KING'S College LONDON

University of London

6CCS3PRJ 3rd Year Individual Project

Security Assessment of Amazon Alexa service running on an Echo Dot (2nd generation)

Final Project Report

Author: Luka Kralj
Programme: MSci Computer Science, Year 3
Supervisor: Dr Jose M. Such
Student ID: 1707488

Department of Informatics
King's College London

23 April 2020

Abstract

Voice-activated virtual assistants, such as Amazon Alexa, Apple's Siri and Google Assistant, are making their way into more and more households around the world as they are easy to use and provide a lot of functionality to help users on their daily basis – all while maintaining a friendly and non-robotic charisma. However, the growing awareness of their "always-listening" nature is instilling distrust in people making them shy away from using a virtual assistant. I focused on one of the most widely used smart assistants – Amazon Alexa. The aim of this report is to provide an overall security assessment of Alexa by considering all of its components and subsystems. I enumerate and extend all known attacks on Alexa as well as present new ways of exploiting its vulnerabilities. Namely, I show how users can be tricked into revealing their personal information to a seemingly innocent skill and how their credentials can be phished through a spoofed login page. I also show how all Echo and Echo Dot devices sold from 2nd hand can be compromised to obtain users' private conversations. I performed a structured risk assessment of all attacks to show that the "always-listening" nature of Alexa is not by far the biggest privacy concern. Rather, I find that the root of many attacks lies in a flawed vetting process and, as unfortunately is the case in many software systems, the users themselves.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Luka Kralj

23 April 2020

Word count: 21004

Acknowledgements

I would like to thank my supervisor Dr Jose M. Such for providing useful comments and his guidance throughout this project.

I would also like to thank Public Scholarship, Development, Disability and Maintenance Fund of the Republic of Slovenia (sl. Javni štipendijski, razvojni, invalidski in preživninski sklad Republike Slovenije) for sponsoring my studies.

Contents

List of Tables

List of Figures

1 Introduction

Smart assistants now come in all shapes and sizes – as a software only agent like Apple's Siri and Microsoft's Cortana, embedded in smart speakers like MyCroft and Google Assistant, and even as a little pet robot like Anki's Vector. However, in this report I will focus on by far the most popular smart assistant, Amazon Alexa.

Amazon Alexa was first launched in September 2014 [7]. Its popularity started rapidly increasing in 2017 [8] and today, it holds over 60% of the US market and almost 40% of the global market share [9]. Since Alexa can run on many different devices, it has successfully made its way into millions of homes, offices and even hotel rooms [10]. With the widespread use of Alexa, users are becoming increasingly worried about its security and privacy. Hence, there is an increasing need for a formal verification of the service's security measures, which would help people trust in these devices and their providers.

Thus, this report focuses on identifying and analysing security flaws of the Amazon Alexa service that is running on an Echo Dot (2^{nd} generation). I will attempt to enumerate, extend and evaluate a range of attacks that are still possible at the time of writing as well as present novel findings and attacks. I will follow the same risk rating methodology for all the attacks to gain a clearer picture of which part of the Alexa ecosystem is most vulnerable.

I will mostly take a look at how a third-party adversary can exploit Alexa and not at how exactly Amazon can exploit the users' data as this would require an insight into Amazon's proprietary internal architecture which I do not have. I will, however, make some observations, that can be deduced indirectly from other evidence, about this architecture.

1.1 Contribution

The main contributions of my work can be summarised as follows:

- I extended some of the known attacks to further explore their potential.
- I show how an attacker can very easily spoof Amazon login page and trick Alexa users to phish their credentials.
- I show how all Echo and Echo Dot devices sold from 2^{nd} hand can be compromised with a small and cheap listening device.

- I uniformly assessed all known and newly discovered attacks to provide a structured overall security rating of Alexa.
- I suggest a range of mitigation techniques for a range of vulnerabilities.

1.2 Report Structure

In Chapter 2, I will first present the basics of Alexa and its ecosystem and provide an analysis of relevant literature. Then, in Chapter 3, I will state the ethical consideration I was adhering to throughout the project and explain some issues I had on the way. Then, I will present the methodologies used in Chapter 4 which will be followed by the first step of penetration testing – reconnaissance (in Chapter 5). After describing the details of the ecosystem, I list the security requirements (in Chapter 6) that will have to be satisfied in order to deem Alexa as a secure system. In Chapters 7 and 8, I describe all the attacks and then use that to provide a structured evaluation. I provide the possible procedures that would mitigate or prevent such attacks in Chapter 9 and then conclude the report and suggest some future research directions in Chapter 10.

1.3 Dictionary

To make the report more concise and easier to read, I will be using the following abbreviations:

- **Alexa**: Amazon Alexa service, a virtual assistant developed by Amazon.
- **Cloud**: Amazon's servers where the "brain" of Alexa resides. This is where all text processing and response generation takes place.
- **Dot**: Amazon Echo Dot (2^{nd} generation), a smart speaker developed by Amazon with Alexa service built-in that I will be using in my tests.
- **App**: Amazon Alexa mobile application (the application is available for Android and iOS, but I will be using the Android version).
- **User**: The person who is using Alexa and has no malicious intent, i.e. a normal everyday user of Alexa.

2 Background

Alexa is one of the most popular virtual assistants currently on the market [9] and has been integrated into various products, such as Amazon Echo devices, Amazon Loop (a ring) and Amazon Frames (glasses) [11]. Regardless of the type of the device, the idea is that the user uses their voice to tell Alexa what to do or ask a question to which Alexa replies by playing a response back to the user. I will describe how this works in the sections below.

Voice-activated assistants are always listening for a wake-up word, which means that they need to continuously process short recordings to see if they match the specific wake-up word. Since increasingly more people are learning about this "always-listening" nature of Amazon's smart speakers, as well as the amount of data Amazon is gathering about its users, there is a raising concern about how this data is actually used and if Amazon devices can somehow be breached in order to obtain sensitive personal information. Another concern is that once an attacker gains control over Alexa, they also gain control over all other connected devices, such as smart light bulbs and smart plugs as well as the more critical ones, such as smart locks.

This project will try to address these concerns by evaluating security risks of known exploits, attempting to uncover new ones and assessing their feasibility. In my tests, I will try to obtain the information, that I believe is most valuable for adversaries:

- **Amazon account details**: Such as credentials, addresses, credit card details, etc.
- **Credentials of linked accounts**: In order to use them, some skills require users to link their other accounts (e.g. an Uber account) to Alexa.
- **Private conversations**: In some cases, an adversary could find users' private conversations valuable (e.g. business secrets).
- **Usage statistics**: Gaining an insight into the usage patterns could help an adversary to decide what is the best time, for example, to break in or conduct other attacks.

2.1 Amazon Alexa

Amazon Alexa is a cloud-based voice-activated virtual assistant. Alexa can, either through a smart speaker, the App or a web interface record a user's voice command and send it to the cloud-based voice processing. It then receives a computer-generated recording of a response which is played back to the user.

When Alexa first came out, it could only understand some simple commands, e.g. to play music, answer simple questions or add items to the lists. However, with the development of Alexa skills (see Section 2.2.4), Alexa became smarter, meaning that the users can complete more complex tasks by using just their voice. Alexa's abilities increased rapidly since the introduction of Alexa Skills Kit in 2015 [12], which enabled developers all around the world to create new skills for Alexa. All the Skills Stores across various countries now contain more than 100,000 skills in total [13].

Today, users can use Alexa for playing music, completing online purchases, calling people and sending messages. Furthermore, Alexa also supports some more advanced features [7]. For example:

- **Voice recognition** enables Alexa to differentiate between multiple users in the household and offer them tailored responses.
- **Routines** enable users to trigger multiple tasks with just one command.
- **Memory** allows Alexa to remember what users ask her to remember so they can retrieve this information later.

2.2 Alexa Ecosystem

Alexa ecosystem consists of the following components:

- **Alexa-enabled device**: This is commonly one of Amazon devices, but it can also be the App or any other product with Alexa integrated into it (it does not need to be an Amazon product). Such device records a user's utterance and sends it to the Alexa Cloud. It then waits for a response and plays it back to the user.

- **Alexa Cloud**: This is where the Alexa's "brain" is located. It transcribes voice commands into text using machine learning and then infers the user's intentions using natural language processing. Then, the relevant parts of the transcription are either sent to Amazon's skill servers or to a third-party skill server, depending on where the skill is hosted.

- **Skill servers**: These servers (either Amazon's or third-party) receive the relevant parts of users' utterances with some meta data and can then request further information from external APIs. All of this is combined into a text response and sent back to the Cloud.
- **Skills**: Skills are applications that produce a response based on the command given. They can be either native skills, designed by Amazon, or third-party skills designed by other developers.
- **User's Amazon account**: In order to use Alexa, all users need an active Amazon account, which is usually the same one as used for online shopping on the Amazon website. This account is used to register Alexa devices. Users can access their accounts through the App or through a website where they can manage their enabled skills, skills' permissions, delete old voice recordings and similar.

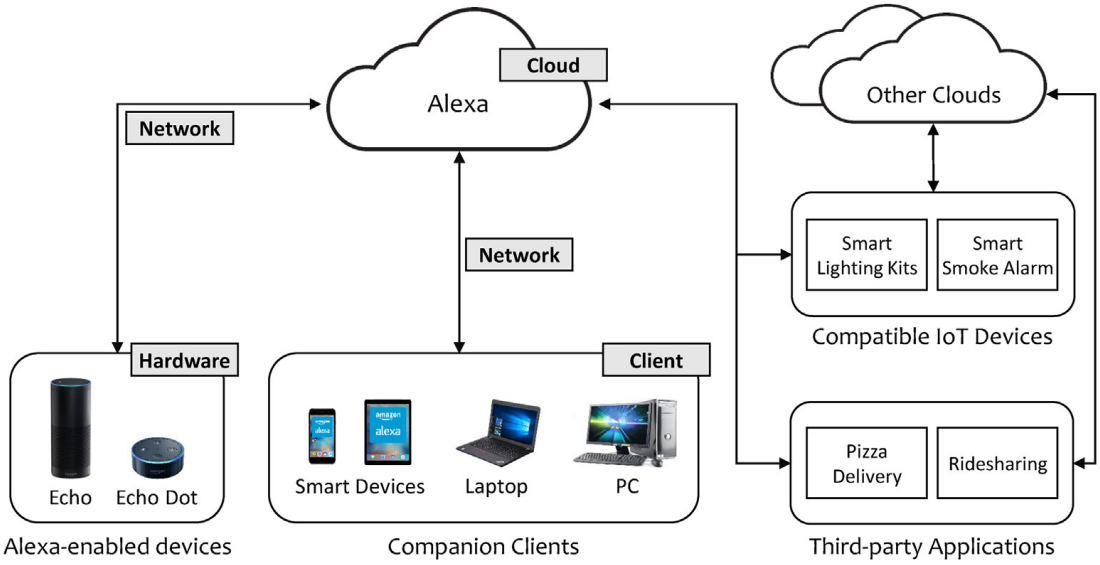

Figure 2.1: Alexa ecosystem (taken from [1]).

2.2.1 Amazon Echo and Amazon Echo Dot

Most widely used Alexa devices are Amazon Echo and Amazon Echo Dot devices that come with Alexa built in. They use omnidirectional microphones that can pick up a user's voice from anywhere in the room [14]. They are both smart speakers, which means they are always listening for the so-called *wake-up word* (or *wake-word*). By default, this is set to be "Alexa", but it can also be changed to "Amazon", "Echo" or "Computer" [10]. Only once the device recognises the wake-up word it starts recording and communicating with the Cloud so only the

wake-up word recognition is executed on the device itself. All other commands are processed in the Cloud [15].

In my testing, I will be using an Amazon Echo Dot (2^{nd} generation) which is smaller and cheaper compared to an Amazon Echo device. Since most of the processing in Alexa happens in the Cloud, there are only a few attacks possible specifically on the Dot. Most of large scale attacks can be conducted irrespective of the device a target user is using.

2.2.2 Alexa Cloud

Alexa resides on Amazon servers and is part of Amazon Voice Services (AVS) which I will refer to simply as the Alexa Cloud. Its main tasks are to transcribe the voice recordings into text, infer the users' intention and send the relevant parts of commands to the correct skill server for further processing. Voice transcription is not a trivial process because of background noise, different accents, nuances and various ways of forming a sentence with the same meaning in spoken language.

Thus, Amazon first performs some acoustic transformation of the recoding to extract only the vital user utterance. Then, the recording passes through Automatic Speech Recognition (ASR) which uses machine learning to convert the audio into text [15]. Alexa uses a one-to-one mapping when transcribing commands, meaning it does not perform any post processing (e.g. removing redundant phrases such as any unnecessary "the" and "um"), which could improve its accuracy [16].

Next, this text is processed by Natural Language Understanding (NLU) which infers the user's intention [15]. The *Intent Manager* recognises which skill the user wants to invoke (invocation might be implicit) and maps the relevant parts of the command to the *slots* (see Section 2.2.4). This is achieved by matching the user's utterance to the utterance patterns specified by the skill [17]. If the intention cannot be inferred, the Intent Manager requests more information from the user.

Once the command is understood, it is sent to the correct skill server for further processing which, then, returns a response in plaintext or in form of Speech Synthesis Markup Language which can be understood by a text-to-speech (TTS) system [9]. The text response is then passed through the Natural Language Generation (NLG) algorithm [15] where it is converted into a recording of a spoken language which is played back to the user.

2.2.3 Skill Servers

There are two types of skill servers: developers can choose to host their skills on either Amazon servers or on their own server. If the skill is hosted on an Amazon server, it is called a *lambda function* which is created as part of Amazon Web Services (AWS) platform, but if a developer decides to host a skill on their own server then they must support HTTP over SSL/TLS connection and implement some other security verifications [18].

The Cloud only sends the most basic meta data to these servers and some details about the current skill session (e.g. which *slots* have already been elicited and their values).

2.2.4 Skills

A *skill* is an application that processes the meta data obtained from the user's utterance and produces a response. Each skill is identified by its *invocation name* and can perform different actions, called *intents* [17]. Each intent has a set of *utterance patterns* (or *sample utterances*) which the Cloud will try to match. These patterns can contain some placeholders, called *slots* [17]. These are arguments (optional or required) that give more information about the intent. Some intents are created by default, for example a *StopIntent* to stop the skill's execution. These usually have a predefined set of utterance patterns created by Amazon.

Native skills are developed by Amazon, but a lot of Alexa's capabilities come from third-party skills. Anyone can build an Alexa skill using Amazon Skills Kit. The advantage of it is that developers do not even need an Alexa device to test their skill. Since most of the processing is done in the Cloud, they can test their skill through a web-based interface and still be sure it will work correctly on all Alexa devices.

2.3 Literature Review

There is a number of resources that focus on a more overall review of Alexa and other smart assistants, but there are even more studies that only focus on a specific aspect of these assistants. Different research teams use different methodologies and evaluation standards, so this report will also address this issue by uniformly presenting and assessing each of the vulnerabilities found.

Papers [15, 19, 20] describe a large variety of attacks and vulnerabilities for which they also provide possible mitigation techniques. Paper [19] only focuses on Alexa, while the other two also describe attacks found in other voice assistants which I will analyse to see if they apply

to Alexa as well. They demonstrated PIN brute force and indecipherable sound attacks and also showed that voice SQL injection and packet replay attacks are not possible. Paper [15] described many vulnerabilities on Alexa and other similar smart assistants and pointed out which studies focused on each of them. Thus, it was also a very good source for locating other papers. They also provided an extensive list of countermeasures which will be summarised in Chapter 9. Paper [20] compares Amazon Echo, Google Home and some other lesser-known smart assistants and explores if attacks, such as rooting smart speakers, BlueBorne and Dolphin attacks, are possible on any of them. Most of these attacks are not possible on Alexa; I described them in some more detail in Section 7.17.

Paper [21] compares basic functionality of popular voice assistants, such as Alexa, Siri, and Cortana. Interestingly, they suggested that such assistants can benefit dementia sufferers because they can repeatedly answer the same questions without losing patience. However, they concluded that many of these assistants will need to improve their security measures before they can be used for anything that requires confidentiality.

Paper [10] compared security of Alexa and Google Assistant and found that the biggest "threat" are the other users who can interact with a smart speaker.

Other papers mainly focus on one of the research topics below. I only briefly described papers that talk about specific attacks. I will give a detailed description of their findings in Chapter 7 so that all information about attacks is condensed in the same part of the report.

Network

By recording the network traffic, researchers [22] found that in a 21-day experiment Alexa recorded 33 conversations that were not meant directly at it, about 35% of which were human conversations.

A recent research [6], compared the vetting processes of Alexa and Google Assistant and found that both vetting processes are insufficient and cannot ensure complete security since not all developers' mistakes can be detected. They also presented a new type of attack – a cloud spoofing attack. This attack and more detailed findings will be presented in Section 7.14.

Privacy

There are several papers that focus specifically on how Amazon is ensuring Alexa's privacy and how this can be improved. Study [12] finds that 75% of all skills do not have a privacy policy, and that even more skills have an improper one. Paper [23] finds that some skills in Google

Assistant[1] acquire personal user information through a voice interface – I will be checking if this can happen on Alexa, too. Paper [24] explores various privacy issues, such as analysis of usage patterns and how a possibility of Alexa accidentally recording some conversations implies that users do not have a complete control over their voice data. Researchers in [25] propose acoustic tagging, a special sound signal, as a mean of preventing accidental recordings; a voice recording with an acoustic tag would be discarded by the Cloud.

User Perceptions

Understanding how end users interact with Alexa and other smart assistants could help highlight the vulnerabilities that would otherwise be difficult to discover in a lab. Hence, a lot of reports focus on analysing how people use these devices.

Papers [26, 27, 28] indicate that most users are not aware of how smart speakers function and of the privacy controls they offer. Researchers in [26] further found that most people do not think their recordings contain sensitive information, but they still find the current audio logs management and other privacy features unacceptable. Paper [27] finds that most users are not aware that skills can be developed by third parties, and that regular users are more likely to assume such a skill is a native Alexa functionality. Researchers in [28] also found that smart speakers are often placed in central locations in the users' homes, and that the current security controls are not aligned with the users' needs; additionally, they found that a lot of users do not perceive audio logs as a sufficient privacy control. Study [29] focused on finding the reasons why the usage of smart assistants is spreading so quickly. They find that these assistants are mostly used for utilitarian purposes and only rarely are they used for entertainment. Study [30] finds that users perceive smart assistants most useful when the dialogues are shorter and when the audio answers are supported with appropriate visual information on the devices' screens.

