Quantifying Completeness of Reconstructed Scenarios: A Case Study on Echo Show

Sarfraz Shaikh
College of Public Affairs
University of Baltimore
Baltimore, U.S.A
sarfraz.shaikh@ubalt.edu

Weifeng Xu
College of Public Affairs
University of Baltimore
Baltimore, U.S.A
wxu@ubalt.edu

***Abstract*—This study presents a novel approach to quantifying the completeness of reconstructed scenarios in IoT forensics, focusing on the Amazon Echo Show as a case study. We introduce a systematic methodology for simulating real-world user interactions in a controlled environment, extracting evidence from the device's internal storage, and reconstructing scenarios based solely on this extracted data. Our empirical study, conducted on an Echo Show 8 device, simulated eleven common use cases over 21 days. We employed chip-off techniques to access the device's eMMC storage and utilized open-source forensic tools for evidence extraction. The results demonstrate that four out of eleven scenarios (network activity, media capture, video, and music streaming) can be fully reconstructed (100% completeness). Five scenarios (device setup, navigation, communication, photo browsing, and web browsing) can be partially reconstructed, while the remaining two (smart home control and scheduling) cannot be reconstructed at all. This research contributes to the field of digital forensics by providing a replicable framework for assessing completeness of reconstructed scenarios. Our study offers ways to quantify evidence and scenario reconstruction, potentially suggesting a method to evaluate the reconstruction potential of forensic tools, methods, or data sources.**

***Index Terms*—Digital Forensics, IoT Forensics, Scenario Reconstruction, Quantifiable Digital Evidence**

I. Introduction

The rise of Internet of Things (IoT) devices has transformed everyday interactions, generating vast data streams that hold significant forensic value. Yet, analyzing such devices poses challenges due to closed ecosystems and the division of data between local and cloud storage. A key difficulty—particularly when only one source is accessible—is assessing the completeness of extracted evidence. In this context, completeness refers to how fully a scenario (i.e., a sequence of user events) can be reconstructed from the data. Accurate scenario reconstruction is critical for ensuring the reliability and legal admissibility of forensic findings.

In forensic investigations, reconstructing scenarios with high completeness is critical for establishing accurate timelines, verifying user actions, and drawing reliable conclusions. Incomplete evidence can result in misinterpretations or overlooked details, which may compromise the integrity of an investigation. This challenge is amplified in cases where cloud-stored data is inaccessible due to legal, technical, or jurisdictional limitations. In such situations, investigators must rely solely on data extracted from local storage and assess whether it is sufficient to construct a coherent and legally defensible account of events.

Despite progress in IoT forensic methods, a key challenge persists: the lack of systematic techniques to assess the completeness of reconstructed scenarios. This gap forces forensic investigators to make critical judgments based on partial data.

This study addresses that gap by proposing a framework to evaluate the completeness of scenarios reconstructed from internal storage, using Amazon Echo Show as a case study. The choice is timely—Echo device sales rose from 0.6 million in 2014 to 32 million in 2018, with projections nearing 130 million units globally by 2025 [1].

We propose a novel reverse engineering approach that compares simulated scenarios **simulated scenarios** (real-world activities mimicked in a controlled environment) and **reconstructed scenarios** (reenactments of actual events based on extracted evidence), allowing forensic analysts to quantify reconstruction accuracy and better understand the evidential limitations of IoT devices.

The research follows a sequential methodology: identify common Echo Show use cases; simulate these in a controlled environment; document expected evidence; acquire the eMMC image via chip-off; extract data using open-source tools; and reconstruct scenarios. These reconstructed scenarios are then compared against the simulations to quantify gaps using simple metrics.

While our focus is the Echo Show, the framework can assess how much of a scenario can be reconstructed using a particular tool, forensic method, or data source (e.g. local or cloud).

