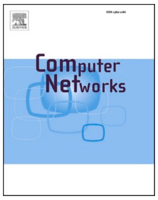
Computer Networks 205 (2022) 108782

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Data Article

Amazon Alexa traffic traces

Rubén Barceló-Armada, Ismael Castell-Uroz [*], Pere Barlet-Ros

Universitat Politècnica de Catalunya

ARTICLE INFO

Keywords:
Amazon Alexa
IoT
Network traffic traces
PCAP

ABSTRACT

The number of devices that make up the Internet of Things (IoT) has been increasing every year, including smart speakers such as Amazon Echo devices. These devices have become very popular around the world where users with a smart speaker are estimated to be about 83 million in 2020. However, there has also been great concern about how they can affect the privacy and security of their users [1]. Responding to voice commands requires devices to continuously listen for the corresponding wake word, with the privacy implications that this entails. Additionally, the interactions that users may have with the virtual assistant can reveal private information about the user. In this document we publicly share two datasets that can help conduct privacy and security studies from the Amazon Echo Dot smart speaker. The included data contains 300.000 raw PCAP traces containing all the communications between the device and Amazon servers from 100 different voice commands on two different languages. The data can be used to train machine learning algorithms in order to find patterns that can characterize both, the voice commands and people using the device as well as Alexa as the device generating the traffic.

Value of the data

•*Why is this data useful?*

Big technology companies like Amazon, Google, Apple or Microsoft invert massive amounts of money and resources to develop the so-called smart speakers, intelligent devices that execute voice commands to perform many of the tasks and routines of their users. However, this kind of systems present many privacy and security concerns. For instance, in order to perform those actions, smart speakers must constantly listen to detect the wakeup words needed to execute voice commands. Letting an external device to constantly listen to the conversations of your home poses a threat to the privacy of its inhabitants. Moreover, intelligent devices rely on machine learning algorithms used to interpret the natural language and infer its meaning. To achieve the massive amount of data needed to train their algorithms in order to correctly decipher voice commands from many countries, languages and different accents, companies purposely collect and use information of their own users, often in a completely transparent way. On top of that, some users connect their smart speakers to other intelligent devices (e.g. lights or door locks), fact that can be seen as a security threat in case an attacker could find a flaw on online systems used by smart speakers (e.g. [2]).

•*Who can benefit from this data?*

In this work we present a collection of network traffic traces with all the information between one smart speaker (Amazon Echo Dot) and their servers. The dataset can help research groups that want to carry out studies based on the behavior of traffic generated by a smart speaker such as privacy studies, task scheduling or payload prediction. For instance, it can be used as input for Deep Learning algorithms in order to find non-obvious patterns that could represent some kind of threat to their users. An attacker could be interested in knowing the amount, genre, and habits of the people that lives inside a house to select places where there is only one person living inside or to infer time frames with higher possibilities of being empty. Thus, previously studying these characteristics can help to prevent those attacks.

•*How can this data be used for further insights and development of experiments?*

As the dataset contains raw PCAP files containing all the information transmitted over the network, researchers can use these datasets to extract the features that they consider appropriate for their algorithms or studies. For instance, some network experiments may only need the data inside the TCP headers, while other studies like network

[*] Corresponding author.
E-mail address: ismael.castell@upc.edu (I. Castell-Uroz).

https://doi.org/10.1016/j.comnet.2022.108782
Received 7 September 2021; Received in revised form 10 December 2021; Accepted 6 January 2022
Available online 8 January 2022

monitoring could profit from the data included in the payload to feed DPI algorithms to discover Alexa traffic characteristics hidden that can lead to an automatic prediction/detection/classification tool.

•*What is the additional value of this data?*

We also include an explanation of the data acquisition methods to help other teams to complement it with other data if needed. For instance, other languages or a determined list of commands can improve the dataset for specific cases. Moreover, it also allows to replicate our results and compare them to find evolution changes between different time frames. Note, that all the collection process is very time-consuming, though. It took almost three months to develop, refine and collect all the measures present in the dataset. Hopefully this dataset can speed up investigations that require this kind of data.

Data

The dataset includes the network traffic communications between the Amazon Echo Dot device and Alexa servers. It does not contain only the communications with the Alexa Voice Servers in charge of the voice command responses, but also the traffic from all the other Amazon services running in the background. Thus, traffic traces include information about the entire Alexa ecosystem. We used Wireshark to inspect a subset of the collected traces and found a total of 5 different services running inside. The next table presents the 5 servers as well as their roles.

