SN Computer Science (2020) 1:272
https://doi.org/10.1007/s42979-020-00285-x

ORIGINAL RESEARCH

Forensic Investigation of Google Assistant

Alex Akinbi[1] · **Thomas Berry**[1]

Received: 1 April 2020 / Accepted: 3 August 2020

Abstract
Google Nest devices have seen a rise in demand especially with Google's huge advantage in search engine results and a complex ecosystem that consists of a range of companion devices and compatible mobile applications integrated and interacting with its virtual assistant, Google Assistant. This study undertakes the forensics extraction and analysis of client-centric and cloud-native data remnants left behind on Android smartphones by the Google Home and Google Assistant apps used to control a Google Nest device. We identified the main database and file system storage location central to the Google Assistant ecosystem. From our analysis, we show forensic artifacts of interest associated with user account information, the chronology and copies of past voice conversations exchanged, and record of deleted data. The findings from this study describe forensic artifacts that could assist forensic investigators and can facilitate a criminal investigation.

Keywords Mobile forensics · Internet of Things forensics · Android forensics · Google Assistant · Google Home · Google Home Mini

Introduction

The Internet of Things (IoT) industry has recently focused on creating internet-connected devices, which is predicted to be used in almost 40% of homes in the UK by 2022 [1]. This growing market has encouraged the proliferation and development of software applications and gadgets that enable remote monitoring and management of several IoT devices, especially in smart homes. It has created significant interest amongst digital forensic researchers and a paradigm shift towards a smart-home IoT forensic ecosystem and the evidence these devices produce [2]. Recently, there has been a huge demand by consumers in smart speakers such as Amazon Echo and Google Home, which feature a voice-activated digital assistant that allows users to control home automation hubs and other IoT devices using voice controls [3]. Google's voice-activated digital assistant like Amazon's Alexa, Apple's Siri, and Microsoft's Cortana is called Google Assistant. As a digital virtual assistant cloud service, Google Assistant interacts with various compatible devices, native Google applications, and some third-party applications by converting the voice requests to native communication protocols [4]. Google Nest devices (both the Nest and Google Home range) by default are compatible with Google Assistant. Google Assistant offers voice commands and conversational interactions that allow users to conduct internet voice searches, perform voice-automated device control, play music, connect with friends and family through video calls and video messages, etc. after using "OK Google" or "Hey, Google" wake words. A single Google Nest device can generate large amounts of data and network traffic as it can be connected to other companion devices and applications simultaneously, making them sources of forensic artifacts in forensic investigations. This always active, always generating characteristic makes them excellent digital witnesses, capturing traces of activities of potential use in investigations [5]. Figure 1 shows the typical architecture of the Google Assistant ecosystem [2] (based on the IoT forensics ecosystem) where all compatible and companion devices can interact with Google Assistant and the potential sources of forensic artifacts (network traffic, smart devices, cloud service, and mobile apps).

Law enforcement agents, legal experts, and forensic investigators have also taken a significant interest in IoT devices as sources of forensic artifacts [4], especially in

✉ Alex Akinbi
o.a.akinbi@ljmu.ac.uk

Thomas Berry
t.berry@ljmu.ac.uk

[1] Department of Computer Science, Liverpool John Moores University, Liverpool, UK

Published online: 14 August 2020

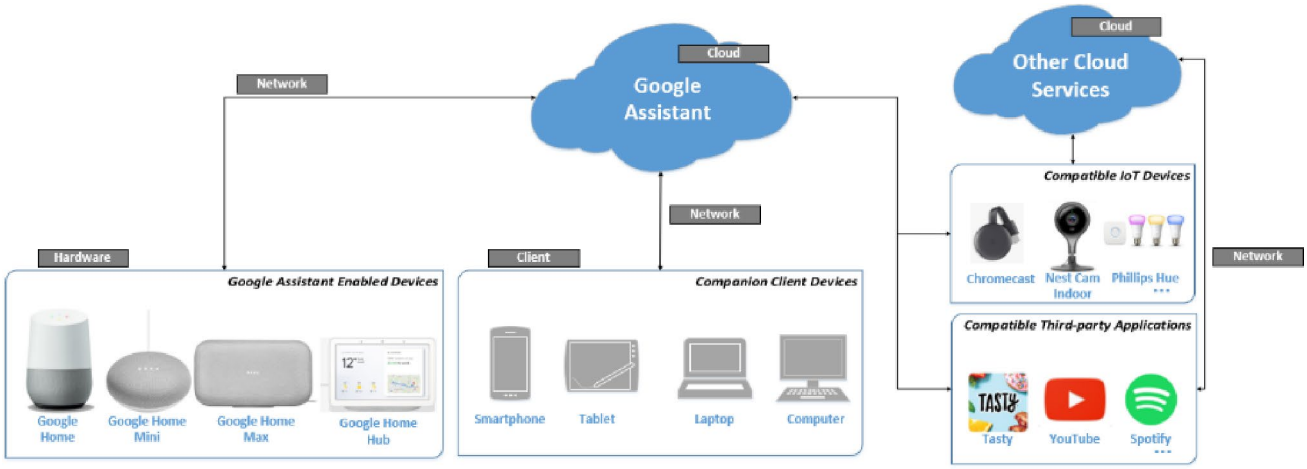


Fig. 1 Google Assistant ecosystem (a paradigm for smart-home IoT forensic ecosystem [2]) (Sources of device images: store.google.com)