Our key contributions are: **(1) Methodological Framework:** A replicable framework for scenario reconstruction and completeness assessment in IoT forensics. **(2) Benchmarking Tool:** A reverse engineering-based benchmark to evaluate forensic methods and tools. **(3) Comprehensive Extraction:** Broader local data recovery from Echo Show than prior work, revealing new insights into user activity and privacy risks.

Paper Outline: Section 2 reviews Echo data storage and related studies. Section 3 details our research design. Section 4 presents the Echo Show 8 case study and findings. Section 5 concludes with implications and future work.

This work is supported in part by BJA under 2019-DF-BX-K001.

II. Literature Review

A. Background of Data Storage in Echo Show Devices

Understanding Echo Show's data storage is essential for digital forensic investigations, yet Amazon provides limited information. The company claims voice commands are stored solely in the cloud [2], but on-device data specifics remain vague. Investigations by Privacy International into Echo Dot devices reveal some local storage of caches, logs, and Wi-Fi credentials [3]. Other studies suggest device settings, app data, temporary files, and limited user data may also be stored locally. However, the scope and retention of this data remain unclear, complicating scenario reconstruction efforts.

B. Extracted Evidence Types from Echo Devices

Existing studies mainly focus on data extraction from the cloud and companion devices, with minimal exploration of local storage on the Echo Show.

Chung et al. examined smart home ecosystems and proposed a forensic method for Echo Dot devices, relying on cloud and client data (such as mobile phones), but omitted analysis of local storage and evidence completeness [4]. Li et al. introduced an IoT forensic model applied to Amazon Echo, also centered on cloud and connected phones [5]. Giese and Noubir highlighted security flaws in Echo Dot, including exposed Wi-Fi credentials and data related to connected devices, without addressing scenario reconstruction [6].

More relevant is Youn et al., who extracted data from Echo Show devices including user accounts, logs, and video history, but noted gaps—such as missing photos [7]. Our study extends this by recovering a broader set of artifacts, including photos, and introducing a framework to assess scenario completeness.

Lorenz et al. used chip-off and ISP techniques on eight Echo Show models, extracting emails, call data, screenshots, and logs [8]. However, they did not evaluate whether this data suffices for reconstructing full scenarios.

The literature reveals a critical gap: limited research exists on extracting evidence from the Echo Show's local storage when cloud access is unavailable, and no established method exists to assess the completeness of reconstructed scenarios. Our research addresses this by identifying recoverable evidence on the device and introducing a novel method to measure reconstruction completeness. This approach not only advances Echo Show investigations but also offers a generalizable framework for evaluating scenario reconstruction across diverse tools, data sources, and methodologies.

III. Research Methodology

A. Reverse Engineering Design

Reverse engineering—deconstructing a system to understand its internal mechanisms [9]—is central to our approach. We applied this method to examine the Echo Show's local storage and uncover digital traces from user interactions, which were used to reconstruct scenarios and assess completeness.

In a controlled environment, we simulated realistic user activities to generate data both locally and in the cloud. This setup enabled us to evaluate the extent to which data is retained on the device. By comparing the simulated activities with the recovered artifacts, we measured the degree of scenario reconstruction achievable solely from internal storage.