Server	**Service**
ntp-g7g.amazon.com	Clock synchronization services with NTP protocol
avs-alexa-14-eu.amazon.com	Alexa Voice Services (voice command interpretation)
s3–1-w.amazonaws.com	Check Internet link status
device-metrics-us-2.amazon.com	Metrics directly collected by Amazon from the device
api.amazonalexa.com	Skill services

To generate the dataset, we executed and collected traces from 100 different voice commands. All the experiments have been repeated in two different languages, English and Spanish. For each of the languages each command has been executed with 3 different voices (synthetic voices available on the Amazon Polly service). Moreover, we collected 500 different samples of each voice command and synthetic voice combination to be able to discover points of interest or possible outliers. In total, the dataset contains 300.000 different network traces.

The dataset is divided in two different files, one for each of the languages used to collect the measures. Each file follows the structure presented at Fig. 1, with two folder levels and traces inside the second level. The first level denotes the synthetic voice used to execute the command. The second level indicates the command performed. All the samples inside the command folder contain raw PCAP files including the entire network communication between the Amazon Echo Dot device and Amazon servers. Each PCAP file consists of a list of all the network packets going through the network interface while the capture is active. For every packet both, TCP headers as well as the packet payload are captured. It also contains information about the timing of each communication. Note that most communications include network security and, consequently, many packets are encoded using TLS, limiting the amount of information from the payload accessible by the user.

In summary each file contains:

1. *DatasetEnglish.tar.gz*

- 500 measures for each one of 100 English voice commands, in 3 different voices (1 male and 2 female).
- 150,000 raw PCAP network traces.
- 12.6GB of network traffic data generated between the Amazon smart speaker and Alexa Voice Service servers.

2. *DatasetSpanish.tar.gz*

- 500 measures for each one of 100 Spanish voice commands, in 3 different voices (1 male and 2 female).
- 150,000 raw PCAP network traces.
- 15.6GB of network traffic data generated between the Amazon smart speaker and Alexa Voice Service servers.

Experimental design, materials, and methods

When a user communicates with the virtual assistant Alexa through a smart speaker or a smartphone, they interact with different components of the Amazon Alexa system. The first component is the smart speaker itself, that will react to command voices executed by the user. The smart speaker has limited computing power and relies on Alexa Voice Service servers (AVS) to decipher the natural language and interpret the given command. Once AVS knows the action to be performed it calls the required skill (Alexa applications) through the so-called AWS Lambda servers, specialized for the task. When the skill or task has finished, Lambda servers notify AVS server and the later answers back to the smart speaker, sending the command reply in text format. Lastly, the smart speaker uses a text-to-speech system to read the response as audio. Fig. 2 shows a summary of the process. All these components compose the Amazon Alexa ecosystem and ensure a proper response to voice

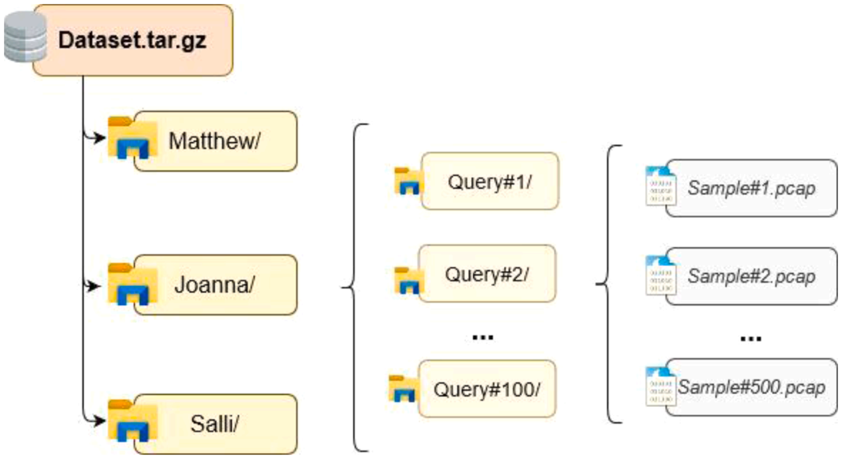


Fig. 1. Dataset file structure.