scenarios where an IoT device has been a witness to a crime [6]. An example is a case in 2018, where a United States judge asked Amazon to hand over audio recordings from an Amazon Echo which was in a house where two women died [7]. The previous year, forensic evidence from a Fitbit was crucial in the conviction of a man suspected of killing his wife in Connecticut, USA [8]. Although IoT devices create opportunities for law enforcement and forensic investigators, there are huge challenges in the acquisition and analysis of forensic artifacts due to the quantity of data generated, the number and variety of devices, the heterogeneity of protocols used, and their distributed nature [5]. Besides, the storage capacity of smartphones that are synced with many smart-home IoT devices is increasing significantly with multiple apps installed which makes it difficult for investigators to identify data related to crimes for investigation [9].

In this study, we focus on the extraction and analysis of forensic artifacts of interest from the *Google Home* and *Google Assistant* apps installed and running on an Android smartphone and used to control a Google Nest device (Google Home Mini smart speaker). Our original contributions in this paper include the exploration and analysis of the client-centric and cloud-native forensic artifacts that can be found as summarised below:

- We recover local copies of conversations exchanged between a user and Google Assistant stored in the main databases and file system of the Android smartphone;
- We show a chronology of two-way conversations between the user and Google Assistant stored in these storage locations;
- We recover copies of past conversations exchanged and stored on the user's *My Activity* cloud service account.

This paper is organized as follows. In "Related Works", we discuss related works. In "Forensic Acquisition and Analysis Methodology", we discuss our forensic acquisition and analysis methodology including tools and their usage. In "Test Environment and Scenario", we discuss the test environment and scenario used in our experiments. Forensic analysis and findings from the technical experiment are presented in "Forensic Analysis and Findings". Finally, in "Conclusion and Future Work" we conclude the paper and highlight potential future research areas.

Related Works

IoT forensics has been widely studied in recent literature [2, 10–15]. There has also been recent literature and studies that have focused on recovering forensic artifacts of interests from virtual assistant enabled devices and their companion devices. Chung et al. [4], focused on the client-centric and cloud-native artifacts stored on companion clients of the Amazon Echo. Li et al. [16], analyzed forensic artifacts retrieved from the Amazon Echo as a use case to demonstrate their proposed forensic analysis model. Forensic analysis of Amazon echo was also conducted in a study by Shin et al. [17]. Hyde and Moran [18] presented a forensic examination of Alexa Echo Dot, generation 1 and 2, and the Echo generation 1 controlled by Android smartphones. They acquired data remnants directly from the motherboards of the Echo via the device's universal asynchronous transmitter receiver (UART) and companion mobile apps and retrieved the Wi-Fi information from the Alexa device's memory and user information from Alexa and Kasa mobile apps.

Other studies include forensic analysis of the *Securifi Almond+* (compatible with Amazon Alexa) [19], forensic analysis of Amazon Echo, and Nest Camera [5]. However,

none of these studies covers the forensic analysis of Google Assistant and Google Nest device (Google Home Mini smart speaker) on Android platforms, which is the focus of this paper. Therefore, to address this gap in the literature and make the most of the potential impact of our study, we show an up-to-date understanding of the client-centric artifacts that can be extracted from an Android smartphone that has been used to interact with Google Assistant.

Forensic Acquisition and Analysis Methodology

In this study, we adopted two approaches in the acquisition and analysis of forensic artifacts of interest. Our first approach was to conduct an experiment to generate data that could be analyzed. We created an investigative scenario in a controlled test environment, to address the two investigative questions that are commonly encountered in IoT digital investigations (Sect. 4). In the scenario, we registered a single Google test account using a Gmail address "behemothpentestlab@gmail.com" to access Google services and login to the Google Assistant and Google Home apps which we installed on the Android smartphone. To set up the Google Home Mini smart speaker, the Google Home, and Google Assistant apps must be installed [20]. We then proceed to use the Google Assistant app to initiate voice commands and two-way conversations with Google Assistant and control the Google Home Mini device. This was done to make the scenario realistic and retrieve as many forensic artifacts as possible.

In our second approach, we adopted Quick and Choo's method of evidence source identification, acquisition, and analysis [21, 22]. Client-centric forensic artifacts were identified and extracted from the Android smartphone (companion device) associated with the Google Assistant and Google Home apps by accessing the internal memory on the device. The primary goal was to conduct a post-mortem by imaging the smartphone and accessing data residing on the internal flash memory [17]. We accessed the internal memory of the Samsung Galaxy S9 smartphone running *Android Pie 9.0* using *Cellebrite UFED 4PC v. 7.28* forensic software. The software is a suitable commercial tool that exploits a boot loader vulnerability that exists via a locked screen bypass in a forensically sound manner. Finally, we analyzed the extraction using *Cellebrite Physical Analyzer v*. 7.25 [23].