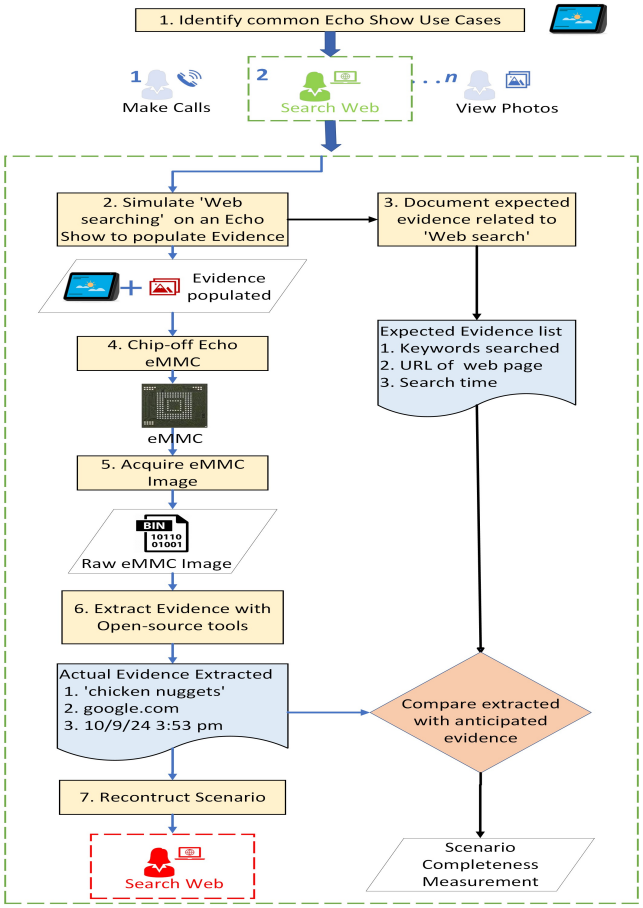

Fig. 1: Reverse engineering approach analyzing gap between simulated and reconstructed scenarios on Echo Show. Figure shows one event from 'Web Searching' use case.

Figure 1 outlines the steps of our methodology.

Step 1 (Identify Common Use Cases): We began by identifying realistic use cases for the Echo Show—such as web searches, video playback, and voice calls—to reflect common user behavior. These scenarios, composed of multiple interrelated actions, provide a basis for generating expected data and serve as a benchmark for comparison during analysis.

Step 2 (Simulate Common Use Cases): The identified scenarios were simulated in a controlled environment to ensure data integrity and avoid contamination. Interactions—e.g., voice-initiated web searches—produced data stored locally and in the cloud. For this study, we focused on data on the device to assess what can be recovered from internal storage alone.

Step 3 (Documenting Expected Evidence): For each simulated use case, we documented the specific evidence expected to be generated (e.g., search keywords, accessed URLs, timestamps). This set of anticipated artifacts formed the baseline for evaluating scenario reconstruction accuracy.

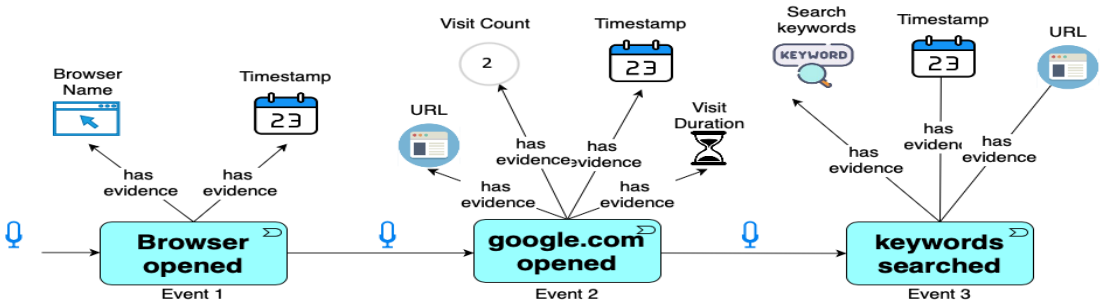

Fig. 2: A Simulated Web Searching Scenario showing Events & Evidence Artifacts

Step 4 (Chip-off): Since traditional access methods (e.g., rooting) are disabled on the Echo Show, we used a chip-off technique to remove the eMMC storage chip. This provided direct access to the device's binary data for forensic analysis.

Step 5 (Acquire Image): A raw bit-by-bit image of the eMMC chip was acquired using specialized software, preserving the integrity of data which was verified using hash values.

Step 6 (Extract Evidence): We applied forensic tools to the acquired image to extract artifacts corresponding to each use case. For instance, in the web search scenario, relevant data included search terms, URLs, and timestamps.

