Forensic Science International: Digital Investigation 50 (2024) 301805

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

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

DFRWS APAC 2024 - Selected Papers from the 4th Annual Digital Forensics Research Conference APAC

Re-imagen: Generating coherent background activity in synthetic scenario-based forensic datasets using large language models

Lena L. Voigt [a,*], Felix Freiling [a], Christopher J. Hargreaves [b]

[a] Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
[b] Department of Computer Science, University of Oxford, Oxford, United Kingdom

ARTICLE INFO

Keywords:
Forensic datasets
User simulation
Large language models (LLM)
Forensic disk image generation
Data synthesis
Digital forensic education
ChatGPT

ABSTRACT

Due to legal and privacy-related restrictions, the generation of *synthetic* data is recommended for creating datasets for digital forensic education and training. One challenge when synthesizing scenario-based forensic data is the creation of coherent background activity besides evidential actions. This work leverages the creative writing abilities of large language models (LLMs) to generate personas and actions that describe the background usage of a device consistent with the created persona. These actions are subsequently converted into a machine-readable format and executed on a virtualized device using VM control automation. We introduce Re-imagen, a framework that combines state-of-the-art LLMs and a recent unintrusive GUI automation tool to produce synthetic disk images that contain arguably coherent “wear-and-tear” artifacts that current synthesis platforms lack. While, for now, the focus is on the coherence of the generated background activity, we believe that the proposed approach is a step toward more *realistic* synthetic disk image generation.

1. Introduction

High-quality forensic datasets are crucial for various digital forensics disciplines, such as education and training. As the use of real-world data is rarely possible due to legal and privacy-related restrictions, the creation of synthetic data appears to be the most viable solution. This has been explored in previous research, where apart from some work on injecting artifacts into disk images (Scanlon et al., 2017), the focus has been on automating the control of virtual machines (VMs) to reduce the amount of manual effort required in generating such datasets (Moch and Freiling, 2009, 2012; Du et al., 2021; Göbel et al., 2022; Wolf et al., 2024). Recent advances by Schmidt et al. (2023) have even enabled full external control of a VM. This means that no artifacts from the automation framework are created within the VM or resulting disk image. While there are still implementation improvements to make, with that part of the disk image generation process at least conceptually solved, the focus can now be directed to other time-consuming aspects of the workflow.

Before commencing automated control of VMs, a script of actions must be prepared. For scenario-based teaching, both overall scenarios and detailed actions that generate relevant evidential artifacts must be created. Typically, activity unrelated to the crime under investigation must be included as well. This is sometimes referred to as ‘background activity,’ ‘background noise,’ or ‘wear and tear.’ Synthetic datasets will more accurately represent those seen in real-world investigations if the activities performed are similar to those seen in real-world disk images in terms of the number, timing, and variety. Additionally, those actions should be semantically consistent in the scenario’s context, i.e., if a ‘suspect’ has a set of interests that form part of the background activity, then the actions should align with those interests. All this arguably contributes to realism in an intuitive understanding of the term. Taking a pragmatic view of the term and thereby avoiding its philosophical pitfalls, we echo the discourse on the concept of ‘realism’ in Section 2 and discuss the difficulties of definition in Section 8. Due to those difficulties, in Section 7, we resort to a qualitative evaluation of the coherence of the generated background activity, which we believe constitutes one aspect of ‘realism.’

Moreover, focusing on the generation of coherent background activity addresses a significant challenge for forensic analysts: finding relevant traces within large volumes of coherent but irrelevant data that obscures the relevant traces. Also, an approach generating background noise would allow the production of longer-running base images that can serve as a starting point for an instructor to add case-specific activity when needed.

[*] Corresponding author.
E-mail addresses: lena.lucia.voigt@fau.de (L.L. Voigt), felix.freiling@fau.de (F. Freiling), christopher.hargreaves@cs.ox.ac.uk (C.J. Hargreaves).

https://doi.org/10.1016/j.fsidi.2024.301805

Available online 18 October 2024

1.1. Contributions

This work presents Re-imagen, a framework that integrates emerging large language models (LLMs) with existing VM control automation tools, allowing the generation of synthetic forensic data with minimum instructor input. The aim is to leverage the creative writing abilities of LLMs to generate personas, corresponding daily schedules, and activity descriptions which can be translated automatically to instruct the VM control automation tool that interacts with a VM. As a result, background activity is performed on the machine so as to generate a disk image with a richer set of artifacts than previously possible that remain internally coherent. We demonstrate our approach with a prototype using GPT-4o as the LLM and the recently proposed GUI automation tool by Schmidt et al. (2023) for VM control automation. More specifically, the paper makes the following contributions.

1. An approach integrating LLMs with VM control automation to create scenario-based datasets with coherent background activity;
2. The overall system architecture of Re-imagen, a framework capable of delivering a full end-to-end workflow from instructor-provided specification through to a synthetic disk image;
3. Prototype implementations of critical components of said architecture;
4. Demonstration and evaluation of the workflow with two examples using GPT-4o and a recent non-intrusive GUI automation tool.

To the best of our knowledge, our work yields the world's first synthetic disk image based on an AI-generated storyline requiring only minimal and high-level initial interaction by an instructor. We provide the code of our prototype implementations as well as the full examples we use to demonstrate the workflow (including ChatGPT queries and outputs, intermediary results, disk images, log files, and screenshots) in a repository.[1]

1.2. Outline

This paper is structured as follows: Section 2 revisits the discussion regarding the realism of datasets within this work's particular context and how this may be measured. It also discusses existing VM control automation tools and the specific challenges of scenario-based disk image generation. Section 3 then outlines our methodology, with Sections 4 and 5, discussing this work's requirements analysis and the system architecture of Re-imagen. Using two case studies, Section 6 then demonstrates the overall workflow with a prototype implementation of Re-imagen, followed by an evaluation of our proposed system in Section 7. Section 8 discusses limitations and further work, and Section 9 provides conclusions.