To access the "My Activity" cloud service account and extract cloud-native artifacts containing the user's past conversations, we used the Google account credentials that were retrieved from the Android smartphone extraction. We parsed the credentials remotely using *Cellebrite's UFED Cloud Analyzer v 7.8* [24] and downloaded an archive that contains a chronology of conversations exchanged between the user and Google Assistant. The archive was then analyzed for forensic artifacts of interest using *Cellebrite Physical Analyzer v. 7.25*. For all the extractions, SHA256 hash values were calculated for original file verification. The list of each tool in our experiment and their usage is presented in Table 1.

Test Environment and Scenario

The experiment conducted in this study is structured around two investigative questions that are commonly encountered in IoT digital investigations and are listed as follows:

1. What client-centric forensic artifacts of interest can be obtained on an Android smartphone that has the Google Home and Google Assistant apps installed and integrated with the Google Home Mini smart speaker?
2. What cloud-native forensic artifacts of interest associated with Google Assistant can be retrieved from a user's personal "My Activity" Google cloud account?

In this section, we list the details of the test environment and scenario used in our experiments. In our scenario creation, all devices were connected to the same Wi-Fi local area network (LAN). The Google Home Mini smart speaker was voice-controlled using the Google Assistant app on the smartphone and the Google Home app is used to manage the speaker settings. We created a personal smart-home network for the Google Home Mini smart speaker and named it "Office speaker". YouTube Music (cloud service) was configured as the default service for playing music and *Google Chrome* browser app was also configured as the default browser by making changes in the Google Home app services and privacy settings. Finally, we configured the

Table 1 Forensic tools and usage

Forensics level	Hardware	Software	Software usage
Client-centric	Samsung Galaxy S9 (Android Pie 9.0)	Cellebrite UFED 4PC (v 7.28) Cellebrite Physical Analyzer (v 7.25)	To extract and analyze Android forensic memory image
Cloud-native		Cellebrite's UFED Cloud Analyzer v 7.8	To extract cloud data

Table 2 Test device and applications in the experiment

Detail	Description	
Google Nest device	Google Home Mini, *Model Number: GA00216-UK*	
Test device	Samsung Galaxy S9 *(Android Pie 9.0) Model: SM-G960F*	
Installed applications	Application name	Android version(s)
	Google Home App	2.14.50.11
	Google Chrome App	77.03
	Google Assistant App	0.1.187
	Google App	10.65
	YouTube Music	Cloud service
Gmail account	behemothpentestlab@gmail.com	

Table 3 Test scenario voice commands and requests

Voice commands	Date and time initiated
Create a shopping list	Oct 10, 2019, at 10:06 A.M UTC
Play Led Zeppelin on Office speaker	Oct 10, 2019, at 09:40 A.M UTC
Play music on Office speaker	Oct 10, 2019, at 09:39 A.M UTC
What's the time?	Oct 10, 2019, at 09:36 A.M UTC
Ok Google play the news	Oct 10, 2019, at 08:40 A.M UTC

smartphone to use *Google app* as the default search engine. These apps were all installed as recommended in the initial set up for Google Home Mini smart speaker [20]. A summary of test devices and applications installed in the experiment is shown in Table 2.

During the setup of the Google Home Mini smart speaker, we were required to train Google Assistant for voice recognition and identification with the Google Assistant app using the 'Ok Google' or 'Hey Google' wake words several times. According to Google, the Google Home Mini smart speaker may be used by all household members, and this setup process allows each member of the household to use voice match to customize their personal experience with the smart speaker [20]. Finally, we initialized a set of voice commands and conversations using the Google Assistant app to enable us to validate our findings and results (See Table 3).

Forensic Analysis and Findings

In this section, we present the client-centric forensic analysis and findings of Google Assistant on Android smartphones. In Sect. 5.2, we present the cloud-native forensic analysis and results.

Android Client-Centric Analysis and Findings

From the Android physical extraction, we found raw copies of voice recordings we used to train Google Assistant

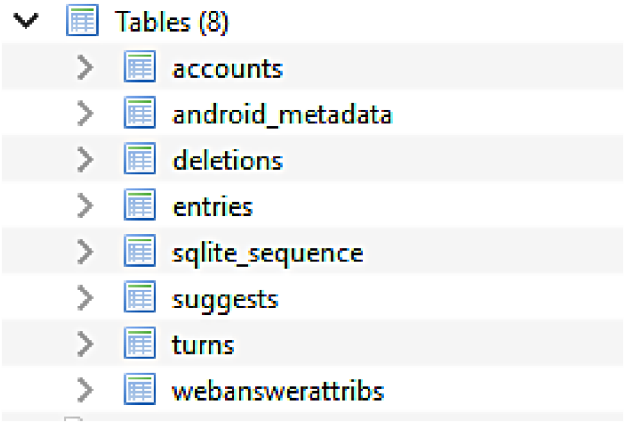


Fig. 2 Structure of the main *opa_history* database

using the *OK Google or Hey Google* wake words. These audio files are in the */data/data/com.google.android.googlequicksearchbox/app_sid* directory in the format "UTCtimestamp-behemothpentestlab@gmail.com-OK_HEY.pcm". The audio files were downloaded and played through Audacity by importing the raw data with encoding: signed 16-bit pcm, no endianness byte order, 1 channel (mono), 0-byte start offset at a sample rate 19000 Hz.