Step 7 (Reconstruct and Analyze Scenarios): Finally, we reconstructed user scenarios using the extracted artifacts and compared them with the original simulations. Discrepancies were measured using defined metrics to evaluate the completeness and accuracy of the reconstruction.

B. Reconstructing Scenarios and Event Decomposition

We employ a structured approach to reconstruct scenarios from artifacts extracted from Echo Show. Each scenario consists of distinct, sequential, and interconnected events [10]. Each event produces specific evidence. For example, a "Web Searching" scenario includes events like opening a browser, navigating to a site, and performing a search. These events generate evidence such as timestamps, URLs, and queries.

Let S represent a scenario, E_i denote an event, and A_j represent evidence artifacts:

Scenario: A scenario can be defined as:

$$S = \{E_1, E_2, \ldots, E_n\} \tag{1}$$

where S is the scenario and E_1, E_2, E_n are the events.

Event: An event is a set of evidence artifacts or attributes:

$$E_i = \{A_1, A_2, \ldots, A_m\} \tag{2}$$

where E_i is an event and A_1, A_2, A_m are the artifacts.

For example, a "Web Searching" scenario can be represented by equation 1 as sets of three events: Where

$$E_1(Open\_browser) = \{\text{timestamp, browser name}\} \tag{3}$$

$$E_2(Navigate\_to\_webpage) = \{\text{timestamp, URL}\} \tag{4}$$

$$E_3(Search) = \{\text{timestamp, search terms}\} \tag{5}$$

C. Measuring Completeness

To assess how well the reconstructed scenario reflects the simulation, we define two metrics: Event Completeness Ratio (ECR) and Scenario Completeness Ratio (SCR).

Event Completeness Ratio (ECR): It measures the completeness of event reconstruction and is calculated by dividing the number of extracted artifacts to the expected artifacts.

Scenario Completeness Ratio (SCR): This measures the completeness of a reconstructed scenario and is simply the *weighted average* of the ECRs.

ECR and SCR can be calculated by the formulae mentioned in equations 6 and 7:

$$\text{ECR}_i = \frac{N_{E_i}^{\text{Extracted}}}{N_{E_i}^{\text{Expected}}} \tag{6}$$

$$\text{SCR} = \frac{\sum_{i=1}^{n} (\text{ECR}_i \times w_i)}{\sum_{i=1}^{n} w_i} \tag{7}$$

where:

- SCR = Scenario Completeness Ratio
- ECR_i = Event Completeness Ratio for event i
- w_i = Weight for event i, typically the number of expected evidence artifacts
- n = Total number of events in the scenario
- $N_{E_i}^{\text{Extracted}}$ is the number of evidence artifacts extracted,
- $N_{E_i}^{\text{Expected}}$ is the expected number of evidence artifacts.

IV. Empirical Study

We conducted an empirical study to evaluate the gap between simulated and reconstructed scenarios on an Echo Show 8. This controlled experiment assessed the extent to which scenarios could be reconstructed solely using data extracted from the device's internal storage. Out investigation was guided by two research questions:

- **Scenario Reconstruction** Which, and how many, scenarios can be reconstructed solely from internal storage?
- **Completeness of Scenarios:** How completely do reconstructed scenarios reflect the simulated ones?

A. Controlled Environment Setup

To ensure isolation and consistency, the following equipment and software were used in a controlled lab setting.

- Echo Show 8 (2nd-gen, Fire OS 7.5.6.0, 8GB storage)