Voice Interface

Various papers explore how the voice interface can be exploited. In particular, they proposed attacks, such as skill squatting, skill masquerading, missense attacks, and similar. These attacks seem to be the most likely candidates for performing large scale attacks and there are already dozens of skills on the market that could be vulnerable to such attacks. Skill squatting targets the speech-to-text translation and skill invocation algorithm [9, 31, 32]. Skill masquerading involves imitating a smart assistant or another skill [32]. Missense attack can target either

[1]Skills in Google Assistant are properly called Actions.

speech-to-text transcription or speech understanding; researchers found that sometimes an utterance is correctly transcribed, but the matched command does not correspond to the transcription [2]. An extreme case of this is sending a distorted command that is indecipherable to a human, but is understood by smart assistants without a problem [3].

Additionally, researchers in [33] explored what data can be extracted from Alexa through the voice interface; amongst others they find that it is possible to extract some of user's health information through the FitBit skill. This is possible mainly due to poor authentication principles. Alexa's authentication was found to be vulnerable by [34] as well, but they also describe how it could be improved by detecting the user's presence with a Wi-Fi signal.

Researchers in [35] analysed the correctness of answers of four smart assistants and how natural they feel to the users and found that Alexa and Google Assistant win by far against other tested assistants. Researchers in [36] found that smart speakers tend to be significantly less reliable for other languages (compared to English). By comparing comprehension of common medications, researchers found that Google Assistant shows a significant performance lead, compared to Siri in the second place and Alexa only being third for understanding medication names [16]. They suggest that there is still room for improvement for all three smart assistants.

Digital Forensics

Since Alexa is making its way into more and more households, the probability of it "witnessing" a crime is increasing. Hence, researchers have started taking a closer look at how an Alexa-enabled device could be used to obtain crucial evidence [1, 37]. Paper [38] also analyses possibilities of such forensic searches and other Alexa's privacy issues in relation to US law. Searching for digital evidence often considers non-standard methods for accessing users accounts (provided they have correct credentials), which might reveal more information about a user than the normal web interface that users normally interact with.

Microphone Exploits

There exist two interesting attacks that exploit the smart speakers' microphones directly. The first one is called a Dolphin attack which uses inaudible sound signals to trigger a command [4, 39, 40] and the second one uses strong light beams, usually lasers. The researchers that discovered this attack called it Light Commands [5, 41, 42].

3 Legal, Social, Ethical and Professional Issues

3.1 Ethical Considerations

All the testing mentioned in this report was conducted on my own Echo Dot device and my mobile phone. They were both connected to my local hotspot, so none of the penetration testing traffic, such as scanning, left my local network. The only requests that were sent outside the network are, naturally, the voice interactions with Alexa, which includes interactions with my test skills. Even when these voice interactions were designed to test the capabilities of Amazon voice recognition software, they did not cause any harm to Amazon. All my test skills were only tested in development mode on my Amazon account and I did not attempt to publish them to the Skills Store as this could be seen as an attempt to trick Amazon. Additionally, I was the only user of Alexa during the testing, so I was only working with my own data.

3.2 BCS Code of Conduct

Throughout the project I followed The British Computer Society (BCS) Code of Conduct to the best of my abilities. Even though I was conducting this security assessment on Amazon's systems, I only used my own devices in a way that it did not cause any damage to Amazon. I also spent a lot of time at the start of the project learning about penetration testing before conducting any actual tests in order to produce a reliable security assessment. In addition, I carefully planned each test so that it would not violate any of the BCS Code of Conduct principles.

3.3 Professional Issues

Since this was my first penetration testing task of an overall software system, I spent a good amount of time at the beginning of the project learning and familiarising myself with various ethical hacking techniques and principles. Subsequently, this left me with a bit less time that

I would wish for for conducting all the known attacks in order to explore them further and to see whether Amazon has already put any mitigation in place for them. Thus, I mainly focused on extending attacks that, I believe, have not had their potential fully explored yet. For example, I did not recreate a skill squatting attack as this attack has been shown numerous times in literature. Instead, I only verified that this attack is still possible by looking at the skills available in the Skills Store. On the other hand, I did implement skills for vishing and phishing as these attack vectors have not been fully explored yet in literature. I believe this was a reasonable decision as the main goal was to provide an evaluation of known attacks, rather than just recreating them, and then attempt to discover new ones.

The other issue I had was the budget. Unfortunately, I could not extend attacks such as the Dolphin and Light Commands attacks because they require very expensive equipment.

4 Specification

The goal of this project was to assess the feasibility of known attacks on Alexa, including looking at particular resources each attack requires, attempt to discover new attacks by thoroughly following the standard penetration testing principles, and then conduct a risk assessment for the assistant based on these attacks. Additionally, I spent some time researching vulnerabilities that have been discovered in other smart assistants, for example, in Google Assistant, to familiarise myself with how other providers are solving similar security issues. I will mention such techniques used by other providers whenever they could be a good addition for Alexa, as well.

I hope that this report will contribute to the ongoing research about the security of smart assistants.

I will be using two methodologies: one to guide my testing and one to use for consistent security evaluation.

4.1 Testing Methodology

I will be following the CREST-PTES testing framework [43] since it is more applicable in my case. Unlike the OWASP testing framework, where phases 1 through 4 are carried out *before* or *during* the development of the system [44], CREST-PTES framework consists of three main phases[1] that can be carried out on an existing system:

- **Phase 1 – Preparation**: This step comprises of identification of target environments, defining the purpose of the penetration testing and producing a requirements specification. I presented the general working of Alexa in Sections 2.1 and 2.2. However, a more in-depth reconnaissance is described in Chapter 5. I combined all this information into a list of security requirements presented in Chapter 6.

- **Phase 2 – Testing**: This includes conducting the tests systematically and identifying and exploiting vulnerabilities. I gave a description of the attacks found, including their set-up and the equipment needed, in Chapter 7.

[1] These phases have some other steps in addition to the ones I specifies, but they are less applicable in my case.

- **Phase 3 – Follow-up**: This contains remediating weaknesses and providing a detailed report. I provide attacks evaluation in Chapter 8 and describe possible mitigation techniques in Chapter 9.

4.2 Risk Assessment Methodology

After attacks and vulnerabilities are identified, I will use the OWASP Risk Rating Methodology [45], to assess each likelihood and impact factor of each attack and vulnerability and then combine these risk estimates into a final rating. Each risk rating is calculated as `Risk = Likelihood * Impact`.

Likelihood estimate is an average of the following two groups of factors:

- Threat agent factors:
  - Skill level
  - Motive
  - Opportunity
  - Size (of a group of threat agents)
- Vulnerability factors:
  - Ease of discovery
  - Ease of exploit
  - Awareness
  - Intrusion detection

Impact estimate is an average of the following two groups of factors:

- Technical impact:
  - Loss of confidentiality
  - Loss of integrity
  - Loss of availability
  - Loss of accountability
- Business impact:
  - Financial damage
  - Reputation damage
  - Non-compliance
  - Privacy violation

Each of these factors will get an evaluation on a scale from 0 to 9, as specified in the Table 4.1. After the testing, I had to modify the definition of some of these factors so that they fit better to the needs of this security assessment. These details were described together with the results of the risk assessment in Section 8.4.

Likelihood and Impact Levels	
0 to <3	Low
3 to <6	Medium
6 to 9	High

Table 4.1: The scale for estimating Likelihood and Impact levels.

The overall risk severity will be obtained by combining the likelihood and impact factors as specified in the Table 4.2.

Overall Risk Severity				
Impact	High	MEDIUM	HIGH	CRITICAL
	Medium	LOW	MEDIUM	HIGH
	Low	NOTE	LOW	MEDIUM
		Low	Medium	High
		Likelihood		

Table 4.2: The overall severity is obtained by combining the Likelihood and Impact levels.

4.3 Testing Equipment

In my testing, I will be using my laptop that is running Ubuntu 18.04.4 LTS with the relevant penetration testing tools installed (same tools as found on Kali Linux). I will be using Amazon Echo Dot device – 2nd generation (Software Version number: 645584020), and Amazon Alexa app (version: 2.2.326015.0) running on an Android phone (Android version 9).

Network tests will be conducted on my own Wi-Fi network.

I have an active Amazon account and an active AWS account.

I will be using techniques I learned from the online courses about ethical hacking [46] and penetration testing [47, 48, 49], which explained the terms used in cyber security, how the security policies are created, how to identify the attack vectors and structure a penetration testing plan and how to carry out the testing using the basic penetration testing tools (e.g. Nmap, Netcat, Metasploit).

5 Reconnaissance

I already presented the basic workings of Alexa and its whole ecosystem in Sections 2.1 and 2.2. Here, I provide more technical details that I discovered during the in-depth analysis of the ecosystem. The goal of this analysis was to uncover the potentially vulnerable parts of the ecosystem which I then attempted to exploit during the testing.

5.1 Echo Dot Hardware

In an attempt to uncover any hidden ports (e.g. debug ports) and to see what the Dot is doing under the hood, I decided to open it up.

The only part of the housing that was using adhesive was a rubber pad at the bottom which is probably used to reduce slippage of the Dot and protect it from vibrations. Under the pad the Dot was held together with four torque screws. There was nothing holding other components of the Dot together, which means the Dot can easily be re-assembled, including the rubber pad, without leaving any marks on the housing.

Figure 5.1: All components of the Echo Dot.

I did not notice any other ports but the Micro USB port for power supply and a 3.5 mm AUX connector to connect an external speaker. The Dot is using 7 microphones, 6 placed evenly on the outer edge and 1 placed in the centre. These microphones are used to identify the direction from which the user is speaking and help to reduce background noise in the recordings. The Dot uses 12 RGB LEDs which show various states of the Dot; e.g. when the Dot is listening the LEDs turn blue. I also noticed that after Alexa is triggered, it will light a small area of the LED ring closer to the user with a lighter blue than the rest of the LED ring. This most likely shows which direction Alexa thinks the utterance is coming from. The Dot has one speaker that is about an inch wide and facing downwards towards the openings at the bottom of the housing. The speaker is housed in an extra piece of plastic casing at the bottom of the Dot (it is about 1 cm tall). This casing is glued together so I had to forcefully open it. Inside, there was nothing but the speaker and 2 smaller pieces of foam. I assume this is some kind of an acoustic chamber.

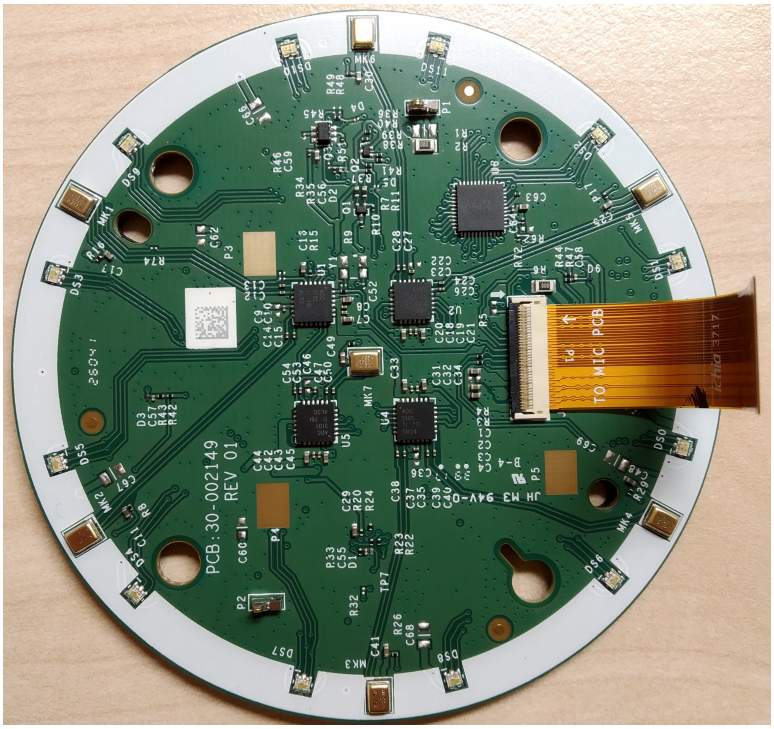

Figure 5.2: Echo Dot has 6 microphones and 12 LEDs placed on the outer ring. In the centre, there is another microphone surrounded by 4 analogue-to-digital converters.

The Dot has 4 analogue-to-digital converters [50] that might indicate that the Dot attempts to process utterances as quickly as possible, while it only has 1 digital-to-analogue converter because it does not need to do any processing on board for that. Thus, it seems that all the Dot is doing is quickly process some raw audio data, perhaps performing noise cancelling, and then playing back the response.

There are some pre-recorded responses saved on the Dot, but they are merely the set-up instructions or they point out that the Dot is not connected to the Internet and direct the user to check the settings in the App.

5.2 Set-up Process

The Dot requires users to use the App to initially set it up and connect it to the Internet. First, the Dot is put into a *set-up mode* during which it creates its own Wi-Fi hotspot. The user then needs to connect their phone to this Wi-Fi. Once the App is connected to the Dot's hotspot, the user is prompted to enter the password of their local Wi-Fi network. The set-up is then completed automatically.

The Dot will stay in set-up mode for about 15 minutes after which it would have to be manually put back in the set-up mode. The Dot's Wi-Fi is not password protected so I was able to connect both my phone and my laptop to the hotspot at the same time. The Dot did not detect that a laptop just connected rather than a phone. This was shown in the responses from the Dot being exactly the same, saying that I was now connected, for both devices.

On my laptop, I used the command `nmap -O [Dot-IP]` to learn that the Dot is running on an Android version 4.1-6.0 (Linux 3.4 or 3.10), which coincides with findings in some other papers [19]. Additionally, with `nmap`, I managed to find three open ports: 443 TCP port running HTTPS service, 8080 TCP port running HTTP-PROXY service, and 53 UDP port running a "domain" service. I was not able to connect to any of these ports. The result I was getting was an OpenSSL error which might have been a problem with the OpenSSL version used with the `curl` command or the lack of appropriate certificates.

5.3 Connecting via USB

The Dot is powered via a Micro USB cable, a common type for many mobile devices, so it can also be powered from the laptop. When I did that, I noticed that the Dot briefly connected to the laptop like a normal phone would. The connection lasted for less than a second so I was unable to explore the possibility of using the Android Debug Bridge (adb) to navigate around the Dot's filesystem.

After an unsuccessful connection, I used the `dmesg` command to find more details about what happened. It seems that the Dot first tried to access an MT65xx Preloader (made by MediaTek), which seems to be a Windows driver [51]. With a quick search online I found some

other suggestions that the Dot might not be Linux compatible [52]. However, I also attempted to connect the Dot to my Windows 10 machine, unfortunately, also without luck.

The Dot can be put in the fastboot mode by holding down the "dot" button on top of the Dot while it is booting. However, unlocking the fastboot fails [53].

5.4 Network Traffic

I connected the Dot and my laptop to the same Wi-Fi network and monitored the traffic coming from the Dot using Wireshark[1]. The Dot was sending multiple TCP packets per minute even in idle state. These are probably keep-alive packets to maintain an active connection with the server, as spotted in some other assistants [20]. These packets were being sent to an IP in the US[2].

When I triggered the Dot, the volume of traffic increased significantly. These larger TLSv1.2 packets were being sent to an IP in Ireland[3]. This was perhaps to reduce the response time and distribute the server load more evenly. All packets were encrypted as expected so I could not inspect the payloads.

Some requests were sent via an unencrypted HTTP. Most of the time it was only a connection test, but as mentioned by [19] the firmware updates are also sent via HTTP. Since such update cannot be triggered manually, I was not able to capture the update binary in an attempt to perform some static analysis.

Such traffic analysis is the basis of en-route profiling.

5.5 Skills Store

I created a polite[4] web crawler to analyse the skills currently available on the Skills Store.

I crawled the Amazon.co.uk store [54] and Amazon.com [55] store separately. I first collected URL for each of skill categories and then navigated through all the skills in that category. Because Amazon limits the number of pages per category to 400, I rerun the crawler a few times, each time with a different skill ordering on Amazon website in an attempt to get details of all the skills. I collected the skills' names and URLs and I used URLs to remove duplicates. The results are shown in Table 5.1.

[1] https://www.wireshark.org/
[2] https://en.asytech.cn/check-ip/143.204.194.136
[3] https://db-ip.com/52.95.121.236
[4] My crawler respected the robots.txt file and I added a 3 second download delay in order not to cause trouble or get banned from the sites.

	Amazon.co.uk	**Amazon.com**
All skills	31122	46791
Unique skills	26099	40613
Non-unique skills	5023	6178
Most common name	*Fun Facts* (46)	*Whose Turn* (74)

Table 5.1: Results from collecting skills from the Skills Stores.

Before a skill is accepted onto the Skills Store it needs to pass Amazon's vetting process which consists of automated and manual tests.

5.6 Skills' Interaction Model

Each skill's interaction model is built in Alexa Developer Console. It is used for defining the *sample utterances* that are mapped to the skill's capabilities. Each capability usually corresponds to its own *intent*. There are some built-in intents that have a set of predefined sample utterances. These intents are: *FallbackIntent*, *CancelIntent*, *HelpIntent*, *StopIntent* and *NavigateHomeIntent*. The *StopIntent* is required to actually stop the skill, otherwise a runtime error is raised.
Below are some other details I learnt from the Amazon Developer Console [56]:

- Skill name needs to be 2-50 characters long.
- Alexa supports 8 different languages, but for some, e.g. English and Spanish, it distinguishes between different regions, e.g. US English and UK English.
- Skill invocation name needs to be at least two words long. It can only contain lower case letters, spaces, apostrophes and periods (in abbreviations). Other characters (including numbers) need to be spelt out. Invocation name cannot contain any of the words which are assigned a special meaning in Alexa, e.g. "launch", "ask", "tell", "open", "Alexa", "Amazon", "app" and similar.
- A sample utterance cannot consist of only slots; it needs a *carrier phrase*.
- Even though the Slot Type Reference [57] claims that a skill should use no more than one *SearchQuery* type slot per intent, I managed to build a skill which used two such slots per intent[5].
- If a phrase slot (such as *SearchQuery*) is used in a sample utterance then no other slot can be combined with it. Doing so results in the build error.