The most significant forensic artifacts of interest are stored in the main *opa_history* SQLite database. All voice conversations exchanged between the user and the Google Assistant are stored in this database and contains eight different tables (see Fig. 2). From our findings, only four out of these eight tables contain information of forensic interest namely "accounts", "entries", "turns" and "deletions". The *opa_history* database is located within the Google App package path directory "/data/data/com.google.android.googlequicksearchbox/databases". This is because upon receiving voice search queries, Google Assistant uses an intent filter called "*SEARCH_ACTION*", to launch the default search engine (Google App) to carry out the search and action queries on the internet. The voice search request is then passed to the default search engine

(Google App) for the query to be completed. We now discuss the contents of each table and how they correlate to answer questions from our scenario.

Identifying User Accounts

The *accounts* table contains information associated with the user's Gmail account and is stored in the "name" field and assigned a unique identifier which is stored in the "id" field (See Fig. 3).

Identifying Conversations Exchanged

The *entries* table store records of all voice conversations exchanged between the user and Google Assistant as a binary large object (BLOB) in the "entry" field. Each record stored in this field is assigned a unique identifier (primary key) and stored in the "id" field. *Cellebrite Physical Analyzer v. 7.25* can partially decode the text translation in the BLOB into plaintext which makes it human-readable (See Fig. 4). From this figure we can see in the 33rd record, a voice request "Play music on office speaker" and the 35th record shows the response from Google Assistant "OK, music from YouTube Music. Playing on Office speaker". In our scenario, we noticed that if Google Assistant needs to perform a request which requires access to a service such as YouTube music or Google search engine, the slight delay is recorded "android-linear-layout2" as shown in the 34th record in the table. In the figure, we also see our voice request in the 52nd record "create shopping list" and the response in the 53rd record "Alright starting a list called shopping…".

Reconstructing Chronology of Conversations Exchanged

The *turns* table (See Fig. 5), contains a timestamp for each entry in the *entries* table in chronological order. Each timestamp is assigned a unique identifier stored in the "id" field (primary key).

Both the *entries* and *turns* table are linked together by means of a foreign key in the *entries* table named "turn_id". The turns table is also linked to the *accounts* table by means of the foreign key named "account_id". To illustrate the chronology of voice conversations exchanged, we linked the relationships between the *accounts*, *entries*, and *turns* tables to reconstruct the date and time as shown in Fig. 6. In this figure, we see one distinct user account *behemothpentestlab@gmail.com* ("id = 1") in the *accounts* table is associated with the "account_id = 1" from the *turns* table. Likewise, the "id" field in the *turns* table is associated with the "turn_id" field from the *entries* table. In the 33rd record of the *entries* table (*id = 33, turn_id = 31 and event_id =* 'RPyeXffAHOzImwXc7KDQBQ'), we see the voice request "Play music on office speaker" was sent on the 10th Oct. 2019 at 09:39:17 A.M UTC (Unix Timestamp = '1570700357517'). The 35th record (*id = 35, turn_id = 32, and* 'RPyeXffAHOzImwXc7KDQBQ') shows the response from Google Assistant "OK, music from YouTube Music. Playing on Office speaker…" was received on the 10th Oct. 2019 at 09:39:17 A.M UTC (Unix Timestamp = '1570700357582'). The matching *event_id*, date, and time show a direct response to the voice request which represents a two-way conversation.

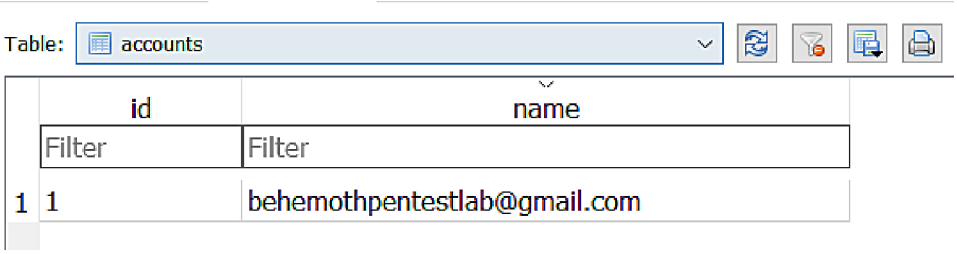
Dealing with Deleted Conversations

Record of deleted voice searches, requests, or conversations is stored in the *deletions* table. The table stores records of each deleted conversation as an event in the "event_id" field. The *account_id* and *event_id* fields both link the *deletions* and *turns* table together. This makes it possible to reconstruct a deleted event associated with a user account. In our scenario, we deleted a previous voice request ("Play music on office speaker") by logging into the user's online "My Activity" account. Figure 7 shows a record of the deleted data in the entries table *(id = 5)*. We observed a record of the deletion as shown in the *deletions* table *(id = 1)* and *turns* table *(id = 5)* match using the unique value stored in both tables' *event_id* fields. Using *Cellebrite Physical Analyzer's* DB Viewer carving tool, we were able to recover the deleted record from the *entries* table as shown in the figure.