TABLE I: List of Simulated Scenarios, Associated Events and Evidence Artifacts

Scenario	Events	Evidence Artifacts
Device setup	User selects language & Wifi User logs in User configures settings	Language, Wifi SSID & password Login ID, password, timestamp Timezone, address, device name, device location
Smart home control	User connects smart bulb to Echo Show User switches smart bulb on/off User changes brightness/color	Device name, network address, connection timestamp On/off state, timestamp brightness level/color, timestamp
Connection to network	Echo Show connects to Wi-Fi Bluetooth device is paired	SSID, PSK, connection timestamp Device name, MAC Address, connection timestamp
Capturing photos/videos	User captures media with device camera	Media file, file type, file size, capture timestamp
Photo browsing	User views photos on device User ask query about photo	Media file, file type, file size, view timestamp Query, timestamp, device response
Video streaming	User launches video streaming service User watches a movie/series episode	Streaming service name, launch timestamp Content type, title, watched duration, timestamp, subtext
Music streaming	User launches music streaming service User plays a music	Streaming service name, launch timestamp Song title, artist/album, listen timestamp
Scheduling	User creates a scheduling task (list/event) User adds items/content to a list/event	List/event name, creation timestamp Event date/time, list items, add items/content timestamps
Navigation	User searches a location User asks for directions & traffic	Location address, search timestamp Directions, traffic condition, timestamp
Web browsing & searching	User opens the web browser User visits a website User performs a keyword search	Browser name, opening timestamp URL, visit timestamp, visit count Search keywords, search timestamp
Communication	User creates a contact User makes a call/sends message	Contact name, phone number, add timestamp Callee name, callee number, call provider, timestamp, duration

- Windows 10 HP laptop with Easy JTAG, Kali Linux VM
- Hot air rework station, prying tools, Easy JTAG hardware
- Eleven scenarios (e.g., setup, browsing, video playback) simulated; 262 interactions over 21 days (see Table I).

B. Experimental Procedure

We followed the seven-step research methodology to conduct this experiment. Below is a detailed account of each step:

Common Use Cases: Eleven use cases were selected to reflect typical Echo Show interactions, such as device setup, photo capture, web browsing, and scheduling. Scenarios involving Alexa skills and games were excluded due to limited evidentiary value. To enable detailed comparison between simulated and reconstructed scenarios, each was broken down into events and associated artifacts (Section III(B)). Table I summarizes each scenario, its events, and expected artifacts.

Simulate Common Use Cases: We simulated eleven use cases over 21 days in a controlled environment to generate evidence on the Echo Show 8. Each interaction was documented with four details: use case, timestamp, issued command, and device response. This process produced a comprehensive dataset reflecting diverse user-device interactions for analysis.

Among the eleven use cases, only the first—Amazon account setup—must occur first, as it enables all subsequent activities. The remaining ten can be simulated in no particular order, given their independence.

Expected Evidence: For each simulated use case, we identified expected artifacts—such as MAC addresses, timestamps, media files, and URLs. Table I maps these evidence types to their corresponding events and scenarios.

Chip-off: To access the Echo Show's internal storage, we attempted to root the device via Android Debug Bridge (adb) in a Linux environment since Echo Show devices operate on Fire OS, a customized version of the Android OS [6]. However, the device was undetectable due to Amazon disabling debugging mode on second-generation Echo Shows and beyond, rendering rooting infeasible. Consequently, we disassembled the device to remove (chip-off) its embedded MultiMedia Card (eMMC) - the internal storage chip. The chip, located on the logic board, was identified an 8GB Samsung BGA 153 package via label printed on it. It was removed using a hot air rework station with a 170^0C–400^0C temperature profile over four minutes and 60% airflow [11].

Image Acquisition: A raw binary dump of the eMMC was acquired using the Easy JTAG tool. The eMMC image was acquired by placing the chip in the EasyJTAG hardware and reading it using the associated software.

Evidence Extraction: Open-source Kali Linux tools were used for evidence extraction. It involved twp phases:

(1) Image Mounting: Disk images are essentially files, and for an operating system (OS) to interact with the file system inside a disk image, the image must first be interpreted as a block device [12]. This was done by configuring the image as a block device, allowing the OS to treat it like a physical disk. To achieve this, we used `mmls` and `lsblk` commands to analyze the partition layout and structure. The image was then mounted using `losetup` and `mount` in a read-only mode to prevent data modifications. The acquired image contained 23 partitions, with 3 GB system and 3.5 GB userdata partitions being the most significant. The system partition mounted successfully, while the userdata partition required the "noload" option to preserve integrity by preventing journal writes.