[5]This might be because they were never both used in the same skill session.

- The response cannot exceed 8000 characters or 24 kB. The final recording (combined converted speech and other audio files) cannot exceed 240 seconds. The re-prompt cannot exceed 90 seconds in total.

5.7 Skills' Backend

The easiest way to host the skill's backend is as an AWS Lambda function. Another way is also hosting it on an own HTTPS server with a valid, not self-signed certificate. Regardless of where the main backend, which implements *intent handlers*, is hosted, it can make unlimited requests to external APIs[6]. Because of this, I was able to send all the payloads my skills received back to my server[7] for a closer inspection (in the following chapters I will refer to this server as a "malicious server"). Most of the payload carries information about how to display the response card on devices with a built-in screen (including in the App). More interesting attributes that are sent are [58]:

- *applicationId*: This is a unique identifier of the skill that that request is meant for. Developers can use this to verify that the incoming request was indeed meant for their skill.
- *userId*: A unique identifier for an Amazon account for which the skill is enabled. This ID is re-generated each time a skill is disabled and re-enabled and is only generated for that one skill[8], so cross-skill tracking is limited.
- *deviceId*: A unique identifier for each device. The Skills Kit Reference [56] does not mention that this ID is ever reset.
- *apiAccessToken*: This attribute contains an access token that is used for authorisation with other Alexa APIs, such as Device Location API and List API. This token is different in each request and encapsulates all permissions that users explicitly accepted in the App.
- Slot values: When slots are used, they will contain the value provided by the user. The format of the value given is specified by the slot type. For instance, if a slot expects a number, a numeric representation of the utterance is given, and if a slot expects a phrase then a larger part of the utterance is transcribed and sent in the request.

[6]This will, however, increase the response time and worsen the user experience.

[7]This was a NodeJs server running on my laptop. I exposed it using an Ngrok tunnel (https://ngrok.com/). The server's URL looked like: https://< tunnelId >.ngrok.io, because I was using the free version. This required an update of Lambda function each time the server was restarted, but I believe this was good enough for a proof of concept. This server was used throughout the project with all the skills.

[8]The user ID is not reset if the skill offers *consumable* purchases.

Amazon never sends a full transcription of an utterance to the skill's backend. The only way to obtain parts of the transcription is through slots. Additionally, even though the *userId* changes every time the skill is re-enabled, the process is cumbersome, and a lot of users do not use it [28]. Hence, some basic profiling might happen even across multiple sessions. Such profiling can also happen using the *deviceId* as it is never reset.

5.7.1 Skills' Backend Dependencies

The skills' backend can be implemented in various popular programming languages, such as NodeJs, Python and Java. I used NodeJs as it is the most common choice. I had to install Alexa Skill Development Kit (SDK), available as an open source npm package, which has a lot of other npm dependencies that were also installed.

Some of these dependencies might not be maintained regularly and a vulnerability in one dependency could compromise a skill's backend. Thus, even if a skill does not have malicious intents it could be vulnerable to leaking users' data through an unsafe dependency.

Due to time constraints, I did not go through all of these dependencies because there is more than 50 of them. However, I did look at a few more interesting ones and found, for example, that "oauth-sign" package still offers HMAC-SHA1 and RSA-SHA1 signatures, despite the fact that SHA1 has been proved insecure due to the possibility of a collision attack [59]. If this encryption is still used somewhere in the Alexa Skills Kit then it could potentially be exploited.

Additionally, developers might want to include some other dependencies, needed for their specific skill, which might suffer from similar vulnerabilities.

5.8 Other Notes

Purchasing with Alexa is not enabled unless you have a one-click purchase enabled on the Amazon account.

Voice profile is not enabled by default. When I created my voice profile, I only had to say four sentences, which seems insufficient.

A privacy feature which enables deleting all the recordings of the day by saying: "Alexa, delete everything I said today" is not enabled by default. It needs to be manually enabled in the App first.

Traditional models for wake-up word detection use Gaussian mixture model – hidden Markov model (GMM-HMM) approach, but Alexa supposedly uses a proprietary Deep neural network –

hidden Markov model (DNN-HMM) approach [25].

Alexa understands English that is spoken in a range of accents, as shown in [60], and it can also understand children fairly well, as shown in [61], but the understanding seems to mainly depend on the annunciation. Often Alexa fails to recognise some of the English accents as they tend to annunciate the words less (e.g. some British accents which often soften the pronunciation of "T"-s and other letters). On the other hand, it easily recognises foreign accents and children as long as they annunciate all syllables.

6 Requirements

In this section, I will explain what security requirements Alexa will need to satisfy in order to be considered secure. These requirements were composed before the testing began and partially also served as a testing guidance.

Parties Involved

There are several different types of parties that can interact with the Alexa ecosystem. These parties are defined below:

- **Primary user:** This is the owner of the device whose account is registered as the first adult account on the Dot.
- **Secondary user:** This is the user whose account is registered as the second adult account on the Dot. There can only be one secondary adult user [62, 63].
- **Child users:** Primary user can add up to four children to the Alexa Household [63].
- **Other users:** These are users who interact with the device but have no account registered on it (e.g. guests).
- **Amazon:** Amazon employees and Amazon Affiliates.

Whoever does not fit into any of these groups should not be allowed to interact with the Dot or the App in any way and should not be allowed to obtain any of the users' personal information.

A Primary user, a Secondary user and all Child users together are considered to be users of the same *household*.

6.1 Security Requirements

The three main goals are to protect the users' personal information, prevent unauthorised modification of data and ensure that authorised users can access the Alexa service at *all* times. To satisfy these three goals, Alexa will need to satisfy the following requirements, respectively:

- **Confidentiality:**

  RC1 User's Amazon account credentials cannot be obtained by an unauthorised party.

  RC2 User's voice recordings cannot be obtained by an unauthorised party.

  RC3 User's usage statistics cannot be obtained by an unauthorised party.

  RC4 User's other Amazon account details (e.g. contact details, addresses, credit card details) cannot be obtained by an unauthorised party.

  RC5 Users within the same household cannot access each other's personal information.

  RC6 Any other personally identifiable information cannot be obtained by an unauthorised party.

- **Integrity:**

  RI1 Amazon account details (e.g. shipping addresses, credentials, contact information) can only be modified by the account owner.

  RI2 A Secondary user and a Child user can complete a purchase only if the Primary user allows them to [64].

  RI3 Other users cannot complete a purchase.

  RI4 Network traffic cannot be tampered with (e.g. modification of over-the-air (OTA) updates).

- **Availability:**

  RA1 All users can interact with Alexa (within their rights) at all times.

  RA2 All users can interact with any available skill at all times.

  RA3 All users can access their Amazon accounts at all times.

7 Attacks on Alexa

In this section, I will explain all attacks that have been discovered so far on Alexa. For each attack, I will state a type of an attacker that can execute it, what that attack is aiming at, what architectural point of Alexa ecosystem it is targeting and what equipment is required to successfully execute the attack. I will also provide a description of how to execute each attack and what are the consequences of it. All this information will be considered when evaluating the feasibility and severity of the attacks in the next chapter.

Attacker Models

Different types of attacks require different types of attackers, which largely depends on the means of executing the attacks. For example, some attacks require a physical contact with the Dot, while others can be executed remotely. I separated attackers into three categories:

- **Remote attacker:** This attacker can exploit Alexa from the comfort of their home – they do not need know where target devices are located or who these devices belong to.
- **Close proximity attacker:** This attacker needs to know where the target device is located, they need to at least see the device from the outside of the household or have a mean of controlling the device while still staying outside of the target household (e.g. they are not allowed to break in).
- **Attacker with physical access:** This attacker requires a physical contact with the target device at some point in time.

In addition, all attackers are assumed to have the knowledge to execute *any* type of attack. They are also assumed to have access to a computer to prepare the attacks with (a computer does not count as special attack equipment).

Attack Points

There are several points in Alexa ecosystem that could be susceptible to an attack:

- **Microphones on the Dot**: An attacker could directly exploit vulnerabilities of the Dot's microphones.
- **Other hardware components**: An attacker could perform a hardware hack that would modify the Dot's behaviour.
- **Speech recognition and understanding**: Since machine learning models for speech recognition and speech understanding are not perfect, they can be exploited with carefully crafted inputs.
- **Voice interface**: Alexa might leak some private information to unauthorised parties through its voice interface.
- **Poor lockout mechanism**: Alexa only enables a poor lockout mechanism when using PIN authorisation, which can be exploited.
- **Poor authentication**: Often wake-up word and, in some cases, using a voice profile are the only two means of authentication which can be exploited.
- **Users' misconceptions about Alexa**: An adversary could trick users that are not aware of exact workings of Alexa.
- **Network traffic**: As the Internet is a public place, the network traffic might be an interesting attack vector.

7.1 Questioning Alexa

Attacker model: Attacker with physical access

Target point: Voice interface of an Alexa device

Aim: Obtain user's personal information by conversing with their device

Equipment: No equipment needed as the questioning happens directly on the target device

Description:

Alexa seems to answer any question it is posed with, so I wanted to check if Alexa might also reveal any user's personal or health information. This would require an attacker that has access to an Alexa device so that they can talk with it. Most likely this would be a house

guest or another user in the household, however, even in such cases some users might feel uncomfortable sharing their health information (e.g. what medicine they need to take) or other private information (e.g. bank account balance).

A similar research has been conducted on Alexa which was connected with some other devices, such as a FitBit Surge Wearable Health Tracker [33]. They were able to obtain the following personal information from Alexa:

- Current location
- Work location
- Music preferences
- Current heart rate
- Resting heart rate
- Steps count
- Sleep quality
- Food preferences
- Drink preferences
- Volume of drink
- Home location
- Sleep location
- Contacts
- Schedule

This demonstrates that when Alexa has access to personal information associated with linked accounts it will share this information without a problem when no voice authentication is used [33]. However, when they created a voice profile none of this information was available any more.

Additionally, I wanted to see how Alexa would respond to the following commands:

- ***"What was the last thing I asked you to do?"*** or ***"Alexa, tell me what you heard"***: With these commands I managed to retrieve the last command if it was said in the last minute. For example, I first asked Alexa "Who is Robin Williams" and then after some time I asked her about my last command and it responded with "I heard: who is Robin Williams". However, if I waited too long, Alexa said it had not heard anything in the last minute and that I can check the audio logs in the App. It is worth noting that even if Alexa did not understand what the last command was, it will still say it back.
- ***"What were the last 5 commands I gave you?"***, ***"What email is my Amazon account associated with?"***, ***"When was the last time I used the device?"***: None of these commands were understood by Alexa.

- ***"What is my time zone?"***: Alexa will tell what the current time zone of the device is.
- ***"What is my email address?"***: Alexa responded: "I can't look up your personal contact information yet."
- ***"What is my address?"*** or ***"What is my device location?"***: Alexa responded: "That is not supported yet."
- ***"What is my latitude(longitude)?"*** or ***"What are my latitude and longitude?"*** or ***"What are my coordinates?"***: To the former, Alexa responded with "The latitude(longitude) for London, England is 51.5085(0.12574) degrees north(west)" and to the other two it responded with: "51.5 degrees north and 0.11 degrees west". These coordinates correspond to the location set for the device in the App settings. Even though Alexa does not reveal exact device location, it will provide correct coordinates for the location based on the town or city (not exact street address). This might not be as useful because the attacker already has access to the device and knows its location. However, the user can have a different location specified in the settings (e.g. if they often move the device between their homes).
- ***"Who am I?"***: Alexa sometimes correctly identified me by my voice, but it often did not recognised me. When it did not recognise me, Alexa only said that this is "Luka's account", but when it did recognise my voice it also told me my name.
- ***"Alexa, add gun/cocaine/methamphetamine to my shopping list"***: Alexa added all three items to the shopping list without a problem. Such behaviour might be harmful for Child users as they might also get to hear what is on the lists.
- ***"Alexa, delete everything I said today"***: Alexa responded with: "You can enable deletion by voice on Review Voice History in the Alexa Privacy section of the Alexa app." This means this feature is not available by default. When I enabled this feature, the App told me that anyone with access to my Alexa devices could delete voice recordings from my account. This could be seen as a violation, because anyone could delete my voice recordings even if I do not wish to do so.
- ***"Alexa, why did you do that?"***: I first asked Alexa what the weather was like and then I asked: "Alexa, why did you do that?" to which Alexa replied: "I thought I heard a request for weather." When I asked the same question again, it replied: "I tried to explain

the last thing I did." This is enabled by default and is one of the recently added privacy features [11]

These findings suggest that Alexa will reveal some of private information, but which is still fairly broad (e.g. it will tell the coordinates of the city but not the exact street address). All commands that I tested were relying on native Alexa functionality.

Since I had a voice profile set up, I asked my sister to ask Alexa the same commands. All all responses were exactly the same way. This means that anyone can access the last command, coordinates, time zone, and all the lists on my account without a problem. This was expected as the only user authentication mechanism that can differentiate between multiple users of the household is voice authentication which proved flawed.

Another issue in a multi-user environment is that all commands are logged under one account, so a Primary user can see interactions from Other users. Although, most users are not too concerned about this as they are usually always near each other and will hear the interactions anyway [26].

In conclusion, while native Alexa functionality seems to be protecting fairly well against revealing private information, the research mentioned above [33] found that third-party skills are less secure against such questioning.

7.2 PIN Brute Force Attack

Attacker model: Close proximity attacker

Target point: Poor lockout mechanism

Aim: Complete a purchase, even if the user has set a PIN protection

Equipment: The equipment depends on how the attack is delivered. An attacker could simply utter the commands or they could combine this attack with another attack. In this case the equipment depends on the type of that other attack.

Description:

This attack can be conducted on its own or in connection with other attacks such as the Dolphin attack or Light Commands. The goal is to place an order and then guess the PIN number the user is using to confirm the order.

Alexa allows the users to enable a 4-digit PIN verification for purchases. When a user confirms the product they want to order they have two attempts to utter the correct PIN. After that they must repeat the whole process of ordering. The issue is that this process can be repeated any number of times [15].

Due to the poor lockout mechanism and because Alexa understands replayed speech and even computer-generated speech, as well, it is very easy to write a program to simply try all possible 4-digit PIN combinations. Indeed, researchers showed that this attack is possible [19]. They discovered that the process of ordering and entering two PINs takes about 30 seconds, so exploring the entire PIN space would take well over a day. However, people normally do not choose truly random PIN numbers. Hence, with a frequency analysis the attack time could be reduced significantly [19].

7.3 Skill Squatting

Attacker model: Remote attacker

Target point: Speech recognition component of the Cloud

Aim: Cause users to invoke a malicious skill unknowingly

Equipment:

- Amazon and AWS account to host a skill (base plans are free of charge)

Description:

Skill squatting attack was defined as exploiting predictable errors to surreptitiously route users to a malicious skill [31]. Thus, this attack exploits the way a skill is invoked and it is

possible because of several vulnerabilities in skill names policies. Namely, Amazon allows two or more skills to use exactly the same name (e.g. *Haiku Reader*), to use names that sound very similar (e.g. *Boil an egg* and *Boyle an egg*) and they even allow one skill's name to be included in another skill's name (e.g. skill *Random cat facts* includes the name of another skill called *Cat facts*).

Since this attack has already been disclosed to Amazon [31], I wanted to verify if this attack is still possible or if they have corrected any of these vulnerabilities. I looked at the skills collected with my web crawler (see Section 5.5) and found that there is still plenty of skills that have same invocation names. This coincides with findings of previous studies [31, 65], which indicates that users can still invoke unwanted skills without realising. This does not necessarily mean invoking a malicious skill. Skills with the same invocation names seem to be selected randomly by Alexa [9].

To prove that such skill squatting skills can indeed pass the Amazon's vetting process, researchers managed to successfully publish 5 squatting skills to the Skills Store [32] .

A subset of skill squatting, called spear skill squatting, can also target individuals based on their dialect or their gender [31].

The Skill Squatting attack is not particularly dangerous on its own, but it is a common first step of other attacks, e.g. skill masquerading (see the next section).

7.4 Skill Masquerading

Attacker model: Remote attacker

Target point: Users' misconceptions about Alexa

Aim: Trick users into revealing their private information to what they think is a legitimate skill

Equipment:

- Amazon and AWS account to host a skill (base plans are free of charge)

Description:

Skill masquerading happens when a malicious skill impersonates either a smart assistant or one of the legitimate skills [65]. For example, a malicious skill that wants to steal users' banking information could wait to be triggered, perhaps through means of skill squatting, and then it would start behaving like its legitimate counterpart by offering same functionality. A user will not be able to tell that they invoked a malicious skill rather than the correct banking skill and will thus interact with such skill the same way as with the legitimate one that they trust.

It has been found that many users are not aware of the fact that some skills are developed by third parties and not by Amazon [27], and not all users understand what smart assistants are capable of or not [32]. Therefore, a malicious skill could impersonate Alexa and behave as if Alexa supported advanced features, such as context switching. Context switching occurs naturally in human conversations and happens when one of the speakers involved switches the topic unexpectedly. For instance, during a conversation about cars one person could ask the other one what the time is. The other person would respond by telling them the time and then they would both resume the conversation about cars. This, however, is not supported by Alexa. When a user triggers a skill, they can only use commands that that skill supports. If the command is unknown to the skill it will respond with an error or a re-prompt. However, if people lack this understanding, they could easily assume they are talking with another skill they just supposedly switched to while in fact still talking with the malicious skill. A malicious skill could try to impersonate a number of legitimate skills to increase its chances of success.

Another way to exploit the users lack of knowledge about Alexa's working, is by faking the skill's termination as explained in the next section.

7.5 Eavesdropping Skill

Attacker model: Remote attacker

Target point: Users' misconceptions about how Alexa works

Aim: Listen to private users' conversations in attempt to overhear some valuable information

Equipment:

- Amazon and AWS account to host a skill (base plans are free of charge)

Description:

Eavesdropping skills belong to a subset of skill masquerading attacks. The idea is that a malicious skill would fake its termination by saying "Goodbye", but then it would actually keep recording the conversations.