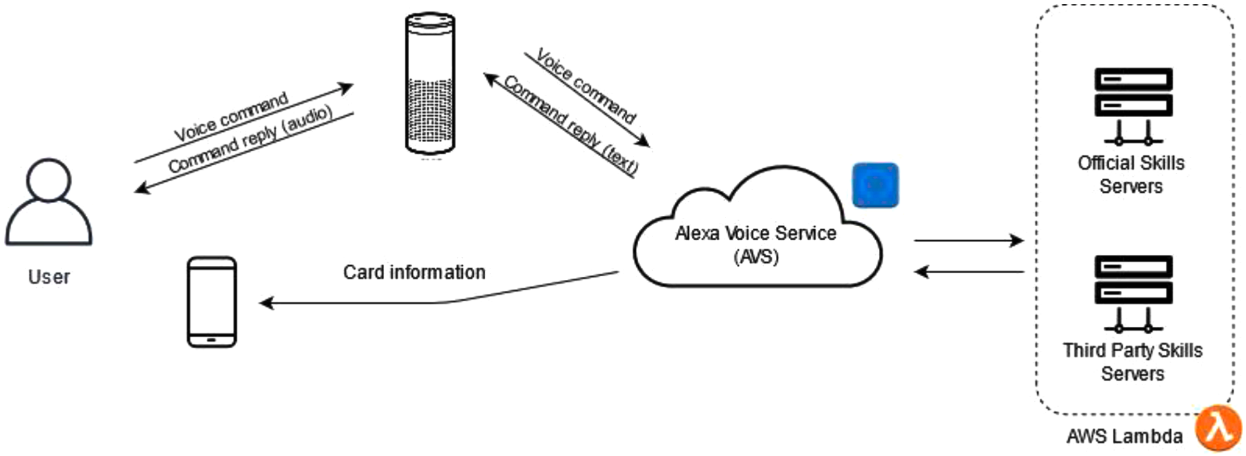


Fig. 2. Amazon ecosystem.

commands issued by the user.

Our main objective is to collect the communications between the smart speaker and the AVS server. In order to obtain all the network traces, we perform a kind of a "man-in-the-middle" attack, forcing all the communications to pass through a hot spot owned by us. In our laboratory experiment the hot spot consist of a *Raspberry Pi* configured as a WIFI hot spot. The Amazon Echo Dot device is connected to the WIFI generated by the *Raspberry Pi*, and the communications arrive to the AVS servers through the *Raspberry Pi* LAN port. Fig. 3 shows the structure of our methodology.

The data collection process was developed on Python [3], and consists of six differentiated steps:

1 The Raspberry Pi starts the trace capture process by means of the TCPDump tool and generates a voice command. To this end it takes one voice command at a time from a file containing a list of the selected commands. The command will be read out loud using the cloud text-to-speech service Amazon Polly (offered free of charge by Amazon Web Services) through the speakers attached to the Raspberry Pi.
2 Alexa Echo Dot wakes up, captures the voice command and sends it to AVS services.
3 TCPDump running inside the Raspberry Pi intercepts the request and saves it inside the network trace.
4 Amazon AVS sends back the response that once more gets intercepted by our TCPDump probe.
5 The response arrives to Echo device where it is converted to an audio message by their own text-to-speech system.
6 Before launching the next voice command, the Raspberry Pi stops the TCPDump capture and saves it as a sample file inside the corresponding folder.

To avoid noise and external influences within the measures the only device connected to the WIFI hotspot are the smart speakers. Moreover, Raspberry Pi OS automatic updates and unneeded services were disabled in order to eliminate non-expected network traffic. On top of that, we also have to insulate as much as possible the device from sounds and external voices that can make it to wake up at unexpected intervals. To this end we setup the device in a completely closed environment with the only presence of the device, the Raspberry Pi and one of the members of the research group to check that everything is correct.

Regarding the voice commands, to inspect different scenarios we tried to select not only the commands most used by real users, but also some more complex commands not so popular. To this end, we took profit of Alexa's own recommendation system. For registered users Alexa periodically sends by email some of the most popular as well as some new and useful commands. We manually selected a subset of them to populate our dataset. We purposely discarded commands setting up alarms, timers and other commands saving data inside memory as the execution of hundreds of them made the device to hang.

Unfortunately, not all the voice commands were correctly detected at the first run using synthetic voices. Thus, we manually checked and tweaked every text-to-speech command to avoid generic error responses. Nevertheless, Alexa Echo Dot device occasionally fails to detect some of the commands. Consequently, we took a total of 600 measures for each voice command, and automatically discarded traces without content or with very small amount of traffic, indicative of those errors. Then, we randomly selected 500 of the remaining ones to create our dataset.