(2) Storage Analysis: Forensic tools like `scalpel`, `foremost`, `photorec`, and utilities such as `grep`, `strings`, and `sqlite3` were employed for data carving, evidence extraction, and searching within the image. Tools

TABLE II: Measurement of Reconstructed Scenario

Sr.	Events	SE	RE	ECR	SCR
1	User selects language & Wifi	3	3	1	0.7
	User logs in	3	2	0.67	
	User configures settings	4	2	0.5	
2	User connects smart bulb	3	0	0	0
	User switches bulb on/off	2	0	0	
	User changes brightness/color	2	0	0	
3	Echo Show connects to Wi-Fi	3	3	1	1
	Bluetooth device is paired	3	3	1	
4	User captures media	4	4	1	1
5	User views photos on device	4	4	1	0.57
	User ask query about photo	3	0	0	
6	Video streaming launched	2	2	1	1
	User watches a movie/episode	5	5	1	
7	Music streaming launched	2	2	1	1
	User plays music	3	3	1	
8	User creates a scheduling task	2	0	0	0
	User adds items/content	3	0	0	
9	User searches a location	2	0	0	0.2
	User asks for directions	3	1	0.33	
10	User opens the web browser	2	1	0.5	0.86
	User visits a website	3	3	1	
	User searches keywords	2	2	1	
11	User creates a contact	3	0	0	0.37
	User makes a call/message	5	3	0.6	

Sr: Scenario (Sequential numbers refer to corresponding Table I scenarios)
SE: Artifacts in Simulated Events, **ECR** : Event Completeness Ratio
RE: Artifacts in Reconstructed Events, **SCR** : Scenario Completeness Ratio

like `dd` were used for creating forensic disk images, and `jq` parsed JSON data. Additionally, `date` converted timestamps, `bulk_extractor` extracted email addresses, and `exiftool` was used to examine media metadata. This multi-tool approach ensured comprehensive evidence extraction for scenario reconstruction.

C. Results and Analysis

This section presents the final step of our methodology - reconstruction of scenarios and comparison with simulated scenarios to assess completeness. Table II summarizes the findings, showing the number of artifacts tied to simulated and reconstructed events, along with event and scenario completeness ratios. Evidence artifacts were used to reconstruct individual events, which were then combined into complete scenarios. For instance, the browser name and timestamp reconstructed the 'opening browser' event, while URL, visit timestamp, and visit count formed the 'web browsing' event. Similarly, search keywords and timestamps reconstructed the 'web searching' event. These events were sequentially connected to recreate the 'web searching and browsing' scenario.

Device setup: The "Device Setup" scenario included three events with expected evidence attributes (see Table I). Reconstruction is detailed as follow: (1) User selects language & Wi-Fi: Language preference, SSID, and password were found in WifiConfigStore.xml - all three recovered. (2) User logs in: An email (likely login ID) was found in *alta.h2clientservice.db-wal*. Account activity and timestamps in *accounts_de.db* inferred login time. Password was not recovered - two out of three recovered. (3) User configures settings: Device name appeared in *deviceNameSharedPref.xml*, and street address in *com.amazon.kindle.otter.oobe*. Device location and timezone were not found - two out of four recovered. Overall, one event was fully reconstructed, two partially, yielding a 7% Scenario Completeness Ratio each. Additional findings included: device serial number, OS version, model, and user name or profile ID.

Smart home control: No evidence related to the smart light bulb was found, resulting in 0% completeness.

Connection to network: This scenario involves two events: (1) *Echo Show connects to Wi-Fi:* The *WifiConfigStore.xml* file revealed the SSID, pre-shared key, and connection timestamp. (2) *Bluetooth device paired:* The *bt_config.conf* file contained paired device names, MAC addresses, and pairing timestamps. All 3 out of 3 expected artifacts were recovered for both events, resulting in 100% event and scenario completeness.