Faking of the skill's termination relies on the fact that 78% of people expect a skill's response to tell them when it has finished executing [65]. Most commonly they expect the skill to say "Goodbye" or they expect silence. However, when implementing *intent handlers*, the programmer needs to specify whether to actually terminate the skill or not with the `shouldEndSession` argument set to `true` or `false`. Thus, an adversary can send a response saying "Goodbye" but set this argument to `false`. This would mean that even after saying "Goodbye" this skill

would be listening for another response from the user.

The next step is to build an eavesdropping *intent* that would be able to accept anything a user might say. Amazon does not send a transcription of the full utterance to the skills; it only sends back transcriptions that correspond to the *slot* values. Therefore, I created a special intent that understood any utterance that started with one of the more common words "I", "So" or "Well". In addition, I also added sample utterances that started with "Stop", "Cancel" and "Exit" to verify if any of these commands, that were supposed to close the skill, can be hijacked as well. All of these, were followed by a slot of type *SearchQuery* so that it was able to accept any kind of speech. The reason I had to include an explicit word at the start is because Amazon does not allow sample utterances that only consist of one slot.

Now a skill can fake its termination and is able to accept any input that follows one of the common words. The last step is an attempt to keep the skill alive and listening for as long as possible. To achieve this, I modified the custom eavesdropping intent and the built-in *FallbackIntent* to respond with silence whenever Alexa heard any utterance. The *FallbackIntent* had to be modified as well because it will be triggered when Alexa hears an utterance that does not start with any of the common words specified above. What is more, Alexa allows a response from the skill to include an audio file to enrich the user experience. Thus, by including only a silent audio file in both the response (maximum duration 240 seconds) and a re-prompt (maximum duration 90 seconds) I was able to keep the skill alive for more than 5 minutes. This process repeats if Alexa manages to hear any input. I tested what such skill can do and found that:

- Eavesdropping works most of the time but the transcriptions are not always entirely correct (this is because of how Alexa matches the utterance to the *SearchQuery* type slot; I discuss this in Section 7.10, as well).
- A built-in *StopIntent* cannot have attribute `shouldEndSession` set to `false`. This means that the StopIntent cannot be used for prolonging the skill's liveliness because the skill would be closed when the audio response finished playing.
- With the custom eavesdropping intent, I was able to hijack the "Cancel" command, but I was not able to hijack the "Stop" and "Exit" commands. This is good news as more than 90% of users use "Stop" to quit the skill rather than "Exit" [65].
- Alexa cannot simultaneously play a response and record the user. This means that even though the skill can be kept alive for a long time, it can only record during the 6-second

period while it is waiting for a command from the user.

A similar skill was developed by [66], as well, and they produced similar results. They also suggested to include words such as “email”, “password” and “address” as the common utterance starters in the custom eavesdropping intent, because they are likely to be followed by personal information. In addition, they also managed to publish their eavesdropping skill to the Skills Store, by updating an innocent skill after it has passed the vetting process.

Despite being reported that Amazon has started detecting empty recordings [32], I did not have any trouble using them. Even if this detection is done during the vetting process the audio files need to be hosted on an external HTTPS server (not on AWS Lambda), which means they can easily be changed even without modifying the skill’s backend code.

7.6 Missense Attack

Attacker model: Close proximity attacker

Target point: Speech understanding component of the Cloud

Aim: Modify device settings (to conceal attacks), unauthorised purchase, spying (by invoking a malicious skill or by initiating outgoing calls), injecting fake information (email, calendar events, etc.) [4], control smart home devices (including smart locks)

Equipment:

- (Optional) a speaker to play the commands

Description:

A missense attack is defined as an attack with a command that is perceived by humans as speech that is unrelated to the attacker’s intent [2]. Unlike with the indecipherable sound attack (see the next section), a human can understand the sound signal perfectly fine. They just do not realise that this signal could be interpreted as a different command in the Cloud.

There are two types of such attacks. In the first case, the command already gets transcribed differently from what a human would understand. In principle, the reason this happens is the same as in the skill squatting attack (see Section 7.3) – a voice recognition system is misled by unrelated utterances which contain alternative meanings and homophones of the words in a target command [2].

The second type of this attack targets natural language understanding rather than recognition. This means that the command gets correctly transcribed, the same way a human listener would perceive it. However, the command is then mapped to a different intent as the human would

expect. For example, it was shown that Alexa understood utterance "How much ice-cream do I have" as a command "How much money do I have". In a more drastic example, Alexa understood "What is a three phase current balance protection relay" as "What is my current balance". In both cases, Alexa responded as if a valid command was uttered despite the transcription being very different from that command [2]. This shows that Alexa may consider the presence of certain keywords as well as the syntactic structure to determine the most likely intent. However, either of these two aspects could be enough to trigger an action [2].

Such adversarial sound signal may also be embedded in a radio or TV broadcast.

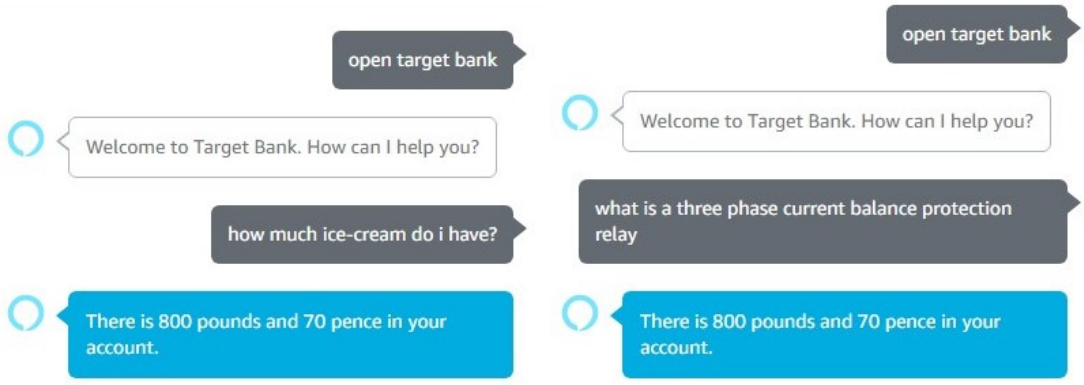

Figure 7.1: An example of a missense attack (taken from [2]).

7.7 Indecipherable Sound

Attacker model: Close proximity attacker

Target point: Speech recognition component of the Cloud

Aim: Same as Missense attack (see Section 7.6)

Equipment:

- A speaker to play the distorted sound signal

Description:

This attack involves generating a sound signal that is understood by Alexa as a valid command but that is perceived as noise by humans. Unlike in the Dolphin attack (see the next section), this sound signal is not inaudible.

Researchers showed that it is possible to generate such sound signal by modifying the so-called Mel-frequency cepstral coefficients (MFCC) which is the most common representation of acoustic features in most speech recognition systems [3]. Then they extracted these features

and regenerated the adversarial sound signal by applying a reverse MFCC to these extracted features [3, 15].

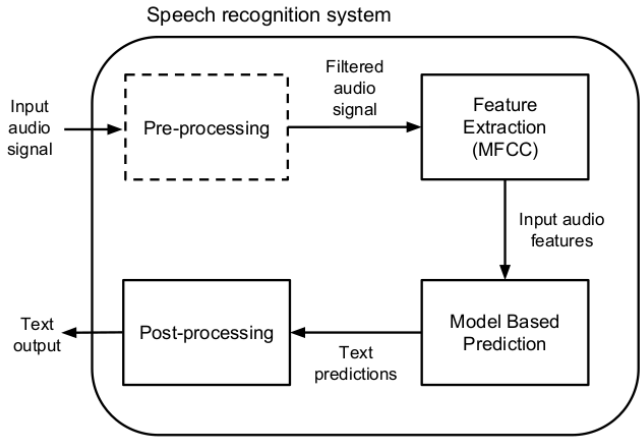

Figure 7.2: A diagram of a typical speech recognition system (taken from [3]).

They demonstrated that this attack is realistic on a Google Assistant. Since they did not make any specific assumptions about how Google's speech recognition system works, they concluded that the attack is most likely possible on any other speech recognition system as well.

Similarly, researchers in [19] managed to hide the word "Alexa" by distorting everything but the distinctive vocal features. They were only partially successful as an attentive user could still make out the word "Alexa".

Regardless, this attack can be prepared in advance through a trial-and-error process and can be tested by an adversary on their own device. Additionally, this attack does not rely on any specific algorithm for generating the adversarial sound signal, and only depends on MFCC transformations which is a standard practice for speech recognition [3]. Because of all this it can be deemed realistic on Alexa, as well.

Such adversarial sound signal may also be embedded in a radio or TV broadcast.

7.8 Dolphin Attack

Attacker model: Close proximity attacker

Target point: The Dot's microphones

Aim: Same as Missense attack (see Section 7.6)

Equipment:

- Ultrasonic speaker Vifa (€450) [4, 67, 68]; alternatively, a tweeter speaker ($52.30) would also work [39, 69, 70]
- Signal source (can be a smartphone)
- Signal generator/modulator ($101.99) [4, 71]
- Audio amplifier ($129.95) [39, 72]
- An alternative portable set-up consists of a low-cost amplifier, an ultrasonic transducer, a battery and a Samsung Galaxy S6 Edge smartphone (first three components cost less than $3) [4]

Description:

This attack involves sending commands to Alexa using inaudible sound frequencies, i.e. frequencies above 20 kHz.

First a wake-up word followed by a voice command need to be generated (either through a TTS system or by recording audio). Then this recording passes a modulation stage which essentially shifts all frequencies from the audible range of 20 Hz to 20 kHz, to above 20 kHz frequencies (a detailed algorithm is explained in [39]). Lastly, this inaudible recording is transmitted with a powerful-enough speaker.

Such audio is inaudible to humans, but a Dot's microphone can pick it up, because these microphones suffer from non-linearity which introduces new frequencies when the microphone processes the sound. This phenomenon occurs before the sound is digitalised, so it is hard to detect with software [39].

The possibility of this attack has been observed in several different devices that support various smart assistants. The maximum distance, from which the attack is possible, also varies depending on the device as well as the power of the ultrasonic speaker. For an Amazon Echo device, which is a larger sibling of the Echo Dot devices, the maximum distance found was 239 cm with the tweeter speaker [39] and 165 cm with a Vifa speaker [4].

I was not able to perform this attack myself due to lack of resources. However, the attack is nicely demonstrated here [40] and here [73]. On Alexa devices, researchers demonstrated

this attack by adding items to the shopping list [39] and opening the back door [4]. These demonstrations prove that various inaudible inputs are consistently being recognised correctly.

The downside of this attack is that it can only be executed from fairly short ranges with very expensive equipment. Using the alternative portable set-up, described above the cost is reduced significantly, but so is the attack distance to as little as 2 cm [4].

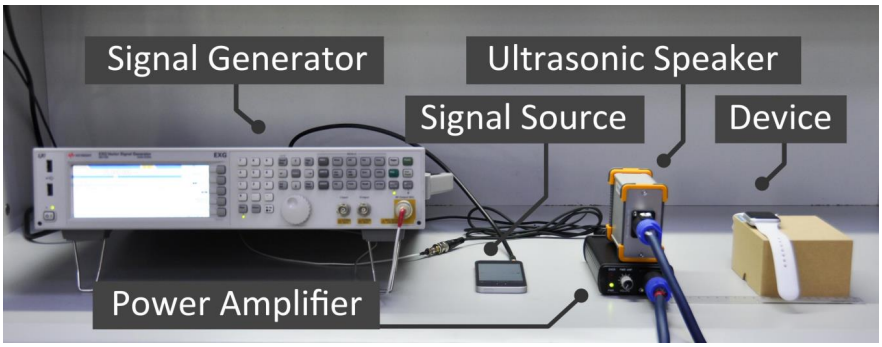

Figure 7.3: An example of a Dolphin attack on a smart watch (taken from [4]).

7.9 Light Commands

Attacker model: Close proximity attacker

Target point: The Dot's microphones

Aim: Same as Missense attack (see Section 7.6)

Equipment (prices taken from [5]):

- Laser pointer ($13.99-$17.99)
- LD5CHA laser diode driver ($339)
- Neoteck NTK059 sound amplifier ($27.99)
- For longer range attacks a telephoto lens (Opteka 650-1300mm, $199.95) can be used to focus the laser and a geared tripod head can be used for greater accuracy. In this case, a telescope or binocular might be needed to see the target device [5].

Description:

This attack exploits a photoacoustic effect that occurs on MEMS (micro-electro-mechanical systems) microphones which can interpret a correctly amplitude-modulated light as a valid voice command [42]. It is still unclear why exactly this occurs.

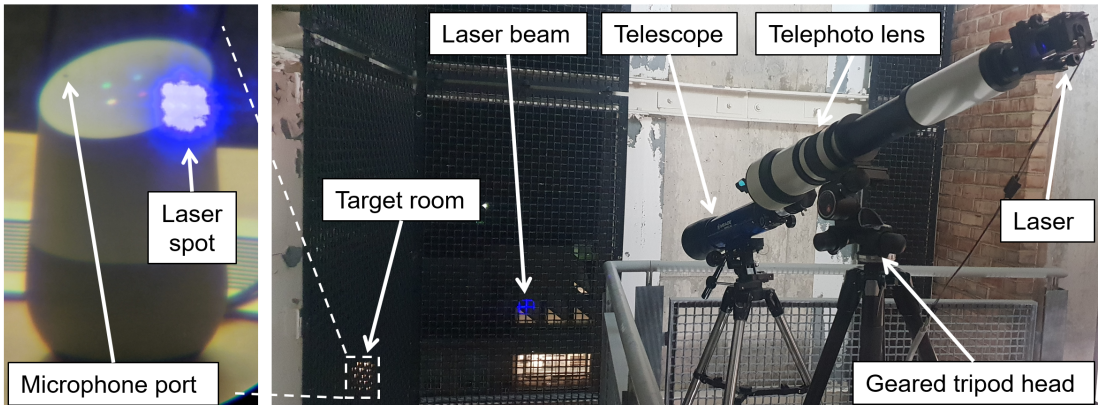

Figure 7.4: An example of a Light Commands attack on Google Home from a building 70 metres away (taken from [5]).

Such an attack works with any type of light not just lasers. This has been demonstrated with the Acebeam W30 laser-excited flashlight [5]. The colour of the laser does not matter and it even works with infrared light and through a window [42].

For this attack to succeed, attacker needs to be able to focus the laser beam precisely on the device's microphone. This becomes significantly harder when the distances increase but it is still possible with some extra equipment as described above. All tested Alexa-enabled devices are susceptible to such an attack from distances of at least 50 m [42]. When using a lower powered

laser, researchers demonstrated that this attack works from 110 m, as well, but they claim the only limitation was the size of their testing environment. They also could not perform longer range experiments with higher powered lasers due to safety concerns as such lasers can be very harmful [42].

I was not able to perform this attack myself due to lack of resources. However, the attack is nicely demonstrated here [5] and here [74]. Researchers showed that they can execute any command using a laser, including brute forcing a PIN number [42].

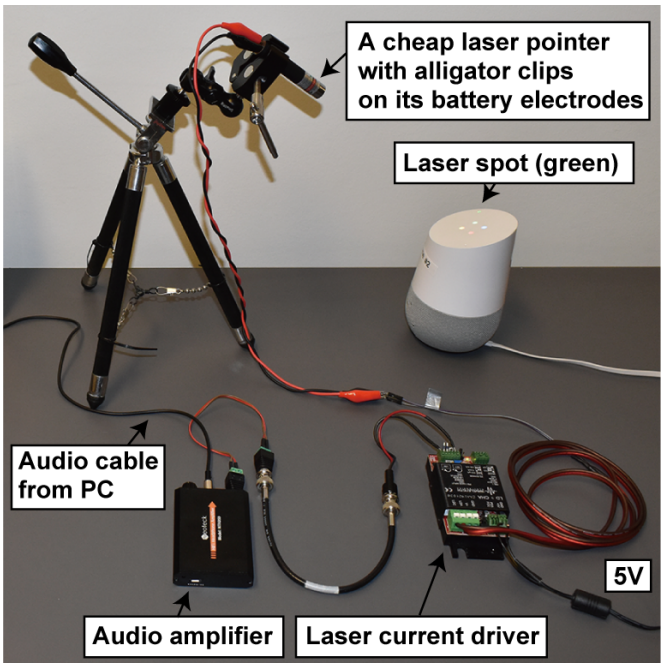

Figure 7.5: A cheaper set-up for the Light Commands attack (taken from [5]).

7.10 Vishing Skill

Attacker model: Remote attacker

Target point: Users' misconceptions about Alexa

Aim: Obtain private user information by employing social engineering through a skill's dialogue

Equipment:

- Amazon and AWS account to host a skill (base plans are free of charge)

Description:

Vishing (also voice phishing) is in essence the same as phishing – an act where an adversary attempts to scam users by making them believe they are using a legitimate service. Vishing

usually involves an adversary calling a person on the telephone and extracting personal information from them through social engineering techniques. I use the term vishing here because a malicious skill can also mimic similar social engineering techniques to scam the user. Arguably, the user might even trust such a skill more than a real person on the other side of the telephone.

Normally, when a skill needs to access users' private information, such as their address (e.g. for shipping purposes or ordering a cab), it should trigger a permissions dialogue in the App. The user would then have to open their phone and manually enable the specific permission [75]. However, since Alexa seems to transcribe any utterance given by the user, I wanted to verify if there are any built-in restrictions as to what type of information a skill can ask the user for and if an adversarial skill could still somehow learn users' private information without utilising the permissions dialogue.

To test this, I developed a simple vishing skill (a Hero Story skill) that asks for the user's name and location and then uses them in a short superhero story[1]. Then it asks the user if they liked the story and offers to send them more similar stories. If the user says they want to be sent more stories, the skill offers to send it by post or by email. Accordingly, it asks for either the user's address or email through the voice interface.

To obtain such information, Alexa offers different slot types. To obtain the name, I used a *Person* slot type and for location I used a *City* type. Obtaining an email or an address was a bit trickier, as Amazon currently does not provide a slot type for an email or an address[2]. For the address they offer some types that focus only on specific parts of an address, such as *StreetAddress*, *StreetName*, *City*, *Country*, but none of these allows to obtain an entire address at once. However, there is one slot type that would listen and transcribe a whole phrase – a *SearchQuery* slot type. This type was designed to capture a more open-ended user input or for search queries a user might enter into a search engine.

