Security Analysis of the Amazon Echo

William Haack
Madeleine Severance
Michael Wallace
Jeremy Wohlwend

May 18, 2017

Abstract

It is currently estimated that more that eight billion devices are connected to the internet worldwide. Since 2016, this number has increased by more than thirty percent and is predicted to continue growing steadily in the next few years. Due to its wide spread, the Internet of Things has redefined the way that we interact with technology, raising important questions about privacy and integrity. This paper provides a security analysis of the Amazon Echo, one of the most widely used IoT devices. We first describe the Amazon Echo and provide an ideal security policy for the device. We then attempt to identify vulnerabilities that go against the policy, by trying sound and network based attacks as well as attacks based on the Echo's API. We show that the Amazon Echo provides satisfiable security, but that the device can still, under particular conditions, be taken advantage of. Finally, a set of recommendations is provided, that addresses the different attacks and their impact.

1 Introduction

Amazon's Alexa is marketed as an intelligent, digital personal assistant. Users of this assistant service interact with it using the Amazon Echo or Amazon Echo Dot devices. These devices register users' voice commands and convey them to the Alexa service. Using these devices, a user can perform a variety of useful tasks, including but not limited to

- retrieving real-time weather, traffic, news and other information,
- streaming music and other media,
- controlling smart-home devices,
- ordering products from Amazon's online store,
- interaction with third-party applications (for example, the ability to order a car from a ride-sharing application, or order a meal from a food-ordering application) [1].

The Alexa service launched in November 2014 and although Amazon does not release sales data, [3] it estimates that eight million Echo devices are being used in the United States.

Given the sensitive information handled by the Amazon Alexa service – including payment information and third-party application credentials – as well as the widespread and growing adoption of Amazon's Echo devices mean that a successful security breach of these devices could have seriously negative consequences. For this reason, it is interesting to review the security of the Amazon Echo device. We begin with an overview of the Amazon Alexa and Echo system architecture, then present an ideal security policy for the Amazon Echo. Following this, we report on a number of attempts to breach the security of the Echo device, including sound-based attacks, packet-based attacks, attacks that use direct interaction with the Amazon Alexa API, and attacks using malicious third-party applications.

2 Amazon Alexa & Echo Architecture Overview

When users access the Amazon Alexa service via Echo devices, there are a number of system components that interface to produce meaningful responses to voice commands. A summary of these components and their interactions are outlined in Figure 1. Furthermore, we give a more detailed description of each component in the subsequent sections.

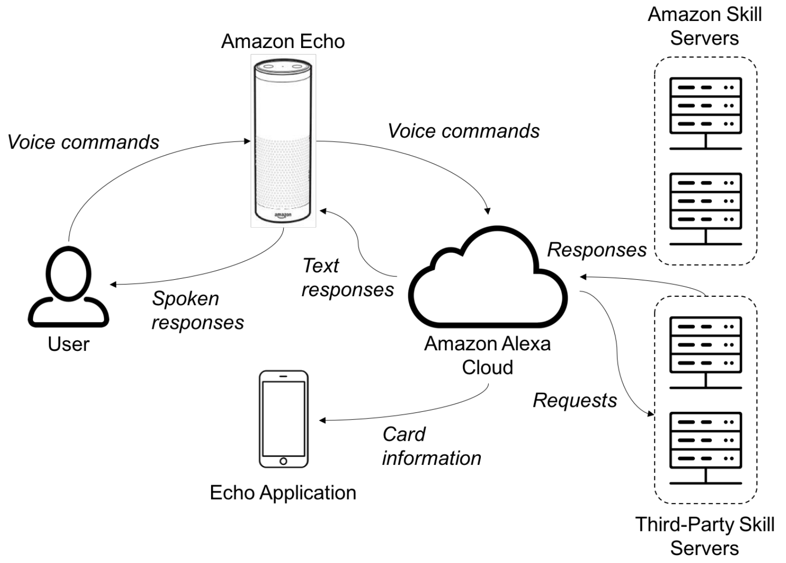

Figure 1: Summary of Amazon Echo system architecture.

2.1 Echo

The Echo (or Echo Dot) device provides a speech-based interface for the user – the user issues spoken commands or asks questions, and the Echo responds with speech. The Echo is always listening for speech, and only when it detects the *wake word* ("Alexa", by default), does it transmit the subsequent audio (presumably the user's command or question) to the Alexa cloud service. It then waits to receive a text-based response from the cloud, then renders and plays the audio for the spoken response.

