Measuring How Much IoT Devices Upload via Traffic Analysis

Enze Liu, TJ Smith, Zesen Zhang
University of California, San Diego
{e7liu,tjs003,zez003}@eng.ucsd.edu

ABSTRACT

We examine the problem of how much voice data is being streamed to the server by Echo. Echo is Amazon's smart speaker that has an always-on microphone. Conceptually, it should start recording after hearing a certain wake word (Alexa, Echo, etc.) and then send the recorded audio to its server for further processing. Previous work has largely focused on stopping the adversaries from inferring the encrypted user data being transmitted. To our best knowledge, little has been done to verify the fundamental assumption that Echo should only start to record and trasmit data after hearing the wake word. **[[what else is not done by the others]]**?

In this work, we performed measurements that shed lights on these mysteries. More specifically, we provided answers to the following questions: 1. Is any conversation before the wake word being streamed to the server? 2. **[[what else did we measure in this work?]]**. To achieve this, we prerecorded a piece of audio that contained an Echo command ("Alexa, where is New York City?") as well as irrelevant conversations prefixing and suffixing the Echo command. Different segments of this audio were played over and over again in a controlled environment, and all packets transmitted by Echo were recorded. After data cleaning and carefully analyzing the traffic data, we are able to reveal **[[X]]** key findings. First, we confirmed that normally Echo would not stream any conversation that happened before the wake word. That said, we did observe that, sporadically, Echo would transmit a non-trivial amount of data during an ongoing conversation that did not contain the wake word. **[[What else did we find]]**.

1 INTRODUCTION

Physical devices connected to the internet, also known as Internet-of-things (IoT), have gained popularity in recent years. Among numerous types of IoT devices, smart speakers with voice assistants, such as Amazon Echo and Google Home, are widely seen in users' home. These devices detect and respond to voice commands. They normally have always-on microphones, which theoretically start recording after hearing a wake word and then send the voice data to a server via internet for further processing [1].

This behavior has sparked many privacy concerns, including but not limited to what is being recorded, how the collected data is used and stored, and whether it is being protected well [4–6, 8, 9]. Much of previous work has been centered around protecting sensitive information from being leaked to adversaries [4–6]. However, little work has been done to confirm the rudimentary assumption – that Echo should only start recording after hearing the wake word. Though Amazon has made it possible for users to view, play and delete the voice data transmitted to its server [7], we are not fully convinced that the voice data made available is the only data transmitted by Echo everytime. More specifically, we want to know if Echo is streaming any conversation that happened before the wake word. **[[and what?]]**

To answer all these questions, we opted for a passive mearsurement approach — analyzing network traffic sent by Echo. We used wireshark to capture all the packets transmitted by Echo. An audio that could trigger Echo to actively send data was also needed. We prerecorded an audio file that contained an Echo command ("Alexa, where is New York City?") that was about 2 seconds long. In this audio file, we also prefixed and suffixed the Echo command with irrelevant conversations. This prerecorded audio file gave us the ability to perform controlled experiments and verify our assumptions. Different segments of this audio were played to Echo over and over again, and each time we would compute the total package size transmitted. Having done this, for each unique segment of the audio, we now had a set of total packet sizes that Echo transmitted when hearing this segment of audio. Then, statistical analysis was to performed to test our assumptions.

Our first contribution: we confirmed that most of the times Echo only started to record after hearing the wake word. To test whether Echo would record the conversation before the wake word or not, we crafted two input audio segments for Echo. The first audio segment (denoted as I_1) was about 2.5 seconds long, and contained only the Echo command "Alexa, where is New York City?". The other audio segment (denoted as I_2) was about 11.5 seconds long, with a 9 seconds irrelevant conversation attached before the Echo command. We played both I_1 and I_2 alternatively overnight in a setting with noise level X**[[do we need noise?]]**. For each segment, we played it for over 100 times, giving as over 100 sample timeframes, within each one could easily compute the total size of packets transmitted. We manually inspected all the timeframes, removing obvious outliers. We then performed **[[I did t-test]]** on these datasets, getting a p-value of **[[X]]**. Given that our null hypothesis is they are not the same, we come to the conclusion that Echo only started to record after hearing the wake word.

Our Second contribution: we confirmed that xxx. [[Other experiments we did.]]

2 BACKGROUND

My background is here. **[[add background]]**

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 EXPERIMENT

We ran some experiments on Alexa and using wireshark to see whether Alexa recorded consumer's voice more than we supposed they do. We then compared the size of package of each experiment to give out our results.

3.1 Environment Set Up

In order to catch the package that was sent out by Alexa to Amazon cloud, we use set up a hotspot from our server and let Alexa connected to it. In the meanwhile, we play a voice and record the time when Alexa replied our command. Further, we use wireshark to record all the package that going through the hotspot which is connected by Alexa.

We did the experiment overnight in the office, to make sure there were no voice generated by person and all the thing Alexa might record was the voice played by our code.

3.2 Baseline Experiment

In the baseline experiment, we kept Alexa in the office overnight without doing anything and catch all the package that sent out by Alexa.

In this part, we assume that Alexa will not transfer any voice message to the cloud as there were no voice except some noise generated by nature. Therefore, we could use the package that were caught up in this experiment to be the baseline, who were the "necessary" package that Alexa communicate with cloud server. And we assume them to be unsuspicious package, which including some handshake package and broadcasting package to let the Alexa confirm which server to use.

3.3 Music Experiment

In the music experiment part, we played a music named Home overnight and catch up the data transmitted by Alexa.