Thus, I was able to capture an email and an address successfully with the *SearchQuery* slot. Even if they were not transcribed entirely correctly, the mistakes were easy to understand and correct by a human reader. For example, "@" was usually transcribed as "at" and "." was transcribed as "dot", which was expected. Interestingly, an address "127 Stamford Street" was transcribed as "6:59 Stamford 3" and an address "30 Aldwych, WC2B 4BG" was transcribed as "30 Aldwych wc to be for BG". However, an adversary who knows these were supposed to be valid addresses could still make out the correct addresses, by replacing "to" with "two", "for"

[1]The idea is that the skill personalises the story by naming the superhero with the user's name and placing the scene to the user's town. A real attacker could come up with more convincing and interesting scenarios by adding more specific details about that location and by offering larger variety of stories.

[2]Types *EmailAddress* and *PostalAddress* do exist, but the former is currently only available in a preview release for Amazon Lex [76] and the latter is only available in a beta release [57].

with “four” and “be” with “B”.

Another example of a vishing attack was presented by [66]. Their tactic was to present the user with a message: “This skill is currently not available in your country” when they invoke the skill. Then they inserted a minute-long pause, after which they prompted the user to say their password with: “An important security update is available for your device. Please say start update followed by your password.” A user that is not aware that Amazon would never request a password through the voice interface, could easily get scammed. What is more, the researchers managed to publish this skill to the Skills Store. They first published an innocent skill which easily passed the Amazon’s vetting process and then later they updated the skill’s functionality to perform vishing.

It has been shown that skills can be updated any number of times after they have been accepted once [15, 65, 66]. The vetting process does not repeat after an update. This implies that an attacker could publish also a simple story-telling skill described above and only update it to ask for user information after it has been accepted to the Skills Store.

In summary, ability to update the skill after vetting and the users’ misconceptions about how Alexa works make a vishing skill attack very realistic.

7.11 Phishing Skill

Attacker model: Remote attacker

Target point: Users’ misconceptions about Alexa

Aim: Attempt to steal user’s credentials by redirecting them to a fake login website

Equipment:

- Amazon and AWS account to host a skill (base plans are free of charge)

Description:

With this attack, I wanted to explore a possibility of a classic phishing attack (unlike the voice phishing attack described in the previous section).

For this purpose, I developed a simple Gift Organiser skill. The idea was that this skill would help the user to organise what gifts they have left to buy, for example, during the busy Christmas time. The skill first creates a new Gifts list (in addition to the built-in Shopping and To-Do lists). Then a user can say “Add a football for Michael to my gifts list”. The skill then adds this item to the Gifts list, but it also adds the second item to the list which is “Finalise gift basket at: [URL].”

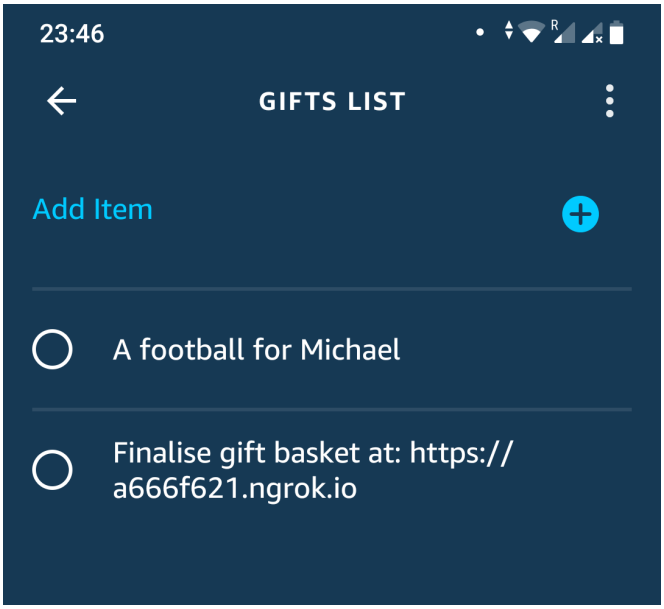

Figure 7.6: An example of an outcome after using a Gift Organiser skill.

The URL in the second item is a malicious URL that is pointing to a fake Amazon login page which is hosted on my malicious server. This URL mostly consists of random numbers and letters as described in 5.7, but an attacker could, for example, use a short URL made with, say, Bitly[3] which would then open a fake Amazon login page hosted on a domain with a similar name to www.amazon.com. For instance, a domain www.amazom.com is still available at the time of writing and is similar enough to the legitimate domain name to not trigger any suspicion to an inattentive user.

The fake Amazon login website is essentially an actual login page that I only slightly modified. To obtain all the HTML, JS and CSS files I simply went to the Amazon's login page and used CTRL+S command to save the website and all the necessary files were downloaded automatically. I then only changed the HTML form action parameter to point back to my server. To make the website even more convincing and "explain" to the user what is happening I included a small warning on top of the page, saying that they have been redirected from Alexa and then I prompted them to sign in for additional security. I did not even have to write the HTML and CSS for the warning box as it was already implemented by Amazon. I only changed its visibility and its text.

I did the same for a desktop version and for a mobile version of a login page, so that the site was convincing regardless of the device used. After the user enters the credentials on this fake login page, their credentials are sent back to my server. I then redirect the user to a "Gift

[3]https://bitly.com/

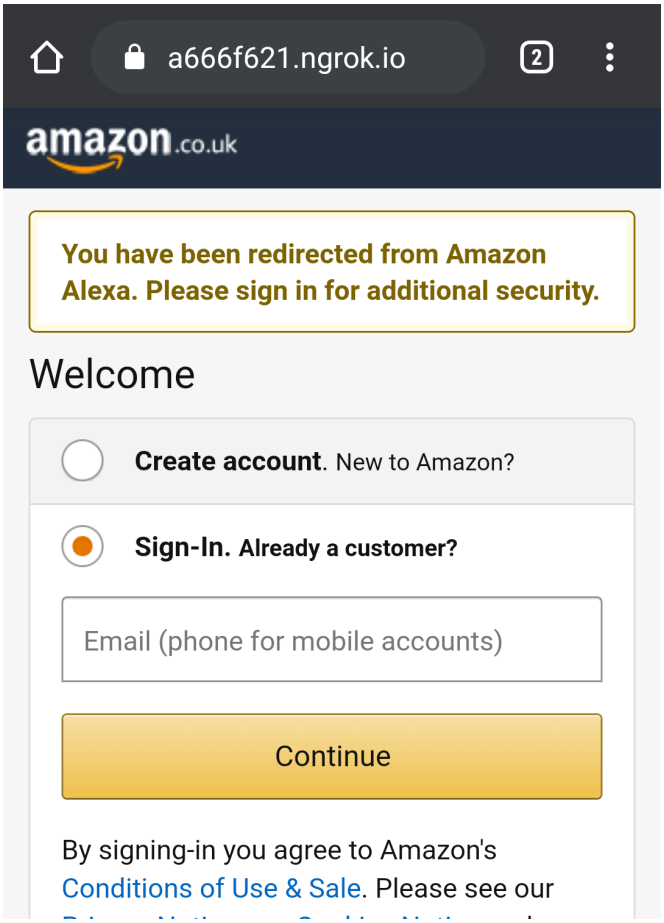

Figure 7.7: A spoofed Amazon login page.

baskets" section on actual Amazon website. This could alert some more users, but at this point I would already be in possession of their credentials. I showed both spoofed pages to 10 other people and they all confirmed that the pages are indeed very believable, which means that a certain percentage of people could indeed get scammed. Due to ethical reasons, I did not deploy this skill to test its full potential.

Regardless, I did manage to learn the following. When the malicious URL is added to the Gifts list it is not made clickable by Alexa. Despite this, mobile OS-s often provide shortcuts for text that looks like a URL, so I did not have to manually copy and paste the URL to my browser. Rather, my phone offered to open the URL in browser as soon as I just clicked on it once. On the contrary, when I opened the Gifts list in my browser in my laptop, I had to manually copy and paste the link to the URL section.

The URL could also be added to the response cards that are displayed in the activity log and on devices with a screen. These URLs are also not made clickable, since they are considered bad practice [77]. Additionally, I could only copy and paste a URL displayed in activity logs

from the browser interface. I could not copy the URL from a card in the App. This is good as users would have type in the URL manually, which is less likely. On the other hand, an adversary could display a fake customer contact number, as suggested by [32], which would be the first step of the more common vishing attack - in person over the phone.

Note that any website could be spoofed the same way as described above to obtain credentials for other platforms. Additionally, an adversary could set that malicious link only appears once in every 100 times, instead after every item added like in my prototype skill, to reduce suspicion.

7.12 Profiling

Attacker model: Remote attacker or Close proximity attacker (depends on type of profiling)
Target point: Network traffic between the Dot and the Cloud
Aim: Obtain private information sent over the Internet or infer users' habits
Equipment: No special equipment needed
Description:

As defined in [15]: "Profiling identifies, infers and derives relevant personal information from data collected from users." This includes their interests, behaviours and preferences.

There are several different types of profiling. The first one is referred to as en-route profiling and requires access to the network traffic of the user. According to [15], the most plausible adversary could, thus, be a dishonest or unethical Internet service provider.

To see how this traffic behaves, I used Wireshark to scan the traffic coming from the Dot (see Section 5.4). Despite all this traffic being encrypted, it is still clear when a user is using their device. Therefore, an adversary could learn their habits and infer when they are not at home. Then they could perform a more elaborate attack on the device without the risk of the user noticing it, or they could perform other crimes, for example break in.

Other types of profiling include processing of the user data that has already been sent to the Cloud. One type of such profiling can happen by third-party skill developers [15]. A user can explicitly say what private information, such as address and email, they want to share with each skill through permissions but they have no control over what a third-party skill actually does with that data and what they manage to infer from it. Users also cannot be sure how that data is stored and shared with other third parties because the privacy policies are often missing, invalid or irrelevant to the skill [12].

Finally, profiling can, and most likely does happen by Amazon. This is mainly done so to keep improving the service and so that Amazon can offer the users more relevant content. However,

due to advances in data analysis it is now possible to infer even some private information that users might not want anyone else to know. For example, it has been shown that smart assistant providers could be able to even infer the intimacy of a couple and how healthy their relationship is [15].

7.13 Denial of Service

Attacker model: Close proximity attacker

Target point: The Dot's network connection

Aim: Prevent Alexa from accessing the Cloud

Equipment: No special equipment needed

Description:

To test the behaviour against a denial of service (DoS) attacks I connected the Dot and my laptop to the same network. I then run the command: `nping --tcp-connect -rate=90000 -c 9000000 -q [Dot-IP]` . While this command was running, I said "Alexa" to trigger the Dot. The first time I said it, the Dot's LEDs turned blue as normal, but they stayed on for some time as if Alexa was still listening for a command (despite my saying the command very clearly and very close to the Dot). The blue light turned off after about 25 seconds. Every subsequent attempt to trigger the Dot resulted in LEDs turning red and Alexa saying: "Sorry, I'm having trouble understanding right now. Please try a little later."

This confirmed that the Dot is indeed vulnerable to such DoS attacks, which prevents the users from using Alexa on the device.

7.14 Cloud Spoofing

Attacker model: Remote attacker

Target point: Poor Cloud authentication mechanism in the skills' backend

Aim: Trick a third-party skill into believing it was sent a request from Alexa and abuse this to send malicious payloads to that skill

Equipment:

- Amazon account (base plans are free of charge)
- A server to host a mock skill (can be a laptop with a valid SSL certificate)

Description:

This attack was discovered recently by [6]. They discovered a vulnerability in third-party skill servers (or endpoints). Each skill hosted on a developer's own HTTPS server, receives encrypted requests from the Cloud. This encryption is based on a public-key method, so the Cloud encrypts the payload with its own private key and then the skill's endpoint can verify the signature using the Cloud's public key. The issue is that the Cloud uses the same private key for all the skills. Additionally, the Cloud also includes an ID of the skill that that payload is meant for and a timestamp. Developers are encouraged to always verify this skill ID and the timestamp. However, not all developers are experienced enough to correctly implement this (especially because almost anyone with only some basic knowledge about programming can create their own skill). As also discovered by [6], the Amazon's vetting process is not thorough enough to detect the lack of skill ID validation, which leads to the possibility of many vulnerable skills already on the market. The attack is conducted in the following manner:

1. An attacker registers a skill with a fake intent that mimics the target skill's intent (attacker may need to guess intent and slot names if the intent expects any). This will ensure that the target skill will understand the spoofed payload.

2. An attacker can trigger their own skill with the malicious payload. The Cloud would then send a request to attacker's endpoint. This request already contains the malicious payload that will be sent to the target skill. For greater accuracy, the attacker can use a text interface to trigger their skill.

3. The request that the attacker's endpoint receives is already correctly signed by the Cloud.

4. Now the attacker only needs to capture and send this same request to the target skill's endpoint.

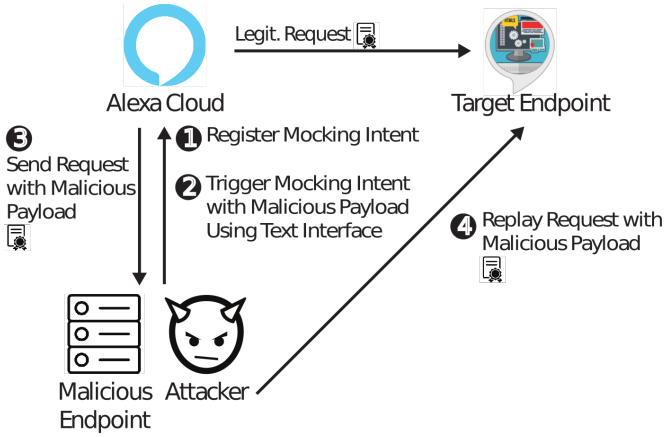

Figure 7.8: A Cloud Spoofing attack (taken from [6]).

This endpoint will believe that the request is coming from the Cloud because it is encrypted with the correct private key (the attacker never learns this key). However, this encrypted request still contains the ID of the attacker's skill, which could, if it was correctly implemented, be detected by the target skill.

Researchers [6] managed to successfully perform this attack (with non-intrusive queries) on 219 real-world skill endpoints. They found that a lot of these vulnerable endpoints are used by skills for controlling electric cars, smart locks, security cameras and similar.

Attackers may also use this approach to perform SQL injections attacks on the third-party servers that do not correctly implement SQL injection defence mechanism (e.g. sanitising inputs) [6]. While this attack worked on their own skill, the real-world attacker would probably need to guess the SQL database structure making such attack less viable.

7.15 Spy Bug

Attacker model: Attacker with physical access

Target point: The Dot's hardware

Aim: Eavesdrop and record user whenever the Dot is plugged in

Equipment (prices taken from Amazon [78]):

- Arduino Pro Mini (£10.25)
- Adafruit Electret Microphone Amplifier – MAX9814 (£7.79)
- SD Card Module (£2)
- Micro SD card (£10.99 for 64GB)
- (Optional) SIM800L GPRS GSM Module (£7.99), SIM card with mobile data plan, Lithium Ion Polymer Battery 3.7V/4.2V battery (£15-£25)

Description:

After I opened up the Dot to check for any additional debug ports (which were not there), I noticed it was very easy to put the Dot back together without leaving any marks on the housing. I realised that should there be a way to modify the Dot and sell it from 2nd hand, the new owner will have no idea that the Dot has been opened.

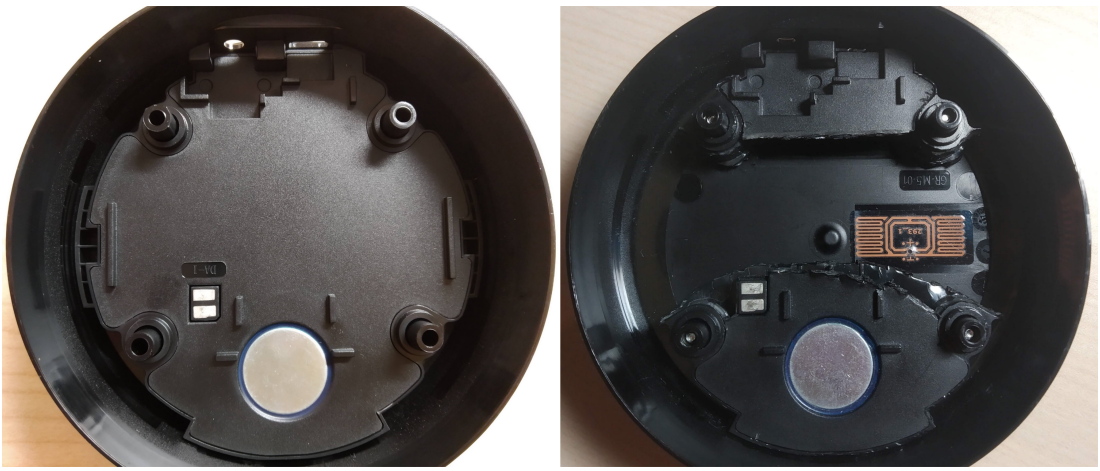

Figure 7.9: Echo Dot's "acoustic chamber" before and after I removed some of the plastic casing. The bits of it that I left in keep the speaker and the ports in place.

Attempting to modify the chipset and other components would require too much time and I would risk destroying my Dot completely (by "bricking" it). However, upon closer inspection, I realised that by removing some of the plastic casing of the supposed acoustic chamber (see Section 5.1) I would produce enough space to fit in a small listening device made with an Arduino Pro Mini, an Electret microphone and an SD card module. This device can be powered

directly from the Dot's main circuit by connecting directly to Micro USB pins at the Dot's power supply port. Since the sound from the Dot is already not of the best quality, removing a larger part of the acoustic chamber did not cause any noticeable difference to the sound quality.