Cloud-Native Forensics: Analysis and Findings

Copies of voice commands, requests, and searches initiated by the user with Google Assistant are automatically stored in real time in the user's *Google My Activity* cloud account. The record contains the test translation of a request and response

Fig. 3 *Accounts* table



entries (70)

id	turn_id	entry
33	31	play music on office speaker0yJ-JRPyeXffAHOzImwXc7...
34	31	D android-linear-layout2RPyeXfT-M4qEmwWvo72YD...
35	32	B :OK, music from YouTube Music. Playing on Office speaker...
36	34	) !Hi, Behemoth. How can I help you?0828202-JY_yeXciV...
37	35	android-linear-layout2Y_yeXdvQO-GEhbIPmeCG...
38	36	
39	36	OK8805-JcvyeXcDDIMTQkwWTza-QCw`
40	37	' #play Led Zeppelin on office speaker0-JgPyeXb6iL6...
41	37	D android-linear-layout2gfyeXaOFGYfVkwXdoIWID...
42	38	g _Sure, check out this YouTube Music station based on Led ...
43	40	1 -set up an hospital appointment for 2 tomorrow0-J...
44	40	5 -Sure, Hospital appointment tomorrow at 14:00.0808-J...
45	41	! Do you want to save that?0808-JMP-eXYiaEuKugwfF...
46	42	yes00H-JN_-eXb-JHLDX0gWQ94n4CQ`
47	42	$ Sure, it's on your calendar.0808H-JN_-eXb-JHLDX0...
48	43	A A)android-linear-layout2Pf-eXeeqConMwAKRm5LgBgJ...
49	45	, (read the headline news on office speaker0-JoP-eX...
50	46	' #stop reading news on office speaker0-J4_-eXdzaAY...
51	46	
52	47	create a shopping list0-JIwKfXaLOM-eHjLsPvcOYmAk...
53	47	L DAlright, starting a list called "shopping". What do you ...
54	48	milk0-JogKfXde7GYjxrgTWi7bwBg`
55	48	9 1OK, I made a list called shopping and added milk.0808-J...
56	50	! add bread to my shopping list0-JqwKfXaOWM5XMgwe5...
57	50	Alright, I added bread.0808-JqwKfXaOWM5XMgwe5mq...
58	52	add eggs to my shopping list0,-JsgKfXZKZH-_grgTVj...
59	52	OK, I added eggs.0808-JsgKfXZKZH-_grgTVjlrYAw`

Fig. 4 *Entries* table

Table: turns

	id	account_id	event_id	timestamp
	Filter	Filter	Filter	Filter
33	33	1	RPyeXffAHOzImwXc7KDQBQ	1570700357605
34	34	1	Y_yeXciVMqakgweA5pToBQ	1570700387247
35	35	1	Y_yeXciVMqakgweA5pToBQ	1570700387275
36	36	1	cvyeXcDDIMTQkwWTza-QCw	1570700402280