2.2 Alexa Cloud

The Alexa cloud service receives audio from Echo devices and is responsible for handling speech recognition and mapping voice commands to *intents* and associated data. Based on the determined intent, the Alexa cloud service delegates the command to a skills server, operated by either Amazon for built-in skills or by a third-party for third-party application skills. When the Alexa cloud service receives a response from the skill server, it conveys this back to the Echo device.

2.3 Amazon Skill Servers

Skills are essentially tasks that the Alexa service may perform for a user, such as checking the weather, or ordering a pizza. Some skills are offered by Amazon and come as built-in functionality for all Amazon Alexa users, and these are handled by Amazon's own skill servers. Requests can be made to these servers by the Amazon cloud service and these servers then provide responses, both in predetermined formats.

2.4 Third-Party Skill Servers

Other skills are offered by third-party developers, such as food-ordering application or ride-sharing applications. These skills are handled by the third-party developers themselves, and the request/response process is almost identical to that for Amazon's skill servers.

2.5 Echo Application

Amazon offers a mobile application that is used to manage the Alexa service. With this application, users can adjust settings for Alexa, see all past interactions with Alexa, and give feedback on the quality of these interactions. When a user interacts with Alexa, the Alexa cloud service pushes *cards* to the mobile application, which contain the relevant information about the interaction to be displayed in the application. Note that this information includes a recording of the user's question or command to their Echo device.

3 Ideal Amazon Echo Security Policy

3.1 Objective

We would like to outline an ideal security policy for the Amazon Echo in this section in order to guide our investigations. With the use of an ideal security policy, we will be able to discern between allowable and unallowable actions and determine what constitutes a vulnerability. If we find vulnerabilities that breach this ideal policy, we will recommend strategies for protection or prevention against them. Our security policy is defined in terms of principals and the actions they can and cannot take in order for the Echo to provide confidentiality, integrity, and availability.

3.2 Security Goals

Our security policy aims to achieve the goals of **confidentiality**, **integrity**, and **availability** as they apply specifically to the Amazon Echo. Sensitive information should not be available to unauthorized parties, it should be protected from unauthorized changes, and the Echo should be ready for use at all times. Applications of these goals specific to the Echo are described below.

3.2.1 Confidentiality

In order to achieve confidentiality, Echo users' sensitive information should not be revealed to unauthorized individuals. Amazon Alexa accounts are linked to Amazon accounts that contain sensitive information like personal contact information, payment and credit card accounts, date of birth, and addresses of residence. With multiple users accessing a single Echo, this information should not be disclosed to any user who is not the owner of this information. Moreover, users should not be able to obtain other users' Amazon or Alexa account credentials or linked third-party account credentials through the Echo.

3.2.2 Integrity

While protecting users' sensitive information is critical, it is not enough when forming an ideal security policy. Just as important of a goal as confidentiality is integrity, preventing users from modifying data in an unauthorized manner. This may be even more crucial when analyzing the Echo, as multiple users may be able to perform certain actions through a single user's account and hardware. Unauthorized users should not be able to make purchases through another user's account, change shipping addresses, or modify account credentials. Adversaries should not be able to modify any traffic coming from or going to the Echo. Any actions dealing with sensitive information should be protected from being performed by unauthorized individuals.

3.2.3 Availability

Lastly, the Echo and its functionalities should remain available to the user. No user, adversary, or unauthorized Amazon employee should be able to remove Echo functionalities, prevent users from accessing their accounts, block Echo speech detection and interpretation, or distort or mute Echo responses. Adversaries should not be able to divert or block incoming or outgoing traffic from the Echo or perform denial of service attacks.

3.3 Principals and Actions

With the above goals in mind, we outline the principals and actions they can and cannot take. The relevant principals who may interact with any given Echo can be classified into four categories.

Primary and **Secondary Users** are those whose Amazon Alexa accounts are registered with the given Echo, whereas **Other Users** are those who may use the Echo but whose accounts are not registered with it. The final category is **Amazon Affiliates**, those who work at Amazon and therefore may have access to a given Echo's information. We further define these principals and the actions they should and should not be able to perform in the subsequent sections.

