEL OK

9. References

- Research Paper
<https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kumar.pdf>
- Amazon Developer Console
<https://developer.amazon.com/alexa-skills-kit>
- Research Paper Presentation
<https://www.usenix.org/conference/usenixsecurity18/presentation/kumar>
- American Express Skill Squatting
<https://arstechnica.com/information-technology/2018/08/researchers-show-alexa-skill-squatting-could-hijack-voice-commands/>
- Blog
<https://www.csoonline.com/article/3273929/voice-squatting-attacks-hacks-turn-amazon-alexa-google-home-into-secret-eavesdroppers.html>
- Hosting and Authentication Service
<https://auth0.com/>
- Alexa SetUp and Tutorial
<https://www.codecademy.com/learn/learn-alexa>
- Alexa Skill Store
<https://www.amazon.com/alexa-skills/b?ie=UTF8&node=13727921011>
- Lau, Josephine & Zimmerman, Benjamin & Schaub, Florian. (2018). Alexa, Are You Listening?: Privacy Perceptions, Concerns and Privacy-seeking Behaviors with Smart Speakers. Proceedings of the ACM on Human-Computer Interaction. 2. 1-31. 10.1145/3274371.
- J. Bugeja, A. Jacobsson, and P. Davidsson. 2016. On Privacy and Security Challenges in Smart Connected Homes. In 2016 European Intelligence and Security Informatics Conference (EISIC). 172–175. <https://doi.org/10.1109/EISIC.2016.044>