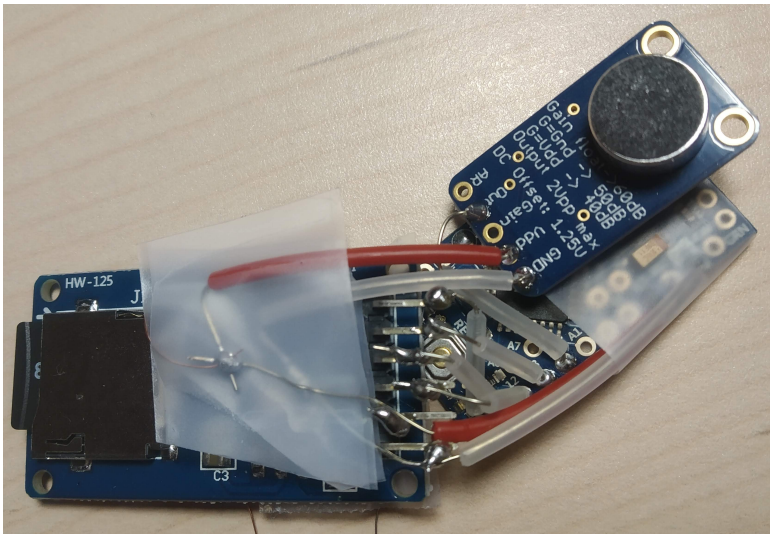

Figure 7.10: A small listening device made with Arduino Pro Mini, a microphone and an SD card module.

An attacker could buy some Echo Dot devices, plant such a device in them and then sell it as a 2nd hand device. Then perhaps after a week or two of recording an attacker could report to the buyer that there was a fault discovered in their device and offer to send them a replacement device. Thus, an attacker would be able to obtain the recordings.

I wanted to upgrade this device to even use a GSM module (which can also fit inside the Dot), which would enable me to record for some time and then spend some time uploading the audio recordings to an FTP server. For example, the device could be set to record at the times that people are most likely to be at home and put into uploading mode during the night. Unfortunately, the SIM800L GSM module can sometimes draw 2 A of current which cannot be provided with a Micro USB connector. A solution would be to use a small rechargeable Lithium Ion Polymer Battery which I did not have access to. Additionally, I would run out of space to include such a battery in the Dot.

However, a larger device, such as an Echo or an Echo Plus, could potentially have enough space even for this upgraded device. Indeed, according to a cross section image of an Amazon Echo (3rd generation) [79] it looks like there is plenty of space around the speaker for such a device (including the battery). Should an attacker use this larger Echo device, they could eavesdrop on the user indefinitely or until some hardware components fail (provided they can keep topping up the SIM card and that the Echo device is plugged in).

Figure 7.11: Spy Bug placed inside the Echo Dot.

With a quick search on Gumtree and eBay, I found around 100 Echo Dot 2nd generation devices being sold and even more Echo Dot 3rd generation, Amazon Echo and Echo plus devices. All these devices could possibly be modified without leaving a mark[4].

The components I used to create this listening device are not the smallest in the market but they are a good trade-off between size and price and accessible to anyone. It is worth stressing again that, since this device has its own microphone, muting the Dot would not prevent it from recording.

Another hardware modification of the Dot by directly connecting wires to the main circuit board was shown in [80]. This further shows that the Dot cannot detect any major changes in its hardware.

[4]To confirm this I would need access to other devices to inspect them in case Amazon introduced some protection against opening these devices up in the newer generations.

7.16 Miscellaneous Voice Attacks

Attacker model: Close proximity attacker

Target point: Poor authentication/Users' misconceptions about Alexa

Aim: Same as Missense attack (see Section 7.6)

Equipment: No special equipment needed

Description:

In this section, I wanted to briefly describe some of the attacks that are in nature very similar to the previously mentioned ones since they just provide different ways of exploiting the same vulnerabilities, but I thought it was worth mentioning them for completeness.

Alexa has little limit as to who or what provides the voice commands. Therefore, it is vulnerable to sound signals coming from radios, TVs, websites or essentially any other home appliance with a speaker. This has occurred before, when Burger King prompted Google Home devices to read about their burgers from Wikipedia in 2017 [24].

An attacker might also exploit vulnerabilities in smart TV-s and then stream a video with voice commands for the target smart assistant via this compromised TV [34]. Same research suggests that this way an adversary could order something online and pick it up in front of the victim's house before they even notice.

What is more, Alexa can even understand voice commands that have travelled through physical barriers, which means an attacker could yell commands from outside the room and possibly make Alexa unlock the door [19].

Poor authentication in Alexa's multi-user environment in a combination with insufficiently short PIN has been yet once again confirmed very vulnerable when a 6-year-old successfully ordered an expensive dollhouse online [24]. This is sometimes referred to as *a curious child attack* [10].

7.17 Unsuccessful Attacks

This section briefly outlines past attacks and vulnerabilities that have either already been fixed or have been proven unsuccessful on Alexa. I decided to include this section for completeness and as a reference for future research who would wish to conduct a similar security assessment on other smart assistants.

It was shown that a simple *voice SQL injection* attack does not yield a success. Researchers told Alexa "Alexa, drop table orders", but Alexa did not understand the command [19].

A replay attack involves capturing packets sent from the Dot to the Cloud and then sending them to the Cloud again. The expected effect is a repeated command to which Alexa should respond normally. However, researchers [19] showed that Alexa did not react which means there are measures in place to protect against such replay attacks.

On older Amazon Echo devices, it used to be possible to boot into the Linux environment of the Dot from an external SD card attached to exposed UART debug ports [20, 38, 53]. Thus, they were able to gain remote root access to the device. This vulnerability has been fixed in devices sold after 2017 [81].

Until it was fixed in 2017, a remote access could have been obtained also by exploiting the Bluetooth BlueBorne vulnerability [20]. This has been quickly patched in many smart speakers.

8 Evaluation

In this section, I will first summarise the findings and then conduct an evaluation of all attacks by following the OWASP Risk Rating Methodology as described in Section 4.2.

8.1 Major Vulnerabilities

Most of vulnerabilities currently present in Alexa can be split into two major groups.

The first larger group of vulnerabilities results from the faults of the voice recognition and understanding component of the Cloud and enables attacks, such as skill squatting, indecipherable sound and missense attacks. Voice recognition is a difficult computational task and even humans often struggle with understanding other humans when posed with new accents. However, humans understand the concept of context much better so they can more accurately predict the words that sound similar. For instance, Alexa could transcribe a word "tale" as "tail" and since it supposedly uses a one-to-one mapping it will not conduct any post processing [16]. Thus, Alexa could understand a "bedtime tale" as a "bedtime tail" which does not mean much. Should Alexa use similar post processing techniques as Google Assistant [2] it could fix the transcription[1] since words "bedtime" and "tale" are more often used in the same context.

A big issue is also the understanding of the users' intentions and idioms. Human language is very rich and every region contains their own idioms. Humans also use different intonation to convey a part of the message. All these are difficult tasks which can result in misinterpretations of various commands by Alexa.

The second group of vulnerabilities results from the weakest point in many systems – the users themselves. Most users are not aware of how Alexa works and do not even know that some skills can be developed by third parties [27]. Arguably, they would trust such skills more, because they believe they were developed by Amazon. Users can also overestimate the abilities of smart assistants and that can be exploited. Furthermore, many users believe that all tasks, including handling permissions and account linking, can be completed through the voice interface [27]. Additionally, users often fail to properly configure the security features available, which might happen because these features are not available by default. Thus, as stated by [33]: "The real vulnerability lies in the user's own portal to conveniently access their data."

[1] However, this introduces new attack possibilities, such as the *nonsense attack* [2].

Such lack of awareness makes attacks, such as voice masquerading, eavesdropping skills, vishing and phishing attacks very viable.

8.2 Evaluation of Previously Discovered Attacks

Since Alexa is developing rapidly, I wanted to explore which of the previously discovered attacks are still possible and which vulnerabilities might have already been fixed by Amazon.

Skill squatting is still possible as there are still many skills with the same or similar names in the Skills Store. Since this attack is usually done in combination with skill masquerading they are both still viable attacks. Even though both vulnerabilities have been around since the beginning, Amazon has not done much about it.

Similarly, vishing and eavesdropping skills can still exist on the Skills Store despite having been reported to Amazon [66]. Amazon also replied that they have put additional measures in place to prevent and detect this type of skill behaviour. They said it is no longer possible for skills that eavesdrop to be submitted for certification and that they have put mitigations in place to detect skills that ask users for their personal information [82]. However, when I was testing my skill I did not have any problems. The reason might have been that the skills were only tested in development mode. On the other hand, the execution of skills in development mode still follows the same process as the execution of published skills, so perhaps the detection of such skills by Amazon mentioned above only happens during the vetting process. Indeed, due to their flawed vetting process, researchers showed that it is still possible to publish vishing and eavesdropping skill to the Skills Store as of 17 December 2019 [66]. For these reasons, I also believe that a social engineering type of a vishing skill, such as my Hero Story skill (see Section 7.10), could also be published.

En-route profiling is still possible but can convey very little information. A more worrying type of profiling is by third-party skills and by Amazon themselves. Even though there is very little information about how exactly Amazon processes the users' information, the fact that Alexa explicitly told me that Amazon Music will contain interest-based ads indicates that Amazon indeed processes user information also for reasons other than to improve voice recognition and understanding models. Moreover, it has been reported that Amazon employees routinely listen to the recordings to improve the voice recognition software [26, 83]. This poses a big privacy concern, especially because privacy policies do not clearly state that other humans may be reviewing the recordings [26]. Additionally, this indicates that user data is not encrypted in the Amazon databases and that any employee with access to it could intentionally or accidentally

leak private user conversations and other data. Another piece of evidence indicating that the voice recordings are not stored encrypted can be understood from Amazon's replies to the Arkansas police in 2017. The police wanted to access the suspect's voice recordings, but Amazon refused to hand them in as this would violate the customer's privacy [38]. If the recordings were indeed encrypted, they would not be available neither to the police nor Amazon.

As users mostly place their devices in central locations or hallways [28], the Dolphin and Light Commands attacks might be less feasible because the devices are unlikely to be close to windows or doors. Also, Light Commands are much more likely in larger urban areas and big cities because an attacker could more easily gain access to an elevated attacking position due to many skyscrapers and other tall buildings.

8.3 Requirements Violation Summary

In Section 6, I presented the security requirements that need to be met for Alexa to be deemed a secure system. Unfortunately, the attacks I listed in the previous chapter show that a lot of the requirements are being violated.

In Table 8.1, I summarised which requirements are broken by which of the attacks. Most of these violations are clear, but I justified others below.

	Confidentiality						Integrity				Availability		
Attacks \ Requirements	RC1	RC2	RC3	RC4	RC5	RC6	RI1	RI2	RI3	RI4	RA1	RA2	RA3
Questioning Alexa					✗		✗						
PIN Brute Force Attack				✗				✗	✗				
Skill Squatting												✗	
Skill Masquerading	✗			✗		✗							
Eavesdropping Skill						✗							
Missense Attack									✗				
Indecipherable Sound									✗				
Dolphin Attack									✗				
Light Commands									✗				
Vishing Skill	✗			✗		✗							
Phishing Skill	✗			✗		✗							
En-route Profiling			✗										
Other Profiling			✗			✗							
Denial of Service											✗	✗	
Cloud Spoofing										✗		✗	
Spy Bug			✗			✗							
Miscellaneous Voice Attacks								✗					

Table 8.1: Summary of which security requirements are violated in each attack.

Vishing skills and skill masquerading violate RC1 because they can both impersonate a native Alexa behaviour and trick the user into giving them the credentials. Skill squatting

partially violates RA2 because for skill names that Alexa consistently transcribes incorrectly, they can be squatted and, thus, a user might never have access to that skill. Miscellaneous voice attacks violate RI2 because of the curious child attack.

8.4 Risk Assessment

In order to gain a more structured security assessment of Alexa, I evaluate each of the attacks (and, thus, each of the vulnerabilities they exploit) according to the OWASP Risk Rating Methodology [45]. Even though the names of the Likelihood and Impact factors are the same as in the methodology description, I adjusted the meaning of some of them to correspond more closely to the type of testing I was doing. Namely, I considered the following changes:

- *Ease of Exploit* also considers the equipment needed for the attack. If an attack requires more expensive equipment then it is "harder" to execute.
- *Intrusion Detection* takes into account of whether a user or Amazon can detect that an attack is taking place.
- *Financial Damage* considers the damage towards the user rather than Amazon, because Amazon does not tend to have any direct financial loss due to any of these attacks unless a user stops using Alexa altogether as a result.
- *Reputation Damage* considers an impact on the use of Alexa should a greater number of these attacks occur in real world.
- *Non-compliance* mainly depends on the security requirements summarised in the previous section.
- *Privacy Violation* relates to the privacy of the user.

In Tables 8.2 and 8.3, I collected the ratings of each factor for estimating attack likelihoods and attack impacts, respectively. Since commenting on each rating would quickly become overwhelming, I will only justify the ratings for which I believe might not be immediately clear.

En-route profiling has a relatively high Motive because usage statistics can be very valuable. On the other hand, its Opportunity is quite low because it requires access to the user's network. Light Commands has a higher Opportunity than the Dolphin attack because it can be conducted from much greater distances. Despite the need for physical access, the Spy Bug has a higher Opportunity because it can be prepared in advance before the target user actually owns the

device. However, its Size factor is lower as it might not be feasible for an attacker from the USA to attack someone in Europe through this approach; it is more feasible for attackers that can send and receive packages for a lower price, i.e. attackers that live closer to the target users.

All attacks that involve implementing various skills can be completed from an attacker's home. Hence, they have very large Opportunity and Size factors. Most of these attacks can also be detected by checking the user's activity history for anomalies and unrecognised activity, but since a lot of users do not check their history very often [26] the attacks may remain unnoticed. Same holds for the Dolphin attack and Light Commands, but these two attacks can send a command "Alexa, delete everything I said today", a recent new feature [84], after having caused something malicious in an attempt to delete the traces.

Intrusion detection is also fairly low for the skill masquerading attack, because as soon as an impersonating skill fails to copy the exact behaviour, a regular user might notice that something is wrong. On the contrary, Intrusion detection is very high for vishing and phishing skills, because most users are not aware about how Alexa works and what kind of information the skills can request from them, so even if the activity is logged it will not raise any suspicion.

Intrusion detection is low for indecipherable sound, cloud spoofing and PIN brute force attacks, because indecipherable sound can still be recognised as a command by an attentive listener [19], cloud spoofing can easily be detected by checking the *applicationId* parameter in the incoming request, and a successful purchase by brute forcing the PIN would still result in multiple order update and notification emails from Amazon to the user's email address.

Dolphin and indecipherable sound attacks could cause a shorter period of availability loss due to the possibility of jamming the device. Eavesdropping skill can also temporarily prevent normal access to Alexa (the eavesdropping skill has to be stopped first) while it is playing the silent audio files as a response. The cloud spoofing attack could crash a skill's server with a carefully crafted malicious payload and, thus, cause loss of availability for that particular skill.

Profiling by third parties and by Amazon is not completely unknown of, but its Accountability factor is still very high because users do not exactly know what is happening with their data and if any of the skills are also doing any profiling. This type of profiling also has a relatively high Non-compliance score because privacy policies are not sufficiently verified by Amazon and are often also missing [12].

Cloud spoofing has a very low Accountability factor because the target skill server can easily trace the *applicationId* parameter back to the attacker's skill.

Dolphin and Light Commands attacks do not directly disclose any personal information

themselves, but would need to trigger an eavesdropping skill and disclose some information this way. Thus, they have a low Privacy Violation factor. Skill masquerading has a higher Privacy Violation factor because it obtains information that was meant to be (more or less) privately conversed between a user and a legitimate skill.

	Threat Agent Factors				Vulnerability Factors				Overall	Severity
Attacks \ Factors	Skill Level	Motive	Opportunity	Size	Ease of Discovery	Ease of Exploit	Awareness	Intrusion Detection		
Questioning Alexa	9	5	1	2	8	7	5	5	5.25	Medium
PIN Brute Force Attack	8	8	1	3	8	7	6	4	5.63	Medium
Skill Squatting	8	5	9	9	6	8	7	7	7.38	High
Skill Masquerading	7	6	9	9	7	6	7	5	7.00	High
Eavesdropping Skill	7	6	9	9	5	6	6	4	6.50	High
Missense Attack	6	2	7	7	5	3	4	4	4.75	Medium
Indecipherable Sound	6	4	7	7	7	6	6	4	5.88	Medium
Dolphin Attack	6	6	4	7	5	4	5	8	5.63	Medium
Light Commands	5	6	7	8	3	4	4	8	5.63	Medium
Vishing Skill	7	7	9	9	8	8	5	7	7.50	High
Phishing Skill	6	9	9	9	6	8	4	7	7.25	High
En-route Profiling	3	4	3	4	7	6	6	9	5.25	Medium
Other Profiling	4	5	9	7	5	6	6	9	6.38	High
Denial of Service	3	1	3	4	6	6	7	3	4.13	Medium
Cloud Spoofing	3	4	9	9	4	4	3	5	5.13	Medium
Spy Bug	5	4	5	6	4	6	3	9	5.25	Medium
Miscellaneous Voice Attacks	8	5	1	2	6	6	4	5	4.63	Medium

Table 8.2: A likelihood estimate for each attack.

Attacks \ Factors	Technical Impact				Business Impact				Overall	Severity
	Loss of Confidentiality	Loss of Integrity	Loss of Availability	Loss of Accountability	Financial Damage	Reputation Damage	Non-compliance	Privacy Violation		
Questioning Alexa	4	3	0	6	1	4	2	3	2.88	Low
PIN Brute Force Attack	5	7	0	5	6	4	7	4	4.75	Medium
Skill Squatting	2	1	0	1	0	4	2	0	1.25	Low
Skill Masquerading	6	0	0	1	2	5	5	4	2.88	Low
Eavesdropping Skill	6	0	2	1	4	5	4	5	3.38	Medium
Missense Attack	2	2	0	5	1	2	2	0	1.75	Low
Indecipherable Sound	2	4	1	5	2	2	2	0	2.25	Low
Dolphin Attack	2	4	1	7	3	3	2	1	2.88	Low
Light Commands	2	4	0	8	3	4	2	1	3.00	Medium
Vishing Skill	8	0	0	5	5	4	7	7	4.50	Medium
Phishing Skill	9	0	0	5	7	5	7	7	5.00	Medium
En-route Profiling	4	0	0	8	0	1	2	4	2.38	Low
Other Profiling	6	0	0	9	0	5	5	7	4.00	Medium
Denial of Service	0	0	9	7	1	1	2	2	2.75	Low
Cloud Spoofing	2	5	5	1	1	2	5	3	3.00	Medium
Spy Bug	6	0	0	5	4	4	5	5	3.63	Medium
Miscellaneous Voice Attacks	3	2	0	5	2	4	2	2	2.50	Low

Table 8.3: An impact estimate for each attack.