3.3.1 Primary Users

A primary user is defined as the primary owner of the Echo, whose account is first registered with the Echo. These users have control over how they would like to use their Echo, what personal information they provide it, and who else may use it. Their allowable and unallowable actions are further detailed below:

- Allowable actions
  - Register own Amazon account with Echo
  - Determine if others can register their Amazon accounts with Echo
  - Use Echo to access sensitive information such as information dealing with payments, home security, personal contact information, etc
  - Use Echo to access non-sensitive information such as weather, music, internet searches, etc
  - Protect sensitive information from other users
- Unallowable actions
  - Should not be able to access Amazon's sensitive information such as others' accounts, payment methods, intellectual property, etc
  - Should not be able to access third-party skills' sensitive information via the Echo
  - Should not be able to perform illegal actions via the Echo (i.e. online shoplifting, credit card fraud, etc)

3.3.2 Secondary Users

A secondary user is someone who registers their Amazon account with an Echo and is not a primary user. Secondary users should only be allowed to register their accounts with an Echo if the primary user of that Echo gives them permission to do so. The primary user should not be able to access the secondary user's sensitive information without permission, however the primary user should be able to remove the secondary user's account at any time. The following outlines their allowable and unallowable actions:

- Allowable actions
  - With primary user's permission, register own Amazon account with Echo
  - Use Echo to access sensitive information such as information dealing with payments, home security, personal contact information, etc
  - Use Echo to access non-sensitive information such as weather, music, internet searches, etc
  - Protect sensitive information from primary user and other users
- Unallowable actions
  - The same unallowable actions for the primary user
  - Access primary user or other secondary users' sensitive information without permission
  - Remove primary user or other secondary users from the Echo

3.3.3 Other Users

Other users are those who interact with the Echo without registering their accounts with it. The actions they can and cannot perform are outlined below:

- Allowable actions
  - Use Echo to access non-sensitive information such as weather, music, internet searches, etc
- Unallowable Actions
  - The same unallowable actions as the secondary user

3.3.4 Amazon Affiliates

Amazon affiliates are those who work for Amazon and may have access to Amazon proprietary information or Amazon users' information. Amazon affiliates should not be able to access any Echo users' sensitive information.

4 Security Testing

With an ideal security policy in place, we can perform a security analysis of the Echo. We review design and implementation decisions made by Amazon and assess whether they breach this ideal policy. Our attacks fall into four main channels: **sound**, **network**, **direct API**, and **third-party skills**. For each of the following attacks, we describe our justification for the attack, method, results, and implications. However, before delving into attacks of our own, we first summarize and attempt to retry attacks previously made by others.

4.1 Previous Work

While there isn't any concrete security review of the Amazon Echo in the literature, a few online posts provided some background information about the device and potential attacks. In particular [5, 6] were able to root the device through a local proxy and watch the traffic coming out the of the echo. This procedure is outlined in section 4.3. Most of the traffic is takes the form of HTTPS requests, which are encrypted, but a few HTTP requests can be found in the mix. One of them, as explained in [5] provides an update binary which contains the Android based OS ran on the Echo, and corresponding libraries. This is important, because if Android vulnerabilities were to be uncovered, it could put the amazon echo at risk. Another interesting thing to note is that the update name contains the word kindle, implying that the OS used on the Kindle tablets may be similar to the one used on the echo. Again, this opens the door to other potential indirect vulnerabilities. It is unclear how much can be achieved by trying to trick the echo into using a corrupted update file. Triggering the echo to update is a problem of itself, and there is no guarantee that the echo will accept an update not signed by amazon, though that remains to be tested.

Others [8, 9] provide a description of the hardware components of the device, including a Texas Instrument processing chip, an analog to digital converter, a bluetooth chip, and memory drivers. The devices runs a version of Android, which implies that the Echo's security may also be affected by Android vulnerabilities.

Another area which has been previously studies are sound based attacks such as the ones described in [4]. These attacks were performed on a phone using the Google assistant but the same principles apply. The authors found that it was possible for the device to understand commands that were impossible to comprehend for a human observant. From a security perspective, this may allow an attacker to pass secret commands to the device, through radio or TV advertising, for instance.