User captures media: This scenario involved capturing six photos and recrdig two videos. All eight media files were successfully recovered from unallocated space using file carving. Metadata was located in *amzn1.account.AFTJBUB4-CIBBMH2S2AQDEY5QCN2A.mixtape.db*. Thus, all four evidence artifacts (media files and three metadats artifacts - file types, file sizes, and timestamps) were recovered, achieving 100% event and scenario completeness. Recovered photos were smaller than the sizes indicated in the database file mentioned above, likely due to compression by the device.

Photo browsing: This scenario included two events: (1) user views photos on the device, and (2) user asks query about photo. All evidence artifacts (media files, types, sizes, and view timestamps) were recovered for the first event, but none for the second, resulting in 1 out of 2 events completely reconstructed and hence 57% completeness.

Photos viewed on device included six captured by the device, 23 synced from Amazon Photos, and several displayed during music playback. However, during search only 12 JPG screenshots were found. Therefore, `foremost` carving tool was used which recovered 14,149 JPG files, that included all expected photos—confirming authenticity via content match, though file sizes were reduced. Interestingly, HEIF images (from iPhone via Amazon Photos) were carved as JPGs, indicating format conversion by the Echo Show. This behavior raises questions about the potential forensic significance.

Video Streaming: This scenario involved two events as shown Table II. Both events were fully reconstructed, achieving 100% scenario completeness. The *dbplaybackhistory.db* file contained detailed metadata for both events: playback timestamps, service name (for the first event), and content types, titles, watch durations, timestamps, and subtitles (for the second event). All of these were successfully recovered.

We also recovered carved JPG images which were screenshots of the videos played on device streaming service. Subtitles for one video was also found within a carved ELF file.

Music Streaming: All evidence related to the music streaming scenario, including song titles, album names, and playback timestamps, was successfully recovered from the log file located in *com.amazon.paladin/files/alexa-anchor*. The reconstruction achieved 100% completeness.

Scheduling: No scheduling tasks evidence was found, except a screenshot of a reminder, resulting in 0% completeness.

Navigation: This scenario achieved 20% completeness, as only user's home location artifact could be found.

Web browsing & searching: This scenario comprised three events. (1) User opens the web browser: Evidence indicating the browser opening was found but its timestamp was not found. (2) User visits a website: During the simulation of this scenario, 88 web pages were visited, with some revisited, making a total of 126 web page visits. We successfully retrieved evidence artifacts URLs, visit counts, and timestamps regarding these visits from file named *History.db*. (3) User searches Keywords: The *History.db* file also contained data about keyword searches, including the URL, timestamp, and keywords searched. This scenario achieved 86% completeness.

Communication: During simulation, we created two contacts, made three calls (audio, video via Alexa app, and phone network), sent a text message, and an audio message. But we found no traces of phone communication data during reconstruction. We found a deleted XML files carrying 'callee' name. Therefore, using `scalpel` carving tool, we carved XML files which contained call log artifacts like caller profile IDs, callee names, timestamps, and call provider (Alexa-to-Alexa or Alexa-to-Phone). However, callee phone numbers and call durations, contacts and messages were not found. This scenario reached 37% completeness.

Based on findings, answers to the research questions are:

- Reconstruction of Scenarios: Our investigation shows that complete scenario reconstruction solely from internal storage data is achievable in four scenarios: network activity, media capture, video and music streaming. Partial reconstruction was possible in five scenarios: device setup, photo viewing, web browsing, navigation usage, and communication. No reconstruction was achieved for smart home control and scheduling scenarios.
- Completeness of Reconstructed Scenarios: The completeness ratios for 11 scenarios are provided in Table II.

Data Repository: Full experiment details, eMMC image, and activity demos are available at our GitHub repository.