"Home" is a music without any voice from human but it did contain some sound that could be recorded by Alexa. As Alexa has mentioned in its document[1]: "By default, Echo devices are designed to detect only your chosen wake word (Alexa, Amazon, Computer or Echo)", it would not record and transmit anything to cloud once you did not call them. We would assume that Alexa would not transmit any voice package to cloud server as there is no human voice in the music, nothing would be recognized by the echo as "Alexa". As a result, if Alexa did the thing in the way as they announced, the package transmission result would be the same as it did in the Baseline Experiment.

3.4 "Prefix" Experiment

In this part, we did two different kinds of experiments. First, we played a voice with a 9 seconds prefix sentence and then followed by a 2.5 seconds command to wake Alexa up and let it transmit the package and answer. The prefix sentence is a random sentence that simulates a person talking in his daily life, and then he asked Alexa a question. The whole prefix sentence is: " I think I am having some trouble hearing what people trying to say to me. I am not feeling particularly well now. I am also thinking switching a new job." And the command is: "Alexa, where is New York City." It is really common in our daily life that people just having some conversation and suddenly find something to deal with, therefore they send a command to Alexa.

Here, we let Alexa to start transmitting the voice to the server. As Amazon has mentioned in his document [1]: "By default, Alexa-enabled devices only stream audio to the cloud if the wake word is detected (or Alexa is activated by pressing a button)." it will not transmit any data before the wake up word. Our assumption is that, if Alexa truly did the transmission as it mention, the size of each transmission package should be the same, as there is only a short command at the end of the sentence. However, if the optimization algorithm of Alexa is not efficient enough, it might transmit some parts of the prefix sentence to the cloud and the package size would wave whole night.

In order to make our results more significant, we did another comparision experiment. We played a voice with full prefix sentence mentioned above and a voice with no prefix sentence but just command for Alexa. And if Alexa did try some part of the prefix sentence, the package size of these two parts would show up a significant difference.

3.5 "Postfix" Experiment

Moreover, we want to measure whether Alexa will keep transmitting package even if Alexa has noticed the whole command and started replying. The assumption here is that, as Alexa has started replying, it means Alexa has fully recognized the whole command and it should not transmit the rest of the sentence onto the server. It fits our real life situation that someone sends a command to Alexa and starts another conversation with others immediately. If Alexa transmits the rest of the conversation, it would definitely rises the privacy concern.

Here, we use "Alexa, where is New York City" as a command sentence and adds a 1 second postfix sentence. We gave an approximately 0.5 seconds gap between the command and postfix sentence to make sure that Alexa has recognized the whole command and started to reply. Then we started playing three different kinds of sentences which are: a) command voice b) command voice with 0.5 second postfix sentence and c) command voice with 1 second postfix sentence. And analyze the package sent out by Alexa.

What's more, we want to see whether that 0.5 second pause is necessary for Alexa to realize the command is over and play the command with postfix sentence immediately. We try to find out whether Alexa would automatically cut off the voice once it recognized the following sentence is meaningless. We played command with 0,1,2,3,4,5,6 seconds postfix sentence, which contains nonsense sentences, to see whether Alexa would cut off the transmission itself.

Further, we try to figure out what is the long enough pause for Alexa to cut off the voice transmission and recognize the whole sentence. We played the command with different pause before playing the whole postfix sentence, to figure out when Alexa would reply a right answer to our command. Here we tried the voice gap between command and postfix sentence with 0.2,0.3,0.4,0.5 seconds.

4 MEASURING TRAFFIC STREAMED TO SERVER

This measures the traffic streamed to server.

5 CONCLUSION

My conclusion is here.

6 RELATED WORK

Amazon Echo is a smart speaker developed by Amazon that is connected to the internet [3]. It has an always-on microphone that starts recording automatically after hearing certain wake word (Alexa, Echo, etc.). Voice commands following the wake word are streamed to Amazon's cloud-based intelligent command handling program, known as Alexa, for further processing. Alexa will then try to respond to users' commands. Amazon also keeps copies of the voice commands and responses, together known as response cards [7]. These response cards have been made available to users by Amazon so that they can be viewed, played, and deleted [2].

REFERENCES

[1] Amazon Echo (2nd generation) — Alexa Speaker. https://www.amazon.com/Generation-improved-sound-powered-design/dp/B06XCM9LJ4?. (????). (Accessed on 10/15/2019).

[2] 2010. GP. (2010). https://www.amazon.com/gp/help/customer/display.html?nodeId=201602040

[3] 2019. Amazon Echo. (Oct 2019). https://en.wikipedia.org/wiki/Amazon_Echo

[4] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. 2019. Keeping the smart home private with smart (er) iot traffic shaping. *Proceedings on Privacy Enhancing Technologies* 2019, 3 (2019), 128–148.

[5] Noah Apthorpe, Dillon Reisman, and Nick Feamster. 2017. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805* (2017).

[6] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. 2017. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *arXiv preprint arXiv:1708.05044* (2017).

[7] Marcia Ford and William Palmer. 2019. Alexa, are you listening to me? An analysis of Alexa voice service network traffic. *Personal and Ubiquitous Computing* 23, 1 (2019), 67–79.

[8] Geoffrey Fowler. 2019. Perspective | Alexa has been eavesdropping on you this whole time. (May 2019). https://www.washingtonpost.com/technology/2019/05/06/alexa-has-been-eavesdropping-you-this-whole-time/

[9] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. 2018. Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 102.