4.2 Sound-based Attacks

The intended form of communication with the Amazon Echo is sound. The Echo knows when to listen based on when it hears its wake word, one of 'Alexa', 'Amazon', 'Echo', or 'Computer' designated by the user. It then records and processes what the user says in the cloud and documents the speech in the user's Alexa application. We tried several different sound attacks and found in general there is little limit as to who or what can provide sound to Alexa and as to the number of times the same sound can be repeated, provided that the sounds pertain to one of Alexa's capabilities.

4.2.1 Modes of Attack

Minimal exploration of the Echo reveals its limited capabilities. Alexa is able to accomplish a few set things, without the use of third-party skills, including ordering items through Amazon, checking the weather, setting alarms, creating to-do lists, and flipping a hypothetical coin. When presented with speech outside of these realms, Alexa will say she does not understand and will stop listening. This stopped us from pursuing a SQL injection attack. Once we learned Alexa did not understand, "Alexa, drop table orders," we concluded we would likely have the same response to other attempts at SQL injection attacks. Alexa will also not reveal or allow users to modify personal information like email or shipping addresses.

The first alternate speech method we tried was speaking in different languages and accents. Alexa does not understand languages other than English but is usually able to decipher British and Australian accents. This negatively affects the Echo's availability, as its use is limited to proficient English speakers.

Alexa does respond to indirect speech. She will execute commands delivered via voicemail, voice recording, or computer generation. Moreover, she will repeatedly respond to the same commands, even if those commands are produced from the exact same sound file. We replayed the sounds,"Alexa, what is the weather?" and "Alexa, flip a coin", and she answered our questions each time. Alexa is also able to hear sound that passes through barriers. She was able to process what we yelled to her through a thick wooden door. The implications of these results are detailed below.

4.2.2 Implications

Through our experiments we can see that Alexa can be influenced by people outside the room in which she sits. Thus, potential attackers do not have to be inside the home; they could easily yell through a window or door or leave a voice message in order to communicate with Alexa. Additionally, the Echo has no voice differentiation capabilities nor protection against non-human or repeated speech. This makes it more difficult for the Echo to distinguish between primary, secondary, and other users. It also allows adversaries to automate their attacks.

One way that Amazon authorizes users for restricted tasks is by requiring a 4-digit PIN. Having to enter a 4-digit PIN to place an Amazon order can be turned on in the user's settings. Notably, the default settings for the Echo are to allow voice-enabled purchases without having to enter a 4-digit PIN. This design decision by Amazon could compromise integrity, as unauthorized users are able to place orders for another user's account.

We explored a brute-force attack on the 4-digit PIN for orders. We wrote the script shown in Appendix B, which caused the computer connected to external speakers to attempt to order an Amazon Echo Dot by trying every PIN from 0000 to 9999. The events, including Alexa's responses, occur in the following order:

1. Computer says wake word followed by the command to order an Amazon Echo Dot
2. Alexa responds with top Amazon search for "Amazon Echo Dot" and asks if user wants to place the order
3. Computer confirms order
4. Alexa asks for 4-digit PIN
5. Computer guesses next PIN in numerical order
6. Alexa accepts or rejects PIN
7. Computer guesses next PIN in numerical order

This process repeats for all possible PINs. Alexa only allows the user to enter two PINs, after which the user must restart the ordering process. However, Alexa does not limit the number of times a user can attempt to order the same item. Each iteration to guess two PINs takes about

30 seconds, so in order to exhaust the entire PIN space of 10,000 PINs, the program would need to run for at most 41 hours and 40 minutes.

However, for many users their 4-digit PIN is not randomly generated. According to [7], 1234, 1111, 0000, and 1212 are the most common PINs accounting for about 20% of users' PINs. The 20 most frequently used PINs cover about 27% of used PINs. These top 20 PINs are shown in Table 1. Additionally, birth dates in the form MMDD and birth years are other commonly used PINs. Thus, the expected time to correctly guess a user's PIN can be significantly reduced.

PIN	Frequency
1234	10.713%
1111	6.016%
0000	1.881%
1212	1.197%
7777	0.745%
1004	0.616%
2000	0.613%
4444	0.526%
2222	0.516%
6969	0.512%
9999	0.451%
3333	0.419%
5555	0.395%
6666	0.391%
1122	0.366%
1313	0.304%
8888	0.303%
4321	0.293%
2001	0.290%
1010	0.285%

Table 1: Frequencies for the 20 most common PINs.