Fig. 5 *Turns* table

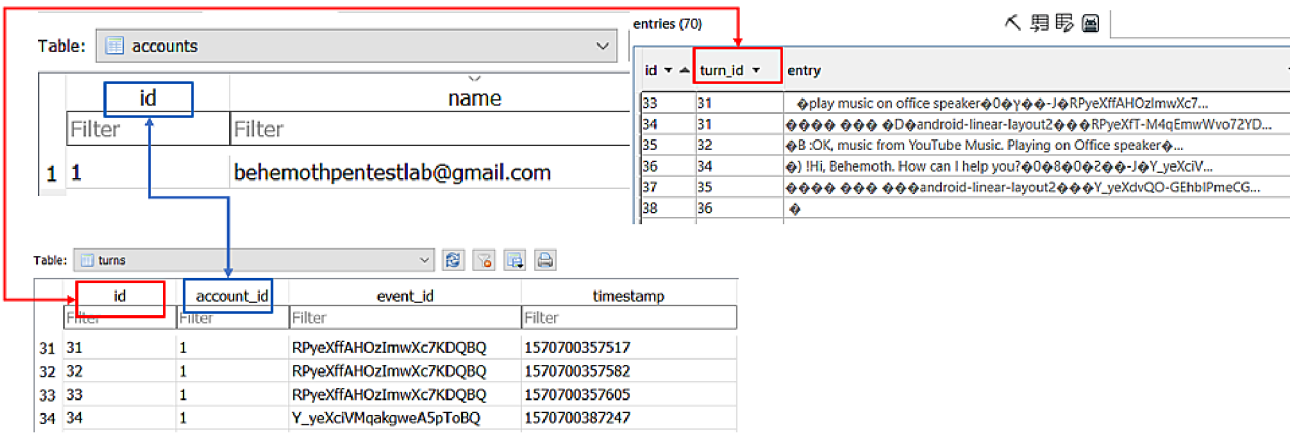


Fig. 6 Reconstructing voice conversations

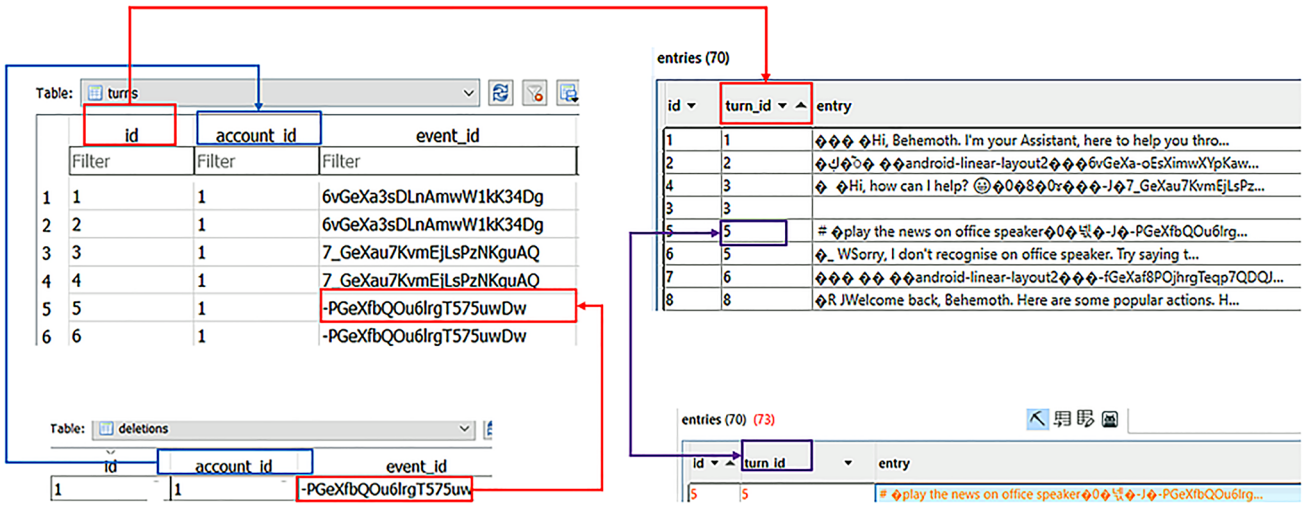


Fig. 7 Recovering deleted conversations

and an audio recording of the user's request which can be played back. In our scenario, we recovered the entire cloud-native archive which contains both text and audio logs in a zip file archive(Fig. 8). We parsed the user's credentials obtained from the forensic extraction into *Cellebrite UFED Cloud Analyzer* and were able to download raw copies of these files via *Google Takeout*. We describe the findings from our analysis of this cloud-native archive as follows.

All voice commands initiated in our test scenario using the Google Assistant app were stored as UTC time-stamped mp3 audio files (Fig. 6). For example, the voice command "Play Led Zeppelin on office speaker" initiated on the 10th Oct. 2019 at 09:40 AM UTC was stored in the format "2019-10-10_09_40_18_590_UTC.mp3". We were able to extract, play and listen to the voice recording. A log of all text translations of each voice request and search are appended with their respective audio mp3 file name and saved in an HTML file named "MyAtivity.html" (See Fig. 9). We were also able to recover the shopping list we created using voice controls on October 10, 2019, at 11:06 A.M stored as a CSV file named "shopping2019-10-10_10_06_26.791.csv". Table 4 shows a description of the forensic relevance of each artifact of interest recovered from the cloud-native archive.