8.5 Evaluation Summary

I obtained the overall severity of each attack by combining data from Tables 8.2 and 8.3 as specified by the methodology (see Section 4.2). The final results are collected in Table 8.4.

Attack	Likelihood	Impact	Overall Attack Severity
Questioning Alexa	Medium	Low	LOW
PIN Brute Force Attack	Medium	Medium	MEDIUM
Skill Squatting	High	Low	MEDIUM
Skill Masquerading	High	Low	MEDIUM
Eavesdropping Skill	High	Medium	**HIGH**
Missense Attack	Medium	Low	LOW
Indecipherable Sound	Medium	Low	LOW
Dolphin Attack	Medium	Low	LOW
Light Commands	Medium	Medium	MEDIUM
Vishing Skill	High	Medium	**HIGH**
Phishing Skill	High	Medium	**HIGH**
En-route Profiling	Medium	Low	LOW
Other Profiling	High	Medium	**HIGH**
Denial of Service	Medium	Low	LOW
Cloud Spoofing	Medium	Medium	MEDIUM
Spy Bug	Medium	Medium	MEDIUM
Miscellaneous Voice Attacks	Medium	Low	LOW

Table 8.4: Overall severity of each attack.

The results show that three out of four high-severity attacks, namely vishing, phishing and eavesdropping attacks, are executed through skills that can be published on the Skills Store by anyone. This further indicates the importance of implementing a thorough and complete vetting process by Amazon. The fourth high-severity attack is probably harder to avoid as profiling by Amazon is one of their main goals for increasing revenues through targeted advertising. But perhaps Amazon could put more measures in place to reduce profiling by third parties which is also a big privacy issue since not all developers acquire the necessary experience to correctly handle and store personally identifiable information and can, thus, leak users' private information (intentionally or unintentionally).

All medium-severity attacks require a target to be set in advance. For example, when skill squatting, an attacker needs to know which skill they are trying to squat; with skill masquerading, an attacker needs to know which skills it is trying to imitate. This is probably the only reason why they are placed in a lower severity rank, but they are nonetheless a viable attack opportunity.

Low-severity attacks cannot be executed on a large scale.

9 Mitigation

Here, I will present what mitigation steps could be undertaken by Amazon to ensure a safer and more trustworthy Alexa. Indeed, a lot of attacks could be prevented by users paying more attention to what they are doing and educating themselves about Alexa, but with millions and millions of users it seems unrealistic to expect this from every single one of them.

Below, I will summarise some mitigation steps suggested by other studies as well as propose my own mitigation methods and security improvements.

9.1 Authentication

With only four available wake-up words, Alexa only offers very poor authentication mechanisms, which makes it hard to properly manage which resources can be accessed by which user in the household. Even if it offers biometric voice recognition this mechanism is not sufficient because users' voices tend to change with age, illness or tiredness [34].

Thus, the first reasonable suggestion would be to allow the users to set any word they want as a wake-up word. This would make it much harder for anyone else to guess it which will also prevent them from executing any other commands.

To prevent accidental recording, Alexa could play a short sound signal whenever it is triggered, regardless of the volume set. This could alert the user when Alexa would, for instance, mishear its name. Another option would be that if the wake-up word match is only, say, 80-90% accurate, then Alexa could ask the user "Did you call me?", which would present another layer of authentication.

Alternatively, to increase the accuracy of recognising only the correct wake-up words, I propose the following. Whenever Alexa is triggered with a wake-up word, it will initially only send a recording of this wake-up word to the Cloud. Since the Cloud has access to more resources it could more accurately transcribe it and verify if it is indeed a valid wake-up word. In the meantime, the Dot would keep recording the rest of the supposed command, but will not send the recording of it to the Cloud until it receives a confirmation from the Cloud that a correct wake-up word has been uttered. If the Cloud rejects the wake-up word, then the Dot would stop recording and discard the command.

I also found other interesting suggestions in the literature. For example, paper [34] mentions a wearable device that would collect skin vibration signal, which would give another confirmation signal to Alexa that the user is indeed the one talking. However, users might be reluctant to wear such a device all the time.

Researchers in [25] suggested implementing a tagging device (perhaps a phone with a special app) that would emit some kind of an acoustic signal which could be audible, inaudible and even hidden (like a watermark). This signal would be embedded in the utterance recording and would notify the Cloud whether the user has given their consent to be recorded or not. Whenever the Cloud detects this tag it would immediately discard such recording before processing it any further [25]. A user could activate such a device whenever they do not wish to be recorded.

A Wi-Fi technology based authentication system has been proposed by [34]. This means that a user could wave their hand to un-mute the speaker and then use the device as normal. They used Wi-Fi rather than infrared signal because Wi-Fi is already installed in people's homes, it is cheaper and it is not restricted by the light-of-sight propagation. They show that this system reliably only recognises indoor movements.

9.2 User Awareness

As mentioned, it is unlikely that all users will learn about all privacy controls prior to using Alexa. Therefore, a possible solution would be to remind them about these controls while they are using it. For example, Amazon could highlight the main security points when the user first registers a device with their account. They should be explicitly informed that skills can be developed by third parties, that they should never give their emails, passwords, addresses, card details and other personally identifiable information through the voice interface even if asked to do so by the skill.

Visual cues with LEDs are often insufficient because Alexa can also be used while the user is not looking at the Dot. Therefore, [38] suggests playing a notification sound at the start and the end of each request. Similarly, [27] proposes playing a different sound for native and third-party skills. They further suggest that a skill could announce the developer's name and similar information to the user when the skill is first enabled. However, they also argue that both suggestions might interfere with the idea of a seamless voice experience that Amazon is trying to build.

9.3 Vetting Process

Amazon's vetting process should be one of the main defences against many vulnerable and malicious skills, but in reality, this process is flawed and misses various important checks [6]. Amazon performs both automated tests and manual tests, which is a better scheme compared to Google Assistant which only conducts manual tests [6]. Despite this, Amazon's review process might still be less thorough due to the sheer volume of skills.

Many attacks could be easily prevented just by making the vetting process a little bit more thorough. Firstly, Amazon could adopt a similar policy as Google Assistant and disallow skills with the same names or skill names that contain other skills' names. They should also extend the list of disallowed words in the skill's name with words such as "I", "please", "thanks" and similar. Secondly, for their skill to pass the vetting process, developers will need to prove that external APIs that their skill is using are legitimate and secure. And thirdly, they should add more thorough automated test. For example, a skill that is vulnerable to the Cloud Spoofing attack could very easily be detected by simply sending a request with an invalid *applicationId* to it and see if it would result in a successful response.

Another issue is that once a skill passes the vetting process, it can be updated any number of times without being re-verified. Amazon could at least run the automated tests on each skill whenever it gets updated. I suggest that skills hosted as an AWS lambda function get automatically tested whenever a new version is uploaded. For skills that are hosted on private servers this might not be possible as Amazon has no way of telling when that skill's backend is updated. Thus, Amazon could randomly trigger automated tests to verify that such endpoints are still secure. Developers would not be able to tell when the tests will be taking place, so the chances of changing the skills to perform something malicious after the vetting process would decrease. Such tests should also include verifying the audio and other resources sent as part of the response to avoid the skill's attaching, for example, silent audio files to their responses for the purposes of eavesdropping.

Amazon should reduce the number of skills and check them more thoroughly like Google does it. But perhaps a really good and robust system is not their priority. Maybe they want everyone to be able to create almost any skill they want so that then Alexa can spread even more as some people will just write skills for fun and then share them with their friends who could, in turn, also learn about other Alexa features and decide to buy one for themselves, increasing the Amazon's revenue.

9.4 Hardware Hacks Mitigation

Since hardware hacks, like the Spy Bug described in Section 7.15 can only be carried out on devices sold from 2nd hand, the best way to prevent such an attack is for each user to avoid buying used devices and always buy Echo Dot from the trusted sellers. If it is in Amazon's interest to prevent re-selling of these devices because they want more people to buy the device from them directly, then they only need to let the users know that this attack can occur on used devices, but not new ones, and, thus, deter the users from ordering elsewhere.

However, if it is in Amazon's interest that as many devices are being used as possible, then they might be happy with people re-selling their used devices. The rationale behind this is that they would rather see the re-selling of their device to another person who might actually start using it, than see the first person keep the device in a box, completely unused. Perhaps the other person would use the device more often and become used to it. In this case, Amazon should ensure that the devices cannot be opened up without leaving obvious marks that would alert the new user that the devices might have been tampered with.

9.5 Profiling Mitigation

A possible solution for en-route profiling might be mimicking the traffic that the Dot sends to the Cloud at random time intervals. This would make inferring actual usage statistics much harder. Simply telling the users to unplug the Dot when they are not at home might not be the best idea because if someone is watching the traffic frequency they will notice that the devices are not being used even more easily.

9.6 Skill Squatting Mitigation

A lot of skill squatting attacks could be prevented if Amazon added a word-based and a phoneme-based analysis of the new skill's invocation name to the automated vetting process and, thus, detect the potential skill squatting attacks [31].

Alternatively, Alexa could implement manual enabling of skills that have same or similar names to any other skill. This way a user could explicitly select with which skill they want to interact with in the future requests.

9.7 Skill Masquerading Mitigation

Researchers in [32] implemented a context-sensitive detector which consist of the Skill Response Checker and the User Intention Classifier. This detector is used to detect anomalies in the user's conversation with the running skill and alert them if a potential risk is detected. Since this approach turned out to be very successful it could prevent a lot of skill masquerading attacks where a malicious skill tries to impersonate Alexa.

9.8 Dolphin Attack Mitigation

The issue with Dolphin attack is that the conversion from ultrasound to audible frequency range occurs on the hardware before it is even converted into a digital signal. This makes Dolphin attack detection with software very difficult, but not impossible. Researchers in [4] showed that they can use a classifier to successfully differentiate between the features of modulated voice commands and genuine ones.

A good solution against Dolphin attack would also be microphone enhancement. As explained in [4] current MEMS microphones allow for frequencies above 20 kHz. Thus, having a microphone designed to suppress any sound signals with these frequencies would prevent Dolphin Attack completely. Alternatively, they suggest using inaudible voice command cancellation which would work with legacy microphones [4].

9.9 Light Commands Mitigation

To protect against the Light Commands and other similar attacks where an adversary cannot directly hear or eavesdrop on the smart speaker, researchers suggested adding an additional layer of authentication which would ask a user a simple randomised question [42]. They also point out that in the case of a Light Commands attack, only one microphone will detect a malicious input signal. This, however, is not normal as the signal is typically similar on all microphones. This could be used to effectively detect most Light Commands attacks[1]. They also suggest a range of hardware solutions which would hinder the laser waves, but they also added that an attacker can always use a laser to burn through these barriers.

[1]The researchers showed that the attack is successful even with a wider-beam flashlight [5], but this would make an attack much more obvious if executed from afar, lowering its feasibility.

9.10 Other Security Features

During my testing and research, I came across some other features that could be implemented to enhance security, privacy, and in some cases even user experience. These are:

- User voice recordings that are considered sensitive could be automatically deleted [26].
- If Alexa is not more than, say, 90% sure which similar-sounding word (e.g. "tale" or "tail") has been said, it could ask the user something like: "Do you mean 'tale' as a story or 'tail' as an animal body part?" This could reduce the mistakes in invocation as well as provide an even more personal feel to Alexa.
- To improve the PIN lockout mechanism, users would have to re-enable the purchasing from the App whenever the PIN is incorrectly entered more than three times. Instead of using a 4-digit PIN, Amazon could also allow the use of passphrases which are easier to remember while offering a much greater entropy [19].
- After a command is transcribed, the audio recording should be encrypted first and then stored in a way that only users will be able to listen to their own recordings. Amazon could then request users to correctly transcribe any command that Alexa got wrong, which would then be sent to the speech recognition model for learning. This would happen automatically and both the audio recording and the transcription will be discarded after the learning step. This would restrict Amazon employees from accessing the users' voice recordings while enabling the voice recognition models to keep learning.
- Amazon should disallow URLs to appear in users' lists as having URLs in the lists is an unlikely use case. Alternatively, Amazon could prompt the users before they open a URL and ask them if they trust that domain. A user would then have to explicitly agree to opening such URLs.
- *Intents* that expect an answer or a follow-up from the user should not contain words like "password" and "email" in their prompts. This would possibly prevent the skill from asking for sensitive data directly.
- Amazon could provide an option for a user to lock their Alexa devices. To unlock them they would have to provide a PIN or a passphrase (that can be different from the PIN used for payments). Since this can be done through the voice interface, it will potentially be more useful than having a physical mute button on the device.

10 Conclusion and Future Work

When I started this project, I assumed Alexa would turn out to be an overall a fairly secure system because it is developed by a big tech company. However, I discovered that certain design decisions made by Amazon exposed Alexa to several medium to high severity attacks, with vishing and phishing attacks being two of the biggest threats. It seems that a lot of these decisions were based on the trade-off between usability, privacy and security, where the latter two usually came second. A lot of vulnerabilities also result from the user's lack of knowledge about how smart assistants work. Because such systems only reached a broader public in the last decade, many people are still coping with entirely grasping these new design principles.

On the other hand, Amazon is, like many other tech giants, a big data collecting and crunching machine and until they offer more transparency over what actually happens behind the scenes there will always be some reluctance from fully trusting Alexa.

Future Work

The future research could build on this security assessment and explore some more advanced potential vulnerabilities that I did not have a chance to explore due to the time constraints. One thing to research further is how the Android OS that the Dot is based on could be exploited. Similarly, the Alexa App can be linked with many other smart home devices and it is unclear how secure this linkage is.

Another potential vulnerability lies in unencrypted over-the-air (OTA) firmware updates for Alexa devices. It will need to be verified whether these updates are cryptographically signed by Amazon. If they are, then it is unlikely that modifying this update binary would yield any success. Potentially, if only some random bits are changed in this binary, it might result into a faulty update which could "brick" the device.

Since the skills' backend is often built with programming languages that allow the usage of various open source dependencies and libraries, it will be interesting to check how a vulnerable dependency could leak any user data.

Parents can enable parental controls (called FreeTime) on Alexa which will allow children to use Alexa without accidentally accessing the content that is not appropriate for them [85]. Unfortunately, this feature was not available on my device, so it remains to be tested.

Another feature that was not available on my device is memory [7]. This enables users to tell Alexa what to remember so they can retrieve this information later. It will be important to verify if Alexa can tell which user told her to remember what and if users can retrieve each other's information.

References

[1] Hyunji Chung, Jungheum Park, and Sangjin Lee. Digital forensic approaches for Amazon Alexa ecosystem. In *Proceedings of the Seventeenth Annual DFRWS USA*, 2017. URL https://www.sciencedirect.com/science/article/pii/S1742287617301974.

[2] Mary K. Bispham, Ioannis Agrafiotis, and Michael Goldsmith. Nonsense attacks on Google Assistant and missense attacks on Amazon Alexa. volume 1, pages 75–87. SciTePress, 2019. URL https://ora.ox.ac.uk/objects/uuid:22a26ee8-9c36-4dda-878f-09f3ec32cc9c.

[3] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition. Technical report, August 2015. URL https://www.usenix.org/system/files/conference/woot15/woot15-paper-vaidya.pdf.

[4] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. DolphinAttack: Inaudible Voice Commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 103–117, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3134052. URL https://doi.org/10.1145/3133956.3134052.

[5] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. Light Commands - Website. *LightCommands*, 2019. URL https://lightcommands.com/.

[6] Hang Hu, Limin Yang, Shihan Lin, and GangWang. Security Vetting Process of Smart-home Assistant Applications: A First Look and Case Studies. *arXiv preprint arXiv:2001.04520v1*, 2020. URL https://arxiv.org/abs/2001.04520.

[7] Ry Crist and Andrew Gebhart. Everything you need to know about the Amazon Echo. *CNet*, September 2018. URL https://www.cnet.com/how-to/amazon-echo-alexa-everything-you-need-to-know/.

[8] Brandon Vigliarolo. Amazon Alexa: Cheat sheet. *TechRepublic*, September 2019. URL https://www.techrepublic.com/article/amazon-alexa-the-smart-persons-guide/.

[9] Holt Spalding. Investigation Of Amazon Alexa's Explicit Invocation Policy. Technical report, December 2018. URL http://www.cs.tufts.edu/comp/116/archive/fall2018/hspalding.pdf.

[10] Candid Wueest. A guide to the security of voice-activated smart speakers (An ISTR Special Report). Technical report, November 2017. URL https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-security-voice-activated-smart-speakers-en.pdf.

[11] David Priest. 12 new Alexa features to up your Amazon Echo game. *CNet*, October 2019. URL https://www.cnet.com/how-to/new-alexa-features-to-up-your-amazon-echo-game/.

[12] Abdulaziz Alhadlaq, Jun Tang, Marwan Almaymoni, and Aleksandra Korolova. Privacy in the Amazon Alexa Skills Ecosystem. Technical report, March 2019. URL https://www.petsymposium.org/2017/papers/hotpets/amazon-alexa-skills-ecosystem-privacy.pdf.

[13] Kyle Wiggers. The Alexa Skills Store now has more than 100,000 voice apps. *Venture Beat*, September 2019. URL https://venturebeat.com/2019/09/25/the-alexa-skills-store-now-has-more-than-100000-voice-apps/.

[14] Erika Rawes. Amazon Echo vs. Echo Dot. *Digital Trends*, November 2019. URL https://www.digitaltrends.com/home/amazon-echo-vs-dot/.

[15] Jide S. Edu, Jose M. Such, and Guillermo Suarez-Tangil. Smart Home Personal Assistants: A Security and Privacy Review. Technical report, March 2019.

[16] Adam Palanica, Anirudh Thommandram, Andrew Lee, Michael Li, and Yan Fossat. Do you understand the words that are comin outta my mouth? Voice assistant comprehension of medication names. *NPJ digital medicine*, 2(1):1–6, 2019. URL https://www.nature.com/articles/s41746-019-0133-x.

[17] Lee Assam. Learning to Build Alexa Skills, February 2019. URL https://www.linkedin.com/learning/learning-to-build-alexa-skills.

[18] Amazon. Amazon Alexa developer website, 2020. URL https://developer.amazon.com/en-US/alexa.