4.2.3 Indecipherable Sound

If the owner of the Amazon Echo is nearby when someone tries to gain unauthorized access by doing something akin to yelling through a window, the owner will notice that someone is trying to use their Echo. One attack vector we explored was the ability to create a sound wave that is recognized by the Echo as the word "Alexa" but just sounds like noise to a human. With the rise of various types of classifiers, the art of creating bogus input that gets incorrectly classified has become more popular. When one knows the function being used to classify either a sound wave or an image, one can usually come up with an input that specifically exploits the mistakes of that function.

For the sake of our problem the function that is used to classify the word "Alexa" is a black box. It is possible that we would be able to discover the code of the recognition function by digging through the Echo source code or by examining the hardware. However, for all words other than the four possible wake words, the Echo sends a sound file to Amazon's servers and Amazon replies to the Echo with the words that were said in the sound file. This is likely done to protect Amazon's proprietary voice recognition software. Regardless of the reason, this means that we can only guess what Amazon is doing in order to recognize words in sentences.

Our methodology for creating an input that was recognized by the Echo but not by us was to take a recording of the word Alexa and apply various distortions to it until we had a working input. We used the program Praat, an open source python tool for analyzing and manipulating sound waves related to human speech. While we don't know for sure how Amazon is classifying words, we can guess that they are looking for clues such as the distinct vocal cues that humans use to recognize words. They are also likely filtering out frequencies outside of the human hearing range. By keeping the distinctive vocal features, but distorting everything else, we can create a sound wave that does not sound like someone saying Alexa, but is still recognized by the Echo.

We did not obtain the result we had wanted from our methods. We were able to obtain some distorted versions of the word Alexa that the Echo recognized, but an attentively listening person could clearly tell that the word Alexa was being said. We believe that further improvements could be made to our distortion techniques that would give us a better audio sample that could be used to secretly give a command to the Echo. The audio sample we did obtain, however, is fairly distorted and it is possible that someone not paying attention would have difficulty noticing that the word Alexa was being said.

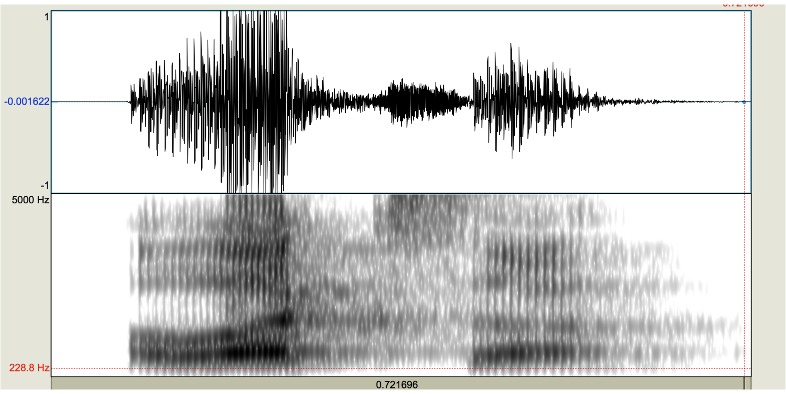

Figure 2: Spectrogram of the word Alexa

4.3 Network Attacks

We attempted a number of network attacks on the Echo device and Echo iOS application. The methodology and results of these attacks are described in the sections below.

4.3.1 Man in the Middle

Wireshark[1] is an application that can intercept and record network traffic. The Echo device transmits data via a wireless network. We connected both the Echo device and a computer running Wireshark to the same 802.11 wireless local area network, and set the Wireshark application to monitor 802.11 packets.

The MAC address of the Echo device is easily retrievable via the Alexa application, and we filtered the wireless traffic for packets with either a source or destination address that matched the Echo's MAC address. We were able to capture a large number of packets sent between the Echo device and a number of other addresses.

Notably, we saw most of the packets being transferred when we interacted with the Echo device; however, there were still packets being sent when we were not interacting with the device. It is entirely possible that these interim transmissions are state updates or firmware updates, though we believe this warrants further investigation.

4.3.2 Recovering Information From Wireless Packets

If listeners on the network can recover meaningful information by monitoring wireless packets, this may compromise the integrity of the security policy described above. In the best case, listeners can determine how the Echo is being used. In the worst case, listeners may be able to recover personal details, including payment information, from the transmitted packets.