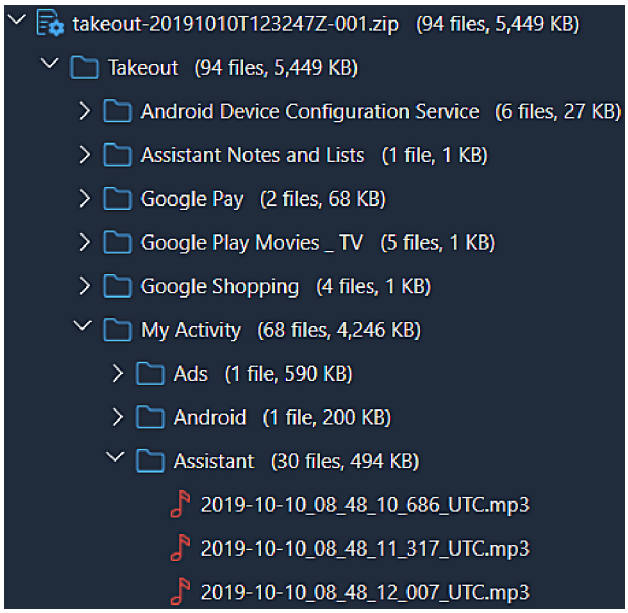


Fig. 8 Exported UFDR zip file imported into Cellebrite Physical Analyzer

Conclusion and Future Work

Apps and devices integrated with Google Assistant can be a silent witness to a crime. The always listening capability of this virtual assistant and record keeping of past conversations can be crucial in criminal investigations. For forensic investigators, being able to recover these forensic artifacts, reconstruct the sequence of events and recover deleted data is vital. In this paper, we discussed the forensic analysis of Google Assistant, a virtual assistant developed by Google and primarily available on mobile and smart home IoT devices. We showed client-centric forensic artifacts stored in the main *opa_history* SQLite database on Android smartphones which contain all local copies of voice conversations exchanged with Google Assistant.

We were able to reconstruct past conversations, time and date of occurrence, identify user account information, and recover deleted conversations from the database tables. We have also shown how cloud-native artifacts which hold records of all past conversations stored in the user's *My Activity* cloud account.

In our experiments, we observed requests made by the user to the Google Home Mini device needs to begin with *Ok Google* or *Hey Google* wake word. However, its use is not a prerequisite for conversations initiated using

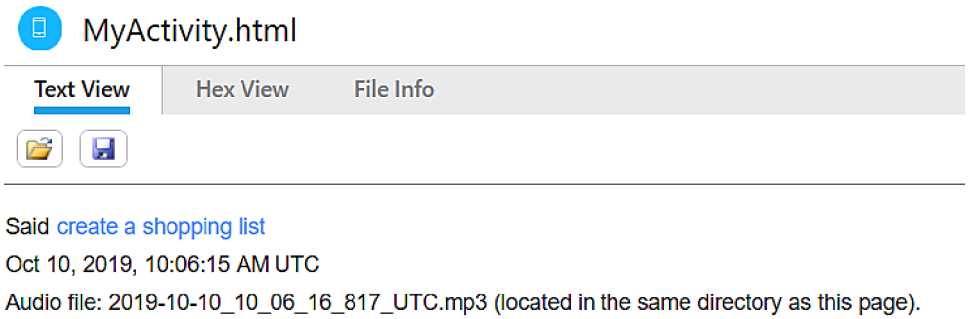


Fig. 9 Logs saved as MyActivity.html

Table 4 Summary of contents from My Activity cloud extraction

Folder name	Contents	Application in digital forensics
Assistant	This folder includes time-stamped audio MP3 files, which contains exchanged conversations with Google Assistant A file named *Activity.html*- text log of user voice interaction with Google Assistant and associated mp3 file name of each voice request recorded	Provides investigators with raw copies of audio recordings and timestamp of conversations exchanged
Assistant notes and links	CSV file with *Shopping List with items in plain text: shopping2019-10-10_10_06_26.791.csv*	Folder stores information associated with reminders created by the user
Search	*MyActivity.html* file contains a log of exchanged conversations in plaintext	Provides investigators with information associated with past audio recordings in plaintext, a chronology of conversations exchanged, and the GPS locations of the smartphone if enabled
Voice and audio	This folder includes several MP3 files, which include audio and voice recordings of user interactions with the Google Assistant	Provides investigators with raw copies of audio recordings and chronology of past conversations

the Google Assistant app provided the microphone button is active. Therefore, audio recordings can be accidentally recorded and saved. All voice conversations initiated by the user are stored in mp3 audio formats in the *Video and Audio* folder in the *My Activity* cloud account. However, in the main database, they are stored as BLOBs, a combination of multimedia objects and text. Using a database viewer, some of the text in the BLOBs can be viewed and can provide valuable forensic information in cases where a user has deleted similar records stored in *My Activity* cloud account.

Before this research, there have been few works of literature on the forensic analysis of Google Assistant-enabled devices and mobile apps. In this paper, we showed forensic acquisition and analysis of client-centric data remnants left behind by the Google Assistant and Google Home apps synced with Android smartphones and used to control the Google Home Mini smart speaker. We also described how these artifacts can assist forensic investigators in scenarios where Google Assistant is a witness to a crime.