V. Discussion and Conclusion

This study presents a comprehensive methodology for evaluating the completeness and accuracy of reconstructed scenarios from IoT device data, using the Echo Show as a case study. Through a controlled reverse engineering approach, user interactions were simulated and scenarios reconstructed from data extracted from the device's internal storage. The results highlight both the potential and limitations of scenario reconstruction from IoT data sources, offering valuable insights for digital forensic investigations.

The core contribution of this research lies in the development of a framework for assessing the completeness of scenarios reconstructed from extracted data. This framework enables investigators to quantify the degree of completeness in scenario reconstructions or determine the reconstruction potential of extracted data. Our empirical study on Echo Show device revealed that complete scenario reconstruction was achievable in specific contexts such as network activity, media capture, and video and music streaming. However, partial reconstructions were noted in scenarios involving device setup, web browsing, viewing photos, and phone communication.

The ability to quantify scenario completeness offers several practical benefits for digital forensic investigations. First, it provides a measurable standard against which the reliability of reconstructed scenarios can be assessed, significantly enhancing the credibility of digital evidence. Additionally, measuring the reconstruction potential of extracted data can guide forensic investigators in developing effective methods, assist forensic scientists in creating reliable tools, and evaluate the reconstruction potential of various data sources.

Future research could build on this study by exploring the application of the proposed methodology to a broader range of IoT devices. Expanding the scope to include various brands and models will help generalize the findings and enhance the robustness of the framework. Additionally, investigating the integration of machine learning techniques to automate and improve the accuracy of evidence extraction and scenario reconstruction could further advance the field of digital forensics

Acknowledgment

This material is based upon work supported in part by the Bureau of Justice Assistance under 2019-DF-BX-K001.

References

[1] Statista. (2022) Worldwide amazon echo unit shipment. [Online]. Available: https://www.statista.com/statistics/1022701/worldwide-amazon-echo-unit-shipment/

[2] Amazon, "Alexa, echo devices, and your privacy - amazon customer service," https://www.amazon.com/gp/help/customer/display.html?nodeId=GVP69FUJ48X9DK8V, n.d., accessed on 01/20/2024.

[3] Privacy-International, "The mystery of the amazon echo data," https://privacyinternational.org/news-analysis/2819/mystery-amazon-echo-data, 2019, accessed on 01/20/2024.

[4] H. Chung, J. Park, and S. Lee, "Digital forensic approaches for amazon alexa ecosystem," *Digital investigation*, vol. 22, pp. S15–S25, 2017.

[5] S. Li, K.-K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "Iot forensics: Amazon echo as a use case," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6487–6497, 2019.

[6] D. Giese and G. Noubir, "Amazon echo dot or the reverberating secrets of iot devices," in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021, pp. 13–24.

[7] M.-A. Youn, Y. Lim, K. Seo, H. Chung, and S. Lee, "Forensic analysis for ai speaker with display echo show 2nd generation as a case study," *Forensic Science International: Digital Investigation*, vol. 38, p. 301130, 2021.

[8] S. Lorenz, S. Stinehour, A. Chennamaneni, A. B. Subhani, and D. Torre, "Iot forensic analysis: A family of experiments with amazon echo devices," *Forensic Science International: Digital Investigation*, vol. 45, p. 301541, 2023.

[9] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, and M. Tehranipoor, "A survey on chip to system reverse engineering," *ACM journal on emerging technologies in computing systems (JETC)*, vol. 13, no. 1, pp. 1–34, 2016.

[10] S. Bhandari and V. Jusas, "An abstraction based approach for reconstruction of timeline in digital forensics," *Symmetry*, vol. 12, no. 1, p. 104, 2020.

[11] S. Shaikh, L. Deng, and W. Xu, "A practical survey of data carving from non-functional android phones using chip-off technique," in *International Conference on Information Technology-New Generations*. Springer, 2024, pp. 43–50.

[12] K. C. Wang, *Block Device I/O and Buffer Management*. Cham: Springer International Publishing, 2015, pp. 345–358.