We supplied Wireshark with the encryption key used for the wireless network, so were able to decrypt the 802.11 packets to recover header information. However, we were not able to decrypt the data portion of these packets. This made it difficult to recover any meaningful information from these packets. An example of the information revealed by the packet is given in Appendix C.

We did attempt to recover information from packets by looking at repeated requests and responses. Namely, we looked for patterns in the number and size of packets transferred for different types of interactions with the Echo device, with the idea that other users of a network may be

[1] https://www.wireshark.org

able to determine how the Echo device was being used. We were unable to predict the type of interaction reliably with any of the techniques that we tried.

4.3.3 Replaying Packets

We wished to explore the possibility of replay attacks on the Echo device. Replay attacks are a form of network attacks that capture valid network transmissions and then attempt to maliciously replay them. If the Echo device was not secure against these types of attacks, an adversary could, for example, monitor traffic for Echo transmissions that involve purchasing items, and then replay these transmissions multiple times, causing the user to order many more of the item than they intended.

Fiddler[2] is an application capable of capturing, modifying, and sending network traffic. We used this application to replay the wireless transmissions captured by Wireshark for a number of interactions with the Echo device including checking the weather and ordering an item. We saw transmissions to the Echo device in response to these replayed transmissions, though due to the encryption on the response packets, we were unable to gain any information from them. We did not see the expected response from Alexa (announcing the weather or purchasing more items, respectively, for the examples given above). This is a good indication that Amazon has taken measures to protect the Echo device against the aforementioned replay attacks.

4.4 API-based attacks

Digging up the Echo web application reveals an unofficial API. Unfortunately, the file isn't particularly well structured, and it is difficult to infer some of the API endpoints. That being said, we were able to make some simple attempts such as trying to change the wake word to an option outside of the predefined list ("Alexa", "Amazon", "Echo", "Computer"). In fact this is how it was first discovered that "Computer" was a valid wake word, before Amazon publicly added the feature. We tried words such as "Kindle" or variations of "Alexa", but the server always recognized an invalid keyword.

As explained earlier, rooting the Echo through a proxy can help record the traffic coming in and out of the device. While most requests are fully encrypted, updates to the device come in a binary format from a regular HTTP request. Attempts to reverse engineer the binary have been done, but most of the code is not relevant to Amazon. In fact, Amazon also publishes the source code of the device, but the intercepted update contains more information. In particular, it is possible to see where the voice recognition models are stored. With better understanding of the recognition procedure, it may be possible to change these models. Thus, should an attacker find a way to trigger an update in the device, modifying update binaries may be an interesting strategy.

5 Recommendations

In this section we analyze how Amazon has decided to deal with the attacks we have attempted. We determine whether their measures for protection are sufficient and provide recommendations in areas that might be improved.

5.1 Amazon Orders and the 4-Digit PIN

We have shown that it is possible for an adversary to perform a brute-force attack on the 4-digit PIN used for protecting unauthorized orders. Furthermore, if the user does not have a randomly chosen PIN, this could significantly reduce the time needed to perform a successful attack. An adversary does not even have to be in the same room as the Echo to complete this attack. He could use voicemail or play a recording through a window.

Amazon takes several precautions to prevent these types of attacks. First, Alexa limits users to trying only two PINs at a time before telling them to check their settings in the Alexa app. Second, Alexa will not reveal or let users change their shipping addresses, so even if an order is placed successfully, the adversary would still have to intercept the package at the owner's address. Amazon also records users' actions in multiple places, which is helpful in notifying users in various ways. All voice commands understood by Alexa appear in the user's Alexa app, and a user receives

[2] http://www.telerik.com/fiddler

an email notification for each order placed. Thus, users are able to detect suspicious activity. However, it could be argued that users may not frequently check their Alexa app. Furthermore, Alexa does not prohibit placing the same order multiple times, so a user may have difficulty canceling many orders in a timely fashion.

Although Amazon has made some design decisions to prohibit unauthorized users from placing orders, we make several recommendations in order to achieve aspects of our ideal security policy. The best solution to these problems, ideally, would be for the Amazon Echo to have voice recognition. That way it could distinguish between primary, secondary, and other users, and users would not have to take extra precautions to protect their accounts. However, with limited technology we can still make improvements. The current default settings for the Echo include allowing voice purchasing without having to enter a 4-digit PIN. These defaults show a direct tradeoff between convenience and security. Amazon makes it easier for users to place orders and thereby earn Amazon revenue while compromising the integrity of its users' accounts. We believe the defaults should be to prohibit voice purchasing and that the user can only enable it with having to enter a 4-digit PIN. Furthermore, the Echo should stop accepting users' requests for orders after a certain number of failed attempts at entering the PIN. The exact number of incorrect PINs it should allow might be a complex decision, as the Echo's margin of error for speech processing paired with users' poor choices for PINs vary the numbers. The user should have to confirm an order on the Alexa app if too many PINs have been tried in order to re-enable voice purchasing.