In our future work, we intend to analyze other potential sources of forensic artifacts which include network and the Google Nest devices for remnants of forensic value. Our future work should also involve Google Assistant analysis on other platforms including iOS.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Statista. Share of households with internet access in the United Kingdom (UK) and the European Union (EU28) from 2007 to 2018. 2019. [Online]. https://www.statista.com/statistics/275043/percentage-of-households-with-internet-access-in-the-uk-and-eu/. Accessed 20 Dec 2018.
2. Dorai G, Houshmand S, Baggili I. I know what you did last summer. In: Proceedings of the 13th international conference on availability, reliability and security - ARES 2018, 2018, pp. 1–10. https://doi.org/10.1145/3230833.3232814.
3. Ilunin I. The most popular smart home devices of 2017. 2017. [Online]. https://hackernoon.com/the-most-popular-smart-home-devices-of-2017-830f479f09df. Accessed 28 Oct 2018.
4. Chung H, Park J, Lee S. Digital forensic approaches for Amazon Alexa ecosystem. Digit Investig. 2017;22:S15–S25. https://doi.org/10.1016/j.diin.2017.06.010.
5. Servida F, Casey E. IoT forensic challenges and opportunities for digital traces. Digit Investig. 2019. https://doi.org/10.1016/j.diin.2019.01.012.
6. Salama U. Investigating IoT crime in the age of connected devices. 2017. [Online]. https://securityintelligence.com/investigating-iot-crime-in-the-age-of-connected-devices/. Accessed 11 Jul 2019c
7. BBC. Amazon asked to share Echo data in US murder case. BBC.co.uk, 2018. [Online]. https://www.bbc.co.uk/news/technology-46181800. Accessed 03 May 2020.
8. Hauser C. In connecticut murder case, a fitbit is a silent witness. New York Times, 2017. [Online]. https://www.nytimes.com/2017/04/27/nyregion/in-connecticut-murder-case-a-fitbit-is-a-silent-witness.html. Accessed 04 May 2020.
9. Kim D, Lee S. Study of identifying and managing the potential evidence for effective Android forensics. Forensic Sci Int Digit Investig. 2020. https://doi.org/10.1016/j.fsidi.2019.200897.
10. Kang S, Kim S, Kim J. Forensic analysis for IoT fitness trackers and its application. Peer-to-Peer Netw Appl. 2018. https://doi.org/10.1007/s12083-018-0708-3.
11. MacDermott A, Lea S, Iqbal F, Idowu I, Shah B. Forensic analysis of wearable devices: Fitbit, Garmin and HETP Watches. In: 2019 10th IFIP international conference on new technologies, mobility and security, NTMS 2019 - proceedings and workshop, 2019. https://doi.org/10.1109/NTMS.2019.8763834.
12. Baggili I, Oduro J, Anthony K, Breitinger F, McGee G. Watch what you wear: preliminary forensic analysis of smart watches. In: Proceedings - 10th international conference on availability, reliability and security, ARES 2015, 2015, pp. 303–311. https://doi.org/10.1109/ARES.2015.39.
13. Yaqoob I, Hashem IAT, Ahmed A, Kazmi SMA, Hong CS. Internet of things forensics: recent advances, taxonomy, requirements, and open challenges. Future Gener Comput Syst. 2019;92:265–75. https://doi.org/10.1016/j.future.2018.09.058.
14. Akatyev N, James JI. Evidence identification in IoT networks based on threat assessment. Future Gener Comput Syst. 2019. https://doi.org/10.1016/j.future.2017.10.012.
15. Goudbeek A, Choo KKR, Le-Khac N-A. A forensic investigation framework for smart home environment. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), 2018, pp. 1446–1451. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00201.
16. Li S, Li S, Choo K-KR, Sun Q, Buchanan WJ, Cao J. IoT fforensics: Amazon Echo as a use case. IEEE Internet Things J. 2019. https://doi.org/10.1109/JIOT.2019.2906946.
17. Shin C, Chandok P, Liu R, Nielson SJ, Leschke TR. Potential forensic analysis of IoT data: an overview of the state-of-the-art and future possibilities. In: 2017 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), 2017, vol. 2018-Janua, pp. 705–710. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.182.
18. Hyde J, Moran B. Alexa, are you Skynet? SANS Digital Forensics and Incident Response Summit. 2017. [Online]. http://www.osdfcon.org/presentations/2017/Moran_Hyde-Alexa-are-you-skynet.pdf. Accessed 31 Mar 2020.

19. Awasthi A, Read HOL, Xynos K, Sutherland I. Welcome pwn: Almond smart home hub forensics. In: Proc. Digit. Forensic Res. Conf. DFRWS 2018 USA, vol. 26, pp. S38–S46, 2018. https://doi.org/10.1016/j.diin.2018.04.014.
20. Google. Set up Google Home. Google Nest Help, 2020. [Online]. https://support.google.com/googlenest/answer/7029485?co=GENIE.Platform%3DiOS&hl=en-AU&oco=0. Accessed 31 Mar 2020.
21. Quick D, Choo KKR. Digital droplets: Microsoft SkyDrive forensic data remnants. Future Gener Comput Syst. 2013;29(6):94–1378. https://doi.org/10.1016/j.future.2013.02.001.
22. Quick D, Choo KKR. Google drive: forensic analysis of data remnants. J Netw Comput Appl. 2014;40:93–179. https://doi.org/10.1016/j.jnca.2013.09.016.
23. Cellebrite. Cellebrite UFED 4PC and Physical Analyzer. 2020. [Online]. https://www.cellebrite.com/. Accessed 31 Mar 2020.
24. Cellebrite. Cellebrite Cloud Analyzer. 2019. [Online]. https://www.cellebrite.com/. Accessed 3 May 2020.