Experimental study based on the dataset

The collected dataset was used for a thesis dissertation [4] presented on July 2021. The objective of the study was to find if it was possible to identify different smart speakers' users looking only at the encoded network traffic. To this end we used a Deep Learning algorithm and based our study in a closed-world assumption. In other words, our resultant system is able to identify exclusively the people used to train

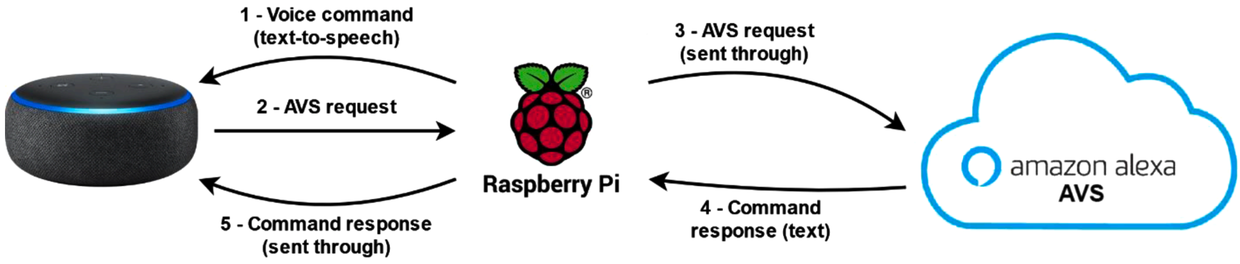


Fig. 3. Trace collection structure.

the system, and is not able to generalize to different scenarios. However, it is a first approximation to study the feasibility of those systems to find identifying patterns between encrypted data.

To train the system we collected the IP addresses of the AVS servers present in the dataset and extracted all the traces corresponding to any of them. The resultant subset was passed through *TCPTrace*, a tool to extract interesting information from a TCP trace (e.g. total packets, ack packets, average segment size). We also computed other composed information such as the average size, average time between packets or the total sum of bytes transmitted. Then, we used 70% of the data for the training phase of an MLP algorithm, a 10% for the validation phase and the remaining 20% for the evaluation. Our results, present in Fig. 4, show more than 80% of accuracy predicting the user that executed the voice command. Although not definitive, it is a first step to demonstrate that this type of information may be extracted just capturing encrypted network data.

Specifications table

Subject	**Computer Science:** *Computer Networks and Communications*
Specific subject area	Data transferred between Amazon smart speakers and Amazon Alexa Servers.
Type of data	Network traffic data
How data were acquired	***Hardware:*** Amazon Echo Dot G3, Raspberry Pi ***Software:*** Tailor-made software written in Python to recollect network traffic data
Data format	Raw
Parameters for data collection	- 100 different commands have been used. - Selected commands include the most common ones as well as Alexa's own suggestions. - Commands are reproduced by three different voices (text to speech voices from Amazon Polly). - Each command is executed 500 times for each voice. - Dataset is taken twice, in two different languages.
Description of data collection	The data has been collected by means of a custom Python too developed to automate the task. The tool allows you to execute voice commands automatically while the traffic flow is being captured.
Data source location	**Institution:** Universitat Politècnica de Catalunya **City**: Barcelona **Country:** Spain
Data accessibility	https://cba.upc.edu/phocadownload/public_documents/ids_datasets/dataset_english.zip https://cba.upc.edu/phocadownload/public_documents/ids_datasets/dataset_spanish.zip
Related research article	Ruben Barcelo Armada, Pere Barlet Ros, "Disección y análisis del tráfico de red de Amazon Alexa", Universitat Politècnica de Catalunya, 2021, https://upcommons.upc.edu/bitstream/handle/2117/348745/155639.pdf

Ismael Castell-Uroz (icastell@ac.upc.edu) is a Ph.D. student at the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, where he received the B.Sc. degree in Computer Science in 2008 and the M.Sc. degree in Computer Architecture, Networks and Systems in 2010. He has several years of experience in network and system administration and currently holds a Projects Scholarship at UPC. His expertise and research interest are in computer networks, especially in the field of network monitoring, anomaly detection, internet privacy and web tracking.