In order to combat this ambiguity with the 4-digit PIN, we recommend Amazon switch this security feature to a few-word passphrase. The entropy gained by using just three words compared to four numbers is orders of magnitude greater. Additionally, passphrases are much easier for users to remember and randomly generate. The current speech processing technology the Echo uses should also be able to accommodate this change.

Lastly, users themselves can take one simple step to combat the more severe of these problems. By turning off the microphone every time they leave their residences, users can prevent those who cannot break into their homes from exploiting their Echos through sound attacks.

6 Conclusion

With the rise of connected devices and the Internet of Things, online security and privacy have become a daily concern. Personal assistants such as the Amazon Echo can now be fully integrated with a user's credit card, phone number, or other home devices. Compromising the Amazon Echo could therefore have dramatic consequences over its users. In this paper, we discussed the basics of the Echo device and applications, and provided an ideal security policy for the device. We then attempted to challenge this policy through a series of attacks including network, sound, and API based attacks. One of the main strengths of the Amazon Echo is that most of the logic happens behind the scenes, on the Amazon cloud servers. In fact the device only handles voice recognition, recording, playing, and some basic configurations. This makes it more difficult to target the Echo.

From our analysis, we found the Amazon Echo to provide satisfiable security. In particular we found the device to be resistant to network and API based attacks. We believe that voice based attacks are promising but require better understanding of the inner working of the voice recognition system. Another potential area for future would be to target the Echo through third party "skills", which can be used to augment the device and may offer their own set of vulnerabilities.

References

[1] Amazon. *Amazon Echo*, May 2017. https://www.amazon.com/dp/B00X4WHP5E.

[2] Amazon. *Set Up Your Echo*, January 2016. https://www.amazon.com/gp/help/customer/display.html?nodeId=201601770.

[3] Consumer Intelligence Research Partners. *Amazon Echo – What We Know Now*, 2017.

[4] Hidden voice commands, Carlini et. al, 2016

[5] Marc Padilla's blog. URL: https://blog.padil.la/category/amazon/

[6] Anonymous blog. URL: https://medium.com/@micaksica

[7] Lifehacker blog. URL: http://lifehacker.com/5944567/the-most-and-least-common-pin-numbers-and-numeric-passwords-is-yours-one-of-them

[8] Amazon Echo teardown gets inside the smart speaker powered by the cloud, Bill Detwiler, 2015. URL: https://www.cnet.com/news/amazon-echo-teardown-a-smart-speaker-powered-by-amazons-cloud/

[9] Two Amazon Teardowns Are Better than One, Kay Kay Clapp. URL: http://ifixit.org/blog/8066/amazon-teardown-echo-dot/

A Audio Distortion Code

```
import math

# Taken from the Sound file we were using, change as needed.
TEMPLATE = ’’’File type = "ooTextFile"
Object class = "Sound 2"

xmin = 0
xmax = 2.136235827664399
nx = 94208
dx = 2.2675736961451248e-05
x1 = 1.1337868480725624e-05
ymin = 1
ymax = 1
ny = 1
dy = 1
y1 = 1
z [] []:
    z [1]:
’’’

def read_z_array(fname):
    with open(fname) as f:
        content = f.readlines()
    content = [x.strip() for x in content]
    content = map(lambda x: float(x[x.index(’=’) + 2:]), filter(lambda x: x[:6] == ’z [1] ’, conte
    return content


def remove_every_n(z_array, n):
    new_array = []
    for i in range(0, len(z_array), n):
        new_array.append(z_array[i])
    return new_array


def add_noise(z_array, freq, amp):
    for i in range(len(z_array)):
        z_array[i] += amp * math.sin(2.0 * math.pi * 1.0/freq * i)
    return z_array

def print_z_array(z_array):
    print(TEMPLATE)
    for i in range(1, len(z_array) + 1):
        #         z [1] [1] = 0
        print(’        z [1] [{}] = {}’.format(i, z_array[i-1]))


def swap_size_n_groups(z_array, n):
    new_array = []
    for i in range(0, len(z_array) - (n * 2), n * 2):
        new_array += z_array[i+n: i+2*n] + z_array[i:i+n]
    while len(new_array) < len(z_array):
        new_array += [0.0]
    return new_array
```

```
# Change Alexa.Sound to the sound file you want to use from Praat.
original = read_z_array('Alexa.Sound')
distorted = remove_every_n(original, 2)
print_z_array(distorted)
```