[19] William Haack, Madeleine Severance, Michael Wallace, and Jeremy Wohlwend. Security Analysis of the Amazon Echo. Technical report, May 2017. URL https://pdfs.semanticscholar.org/35c8/47d63db1dd2c8cf36a3a8c3444cdeee605e4.pdf.

[20] Youngseok Park, Hyunsang Choi, Sanghyun Cho, and Young-Gab Kim. Security Analysis of Smart Speaker: Security Attacks and Mitigation. *Computers, Materials & Continua*, 61(3):1075–1090, 2019. ISSN 1546-2226. doi: 10.32604/cmc.2019.08520. URL http://www.techscience.com/cmc/v61n3/35289.

[21] Matthew B Hoy. Alexa, Siri, Cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88, 2018. URL https://www.tandfonline.com/doi/full/10.1080/02763869.2018.1404391.

[22] Marcia Ford and William Palmer. Alexa, are you listening to me? An analysis of Alexa voice service network traffic. *Personal and Ubiquitous Computing*, 23(1):67–79, 2019. URL https://link.springer.com/article/10.1007/s00779-018-1174-x.

[23] Atsuko Natatsuka, Ryo Iijima, Takuya Watanabe, Mitsuaki Akiyama, Tetsuya Sakai, and Tatsuya Mori. Poster: A First Look at the Privacy Risks of Voice Assistant Apps. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 2633–2635, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3363274. URL https://doi.org/10.1145/3319535.3363274.

[24] Hyunji Chung, Michaela Iorga, Jeffrey Voas, and Sangjin Lee. Alexa, can I trust you? *Computer*, 50(9):100–104, 2017. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8048642&tag=1.

[25] Peng Cheng, Ibrahim Ethem Bagci, Jeff Yan, and Utz Roedig. Smart Speaker privacy control-acoustic tagging for Personal Voice Assistants. In *IEEE Workshop on the Internet of Safe Things (SafeThings 2019)*, 2019. URL https://cora.ucc.ie/handle/10468/8396.

[26] Nathan Malkin, Joe Deatrick, Allen Tong, Primal Wijesekera, Serge Egelman, and David Wagner. Privacy Attitudes of Smart Speaker Users. *Proceedings on Privacy Enhancing Technologies*, 2019(4):250 – 271, 2019. URL https://content.sciendo.com/view/journals/popets/2019/4/article-p250.xml.

[27] David J Major, Danny Yuxing Huang, Marshini Chetty, and Nick Feamster. Alexa, Who

Am I Speaking To? Understanding Users' Ability to Identify Third-Party Apps on Amazon Alexa. *arXiv preprint arXiv:1910.14112*, 2019. URL https://arxiv.org/abs/1910.14112.

[28] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. Alexa, Are You Listening? Privacy Perceptions, Concerns and Privacy-Seeking Behaviors with Smart Speakers. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW), November 2018. doi: 10.1145/3274371. URL https://doi.org/10.1145/3274371.

[29] Graeme McLean and Kofi Osei-Frimpong. Hey Alexa ... examine the variables influencing the use of artificial intelligent in-home voice assistants. *Computers in Human Behavior*, 99:28 – 37, 2019. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2019.05.009. URL http://www.sciencedirect.com/science/article/pii/S0747563219301840.

[30] Rianna R. Baeza and Anil R. Kumar. Perceived Usefulness of Multimodal Voice Assistant Technology. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 63 (1):1560–1564, 2019. doi: 10.1177/1071181319631031. URL https://doi.org/10.1177/1071181319631031.

[31] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. Skill Squatting Attacks on Amazon Alexa. Technical report, August 2018. URL https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kumar.pdf.

[32] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1381–1396. IEEE, 2019. URL https://ieeexplore.ieee.org/abstract/document/8835332.

[33] Eoghan Furey and Juanita Blue. She Knows Too Much – Voice Command Devices and Privacy. In *2018 29th Irish Signals and Systems Conference (ISSC)*, pages 1–6. IEEE, 2018. URL https://ieeexplore.ieee.org/abstract/document/8585380.

[34] Xinyu Lei, Guan-Hua Tu, Alex X Liu, Kamran Ali, Chi-Yu Li, and Tian Xie. The Insecurity of Home Digital Voice Assistants – Amazon Alexa as a Case Study. *arXiv preprint arXiv:1712.03327*, 2017. URL https://arxiv.org/abs/1712.03327.

[35] Ana Berdasco, Gustavo López, Ignacio Diaz, Luis Quesada, and Luis A Guerrero. User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and

Cortana. In *Multidisciplinary Digital Publishing Institute Proceedings*, volume 31, page 51, 2019. URL https://www.mdpi.com/2504-3900/31/1/51.

[36] Toine Bogers, Ammar Ali Abdelrahim Al-Basri, Claes Ostermann Rytlig, Mads Emil Bak Møller, Mette Juhl Rasmussen, Nikita Katrine Bates Michelsen, and Sara Gerling Jørgensen. A Study of Usage and Usability of Intelligent Personal Assistants in Denmark. In Natalie Greene Taylor, Caitlin Christian-Lamb, Michelle H. Martin, and Bonnie Nardi, editors, *Information in Contemporary Society*, pages 79–90, Cham, 2019. Springer International Publishing. ISBN 978-3-030-15742-5. URL https://link.springer.com/chapter/10.1007%2F978-3-030-15742-5_7.

[37] İlkan Yıldırım, Erkan Bostancı, and Mehmet Serdar Güzel. Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and Google Assistant Build-In Smart Home Speakers. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pages 1–3. IEEE, 2019. URL https://ieeexplore.ieee.org/abstract/document/8907007.

[38] Catherine Jackson and Angela Orebaugh. A study of security and privacy issues associated with the Amazon Echo. *International Journal of Internet of Things and Cyber-Assurance*, 1(1):91–100, 2018. URL https://pdfs.semanticscholar.org/e80b/6646a8d6c5a6e4b8904db11d8115e83c6b09.pdf.

[39] Liwei Song and Prateek Mittal. POSTER: Inaudible Voice Commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 2583–2585, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3138836. URL https://doi.org/10.1145/3133956.3138836.

[40] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. DolphinAttack: Inaudible Voice Command, August 2017. URL https://youtu.be/21HjF4A3WE4.

[41] Andy Greenberg. Hackers Can Use Lasers to 'Speak' to Your Amazon Echo or Google Home. *Wired*, November 2019. URL https://www.wired.com/story/lasers-hack-amazon-echo-google-home/.

[42] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. Light

Commands: Laser-Based Audio Injection Attacks on Voice-Controllable Systems. November 2019. URL https://lightcommands.com/20191104-Light-Commands.pdf.

[43] Jason Creasey and Principal reviewer. A guide for running an effective Penetration Testing programme. Technical report, April 2017. URL https://www.crest-approved.org/wp-content/uploads/CREST-Penetration-Testing-Guide-1.pdf.

[44] Matteo Meucci and Andrew Muller. OWASP Testing guide – Version 4.0. *OWASP Foundation*, 2014. URL https://www.owasp.org/images/1/19/OTGv4.pdf.

[45] The OWASP Foundation. OWASP Risk Rating Methodology. 2020. URL https://owasp.org/www-community/OWASP_Risk_Rating_Methodology.

[46] Lisa Bock. Ethical Hacking: Introduction to Ethical Hacking, February 2019. URL https://www.linkedin.com/learning/ethical-hacking-introduction-to-ethical-hacking/.

[47] Lisa Bock. Ethical Hacking: Penetration Testing, November 2016. URL https://www.linkedin.com/learning/ethical-hacking-penetration-testing/.

[48] Malcolm Shore. Penetration Testing Essential Training, October 2017. URL https://www.linkedin.com/learning/penetration-testing-essential-training.

[49] Heath Maverick Adams. Full Ethical Hacking Course – Network Penetration Testing for Beginners, July 2019. URL https://youtu.be/3Kq1MIfTWCE.

[50] iFixit. Gadget Guts: Amazon Echo Dot (2nd Generation), 2017. URL https://www.youtube.com/watch?v=2m5ra4hiX-Q.

[51] DriverScape. MediaTek MT65xx Preloader Drivers Download, 2020. URL https://www.driverscape.com/download/mediatek-mt65xx-preloader.

[52] Doug G. Trouble connecting to composite USB serial port – Reply 3, November 2016. URL https://www.linuxquestions.org/questions/linux-hardware-18/trouble-connecting-to-composite-usb-serial-port-4175594113/.

[53] Alex Vanderpot. Echohacking Wiki (on GitHub), June 2017. URL https://github.com/echohacking/wiki/wiki.

[54] Amazon. Alexa Skills Store (UK), . URL https://www.amazon.co.uk/b?ie=UTF8&node=10068517031.

[55] Amazon. Alexa Skills Store (US), . URL https://www.amazon.com/alexa-skills/b?ie=UTF8&node=13727921011.

[56] Amazon. *Alexa Skills Kit*, 2020. URL https://developer.amazon.com/en-US/alexa/alexa-skills-kit.

[57] Amazon. *Slot Type Reference*, 2020. URL https://developer.amazon.com/en-US/docs/alexa/custom-skills/slot-type-reference.html.

[58] Amazon. *Amazon Alexa developer website – Request and Response JSON Reference*, . URL https://developer.amazon.com/en-US/docs/alexa/custom-skills/request-and-response-json-reference.html.

[59] Weikeng Chen. Is RSA-SHA1 signature still considered safe?, 2018. URL https://crypto.stackexchange.com/questions/60619/after-googles-collision-attack-is-rsa-sha1-signature-still-safe.

[60] Wired. 8 People Test Their Accents on Siri, Echo and Google Home, May 2017. URL https://youtu.be/gNx0huL9qsQ.

[61] Wired. 8 Kids Test Their Speech on Siri, Echo and Google Home, November 2017. URL https://youtu.be/GZnUibN6m4A.

[62] Amazon. Using Household Profiles on Alexa Devices, 2019. URL https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=201628040.

[63] Barbara Krasnoff. How to use your Echo with two Amazon accounts. *The Verge*, November 2019. URL https://www.theverge.com/2019/11/19/20970798/amazon-echo-alexa-household-accounts-adults-kids-how-to-add-use.

[64] Alexa's voice apps for kids can now offer purchases that parents approve, June 2019. URL https://techcrunch.com/2019/06/14/alexas-voice-apps-for-kids-can-now-offer-purchases-that-parents-approve/.

[65] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. Understanding and Mitigating the Security Risks of Voice-Controlled Third-Party Skills on Amazon Alexa and Google Home. Technical report, June 2018. URL https://arxiv.org/pdf/1805.01525.pdf.

[66] Security Research Labs. Smart Spies: Alexa and Google Home expose users to vishing and eavesdropping. *Security Research Labs*, October 2019. URL https://srlabs.de/bites/smart-spies/.

[67] Avisoft Bioacoustics. Ultrasonic Dynamic Speaker Vifa, 2020. URL http://www.avisoft.com/playback/vifa/.

[68] Avisoft Bioacoustics. Price List, 2020. URL http://www.avisoft.com/price-list-ordering-information/.

[69] Fostex. FT17H Horn Tweeter, 2020. URL https://www.fostexinternational.com/docs/speaker_components/pdf/ft17hrev2.pdf.

[70] Fostex. FT17H Horn Tweeter, 2020. URL https://www.madisoundspeakerstore.com/bullet-tweeters/fostex-ft17h-horn-super-tweeter/.

[71] KKmoon. Signal Generator by KKmoon, 2020. URL https://www.amazon.com/KKmoon-Precision-Dual-Channel-819214bits-Modulation/dp/B0779Z7RC8/ref=sr_1_1?dchild=1&keywords=signal+generator+modulator&qid=1587135910&sr=8-1.

[72] Yamaha Corporation. R-S202 Natural Sound Stereo Receiver, 2020. URL https://usa.yamaha.com/products/audio_visual/hifi_components/r-s202/index.html.

[73] Liwei Song and Prateek Mittal. Inaudible Voice Commands, 2017. URL https://www.youtube.com/watch?v=wF-DuVkQNQQ.

[74] Smarter Every Day. Breaking Into a Smart Home With A Laser – Smarter Every Day 229, 2019. URL https://www.youtube.com/watch?v=ozIKwGt38LQ.

[75] Amazon. *Configure Permissions for Customer Information in Your Skill*, 2020. URL https://developer.amazon.com/en-US/docs/alexa/custom-skills/configure-permissions-for-customer-information-in-your-skill.html.

[76] Amazon. *AMAZON.EmailAddress Reference*, 2020. URL https://docs.aws.amazon.com/lex/latest/dg/built-in-slot-email.html.

[77] Amazon. *Best Practices for Skill Card Design*, 2020. URL https://developer.amazon.com/en-US/docs/alexa/custom-skills/best-practices-for-skill-card-design.html.

[78] Amazon UK – Homepage, 2020. URL https://www.amazon.co.uk/.

[79] Echo (3rd Gen) – Smart speaker with Alexa, 2020. URL https://www.amazon.com/all-new-Echo/dp/B07R1CXKN7.

[80] Matthew Wetherell. Hi-Fi Digital Audio from the Echo Dot, 2018. URL https://hackaday.io/project/28109-hi-fi-digital-audio-from-the-echo-dot.

[81] Conner Forrest. How one simple hack can turn your Amazon Echo into a spy device. *TechRepublic*, August 2017. URL https://www.techrepublic.com/article/how-one-simple-hack-can-turn-your-amazon-echo-into-a-spy-device/.

[82] Dan Goodin. Alexa and Google Home abused to eavesdrop and phish passwords. *Ars Technica*, October 2019. URL https://arstechnica.com/information-technology/2019/10/alexa-and-google-home-abused-to-eavesdrop-and-phish-passwords/.

[83] Charlotte Jee. Smart speakers can be hijacked by apps that spy on users. *MIT Technology Review*, October 2019. URL https://www.technologyreview.com/f/614602/smart-speakers-can-be-hijacked-by-apps-that-spy-on-users/.

[84] Sharon Profis and Rick Broida. You can finally delete (most of) your Amazon Echo transcripts. Here's how. *CNet*, July 2019. URL https://www.cnet.com/how-to/you-can-finally-delete-most-of-your-amazon-echo-transcripts-heres-how/.

[85] Taylor Martin. How to use Alexa's parental controls. *CNet*, September 2018. URL https://www.cnet.com/how-to/how-to-enable-alexa-parental-controls/.

A Spy Bug Schematics

Here I included precise schematic and a wiring diagram for the Spy Bug. Both were created with Fritzing[1].

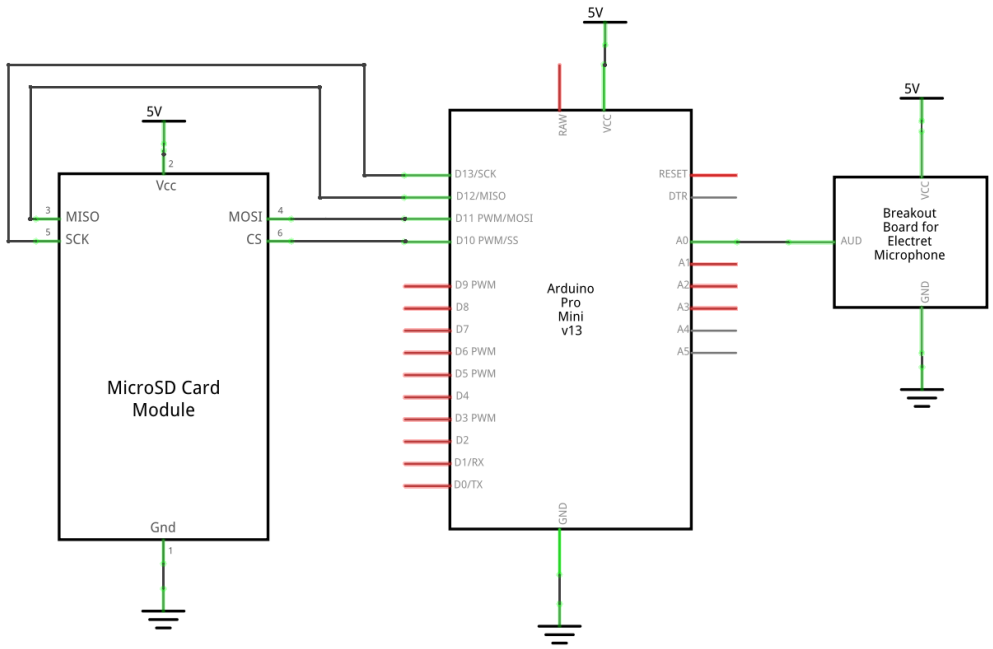

Figure A.1: A precise Spy Bug schematic.

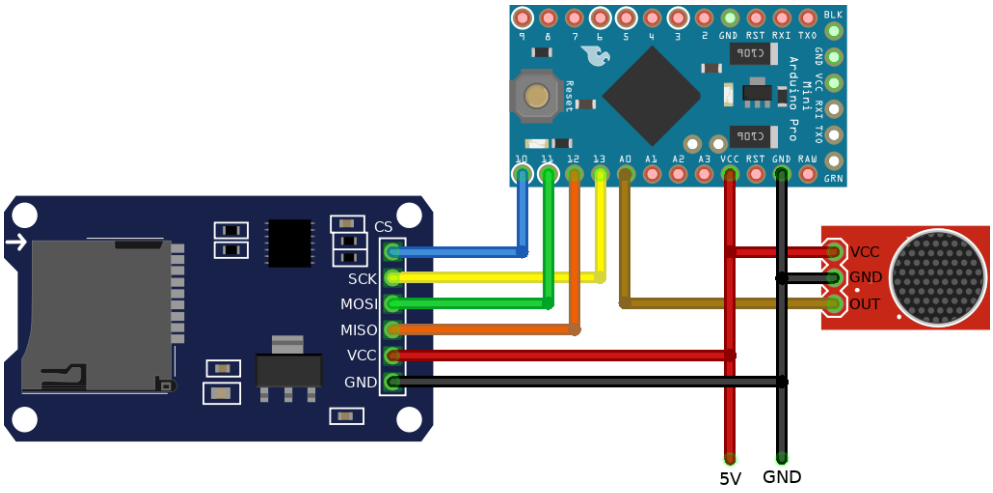

Figure A.2: A wiring diagram for the Spy Bug.

[1] https://fritzing.org/