Pere Barlet-Ros (pbarlet@ac.upc.edu) is currently an Associate Professor with the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, and Scientific Director at the Barcelona Neural Networking Center (BNN-UPC). He received the M.Sc. and Ph.D. degrees in Computer Science from UPC, in 2003 and 2008, respectively. From 2013 to 2018, he was Co-founder and Chairman of the machine learning startup Talaia Networks. His research has focused on the development of novel machine learning technologies for network management and optimization, traffic classification and network security, which have been integrated in several open-source and commercial products, including Talaia, Auvik TrafficInsights, Intel CoMo and SMARTxAC.

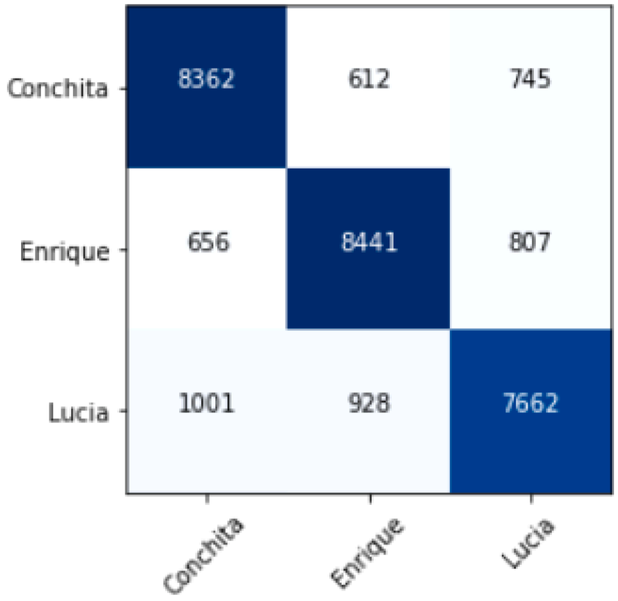


Fig. 4. Confusion matrix.

Ruben Barcelo Armada (ruben.barcelo@estudiantat.upc.edu) is a graduate in Informatics Engineering from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently studying a Master's Degree in Cybersecurity. His research interests include network security and understand the risks of the Internet of Things (IoT) devices, in order to preserve privacy, confidentiality and security of the data and users.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Lau, B. Zimmerman, F. Schaub, Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers, in: Proceedings of the ACM on Human-Computer Interaction, 2018, pp. 1–31, https://doi.org/10.1145/3274371.

[2] I. Castell-Uroz, X. Marrugat-Plaza, J. Solé-Pareta, P. Barlet-Ros, CoNEXT '19 Companion, in: Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies, December 2019, pp. 4–6, https://doi-org.recursos.biblioteca.upc.edu/10.1145/3360468.3366769.

[3] Ruben Barcelo Armada, Ismael Castell-Uroz, Alexa Dataset Collection Scripts, Universitat Politècnica de Catalunya (2021). https://github.com/RubenBar/AlexaDataset.

[4] Ruben Barcelo Armada, Pere Barlet Ros, Disección y análisis del tráfico de red de Amazon Alexa, Universitat Politècnica de Catalunya (2021).

Ruben Barcelo Armada (ruben.barcelo@estudiantat.upc.edu) is a graduate in Informatics Engineering from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently studying a Master's Degree in Cybersecurity. His research interests include network security and understand the risks of the Internet of Things (IoT) devices, in order to preserve privacy, confidentiality and security of the data and users.

Ismael Castell-Uroz (icastell@ac.upc.edu) is a Ph.D. student at the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, where he received the B.Sc. degree in Computer Science in 2008 and the M.Sc. degree in Computer Architecture, Networks and Systems in 2010. He has several years of experience in network and system administration and currently holds a Projects Scholarship at UPC. His expertise and research interest are in computer networks, especially in the field of network monitoring, anomaly detection, internet privacy and web tracking.

Pere Barlet-Ros (pbarlet@ac.upc.edu) is currently an Associate Professor with the Computer Architecture Department of the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, and Scientific Director at the Barcelona Neural Networking Center (BNN-UPC). He received the M.Sc. and Ph.D. degrees in Computer Science from UPC, in 2003 and 2008, respectively. From 2013 to 2018, he was Co-founder and Chairman of the machine learning startup Talaia Networks. His research has focused on the development of novel machine learning technologies for network management and optimization, traffic classification and network security, which have been integrated in several open-source and commercial products, including Talaia, Auvik TrafficInsights, Intel CoMo and SMARTxAC.