B Script For Brute-Force PIN Attack

```
import os
import time

first_attempt = True

for i in range(10000):
  if first_attempt:
    os.system("say 'Alexa... buy an Amazon Echo Dot'")
    time.sleep(10)
    os.system("say 'yes'")
    os.system("say 'please'")
    time.sleep(5)
    os.system("say '" + str(i).zfill(4) + "'")
    time.sleep(5)
    first_attempt = False
  else:
    os.system("say '" + str(i).zfill(4) + "'")
    time.sleep(5)
    first_attempt = True
```

C Example 802.11 Packet Information

```
Frame 5656: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0
    Interface id: 0 (en0)
    Encapsulation type: IEEE 802.11 plus radiotap radio header (23)
    Arrival Time: Apr 27, 2017 02:34:11.096899000 EDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1493274851.096899000 seconds
    [Time delta from previous captured frame: 0.000138000 seconds]
    [Time delta from previous displayed frame: 0.000138000 seconds]
    [Time since reference or first frame: 14.160336000 seconds]
    Frame Number: 5656
    Frame Length: 142 bytes (1136 bits)
    Capture Length: 142 bytes (1136 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan_radio:wlan:data]
Radiotap Header v0, Length 48
    Header revision: 0
    Header pad: 0
    Header length: 48
    Present flags
    MAC timestamp: 57775764
    Flags: 0x14
    Channel frequency: 2437 [BG 6]
    Channel flags: 0x0480, 2 GHz spectrum, Dynamic CCK-OFDM
    SSI Signal: -39 dBm
    SSI Noise: -96 dBm
    Antenna: 0
    Channel number: 6
    Channel frequency: 2437
    Channel flags: 0x00010480, 2 GHz spectrum, Dynamic CCK-OFDM, HT Channel (20MHz Channel Width)
    MCS information
    [Data Rate: 65.0 Mb/s]
    A-MPDU status
802.11 radio information
    PHY type: 802.11n (7)
    MCS index: 7
    Bandwidth: 20 MHz (0)
    Short GI: False
    Greenfield: False
    FEC: BEC (0)
    Data rate: 65.0 Mb/s
    Channel: 6
    Frequency: 2437 MHz
    Signal strength (dBm): -39 dBm
    Noise level (dBm): -96 dBm
    TSF timestamp: 57775764
    Last part of an A-MPDU: False
    A-MPDU delimiter CRC error: False
    A-MPDU aggregate ID: 324
    [Duration: 48 us]
IEEE 802.11 QoS Data, Flags: .p.....TC
    Type/Subtype: QoS Data (0x0028)
    Frame Control Field: 0x8841
    .000 0000 0011 0000 = Duration: 48 microseconds
    Receiver address: be:9f:ef:d3:18:12 (be:9f:ef:d3:18:12)
    Destination address: be:9f:ef:3d:93:64 (be:9f:ef:3d:93:64)
    Transmitter address: AmazonTe_be:48:01 (40:b4:cd:be:48:01)
    Source address: AmazonTe_be:48:01 (40:b4:cd:be:48:01)
    BSS Id: be:9f:ef:d3:18:12 (be:9f:ef:d3:18:12)
    STA address: AmazonTe_be:48:01 (40:b4:cd:be:48:01)
    .... .... .... 0000 = Fragment number: 0
    0001 0100 0100 .... = Sequence number: 324
    Frame check sequence: 0xf4a832a2 [correct]
    [FCS Status: Good]
    Qos Control: 0x0000
    CCMP parameters
        CCMP Ext. Initialization Vector: 0x000000000149
        Key Index: 0
Data (56 bytes)
    Data: 2bc4775634b7c861a2ac977e0c70569cc702851749f55a51...
    [Length: 56]
```