2. Background and related work

This section explores the background and related work that motivates the need for automating further parts of scenario-based disk image generation to complement existing work on VM control automation tools.

2.1. Realistic synthetic datasets

Garfinkel et al. (2009) categorize datasets into: 'sampled data,' 'realistic data,' 'real and restricted data,' and 'real but unrestricted data.' Yannikos et al. (2014) differentiate 'synthetic data corpus generation' from 'manually reproducing real-world actions,' and Grajeda et al. (2017) consider datasets from multiple perspectives, e.g., by origin, which leads them to a categorization scheme of: 'experiment-generated,' 'user-generated,' and 'computer-generated' datasets. Then, when considering the type of dataset, they distinguish between 'malware datasets,' 'email datasets,' 'file sets,' 'RAM dumps,' 'images of computer drives,' 'images of other devices,' 'network traffic,' and 'scenarios for analysis.' Breitinger and Jotterand (2023) present a more comprehensive hierarchy with 'synthetic' vs. 'human-driven' creation of datasets at the top level. Then they subsume 'random data,' 'rule-based data,' 'AI-generated data,' and 'computer-simulated data' under synthetic data, the last of which they subdivide into 'test data' and 'scenario data.'

Taking all these classifications into account, the aim of automatically generating complex disk images for *educational purposes* clearly falls under 'scenarios for analysis' (Grajeda et al., 2017), due to the automation 'synthetic data corpus generation' (Yannikos et al., 2014), and desirably 'realistic data' (Garfinkel et al., 2009). According to the taxonomy by Breitinger and Jotterand (2023), this would be: 'Synthetic → Computer Simulated → Scenario.'

From a conceptual perspective, Garfinkel et al. (2009) present the aspect of realism in synthetic forensic data as "similar to what a forensic investigator might encounter in an investigation, but the data set was in fact artificially constructed." They describe this as involving a clean operating system installation, running programs, and performing basic operations or even 'sophisticated role play.' Moch and Freiling (2009) echo this by critiquing the use of secondhand disks for teaching, arguing that they are "not always very interesting and not typical for the cases that students will encounter later in their professional life."

Göbel et al. (2023) provide a further discussion of realism in digital forensic datasets. Building on Horsman and Lyle (2021), they discuss three use cases for datasets with requirements specified for each: tool testing and validation, training and education, and forensic research (adding a fourth use case of machine learning). For the education and training use case, they suggest: a broad spectrum of data needed (simple to complex), a ground truth may be required, and randomization per user may be necessary to avoid plagiarism in some circumstances.

Further comments by Göbel et al. (2023) on the realism of synthetic data created with a data synthesis framework highlight the important aspect that "it utilizes system functionality and thus represents reality." Therefore, they propose a different question: "How does the synthetic data set differ from a similar real-world data set?" To allow for such a comparison, they identify the properties of shareability, ground truth availability, updateability, determinism, background noise, regular 'wear and tear,' timeliness, and predictability.

This approach of assessing realism through a series of properties marks an important step forward from previous high-level descriptions of 'realistic.' We explore this further in Sections 4 and 8. However, the crux of this section's treatment of realism is the need for synthetic datasets to closely resemble real-world data, particularly in terms of a defined subset of properties.

2.2. VM control automation for forensic data synthesis

One of the properties previously identified for realism is background noise or 'wear and tear,' which much of the previous work on the automated generation of disk images has cited as a motivation. On account of the time-consuming nature of creating datasets that contain regular use of a system in addition to actions that generate evidential artifacts, existing work has primarily attempted to automate user actions to generate bulk background activity.

Du et al. (2021) categorize actions into external machine control and user actions to be performed inside the machine. They also describe that the "gold standard for this work is full external control of the guest operating system from the host, resulting in no internal artifacts that relate to the automation." Although their work could not achieve that, this provides another property that may be used to assess whether a disk image is 'realistic.'

Other work also includes an agent within the VM, with the ForTrace framework by Göbel et al. (2022) providing an even broader set of

[1] https://github.com/lenavoigt/re-imagen.

actions that can be performed compared to earlier work but again facing the problem of automation framework traces within the data created. However, they seek to mitigate this issue retrospectively through the use of 'antiforensic capabilities,' e.g., deleting Registry keys, such as UserAssist, or other Windows artifacts, such as Prefetch or Thumbcache files, thereby further supporting the desired lack of traces from the automation framework.

Recent work by Schmidt et al. (2023) has overcome this 'intrusiveness' problem. It thus provides one of the main building blocks for automatically creating synthetic disk images that do not contain traces from the automation framework used. They present *pyautoqemu*, which uses a combination of OpenCV-based computer vision to allow GUI elements to be identified and communication with the QEMU hypervisor to virtualize input devices that can be controlled externally. In particular, their tool allows for sending keyboard input and interacting with GUI elements via mouse clicks. It locates these elements with OpenCV through the use of visual templates – essentially images of specific GUI areas of interest. The work provides an implementation that exists entirely outside the guest virtual environment, achieving the sought-after goal of not creating automation artifacts within the disk image. It also provides a platform-independent solution since it is based on sending keyboard input or finding and interacting with defined graphical templates within the display of the guest VM, which can be defined for any guest operating system. While there are limitations to this framework, acknowledged by the authors and discussed in this paper in more detail later in Section 8, this is a fundamental component for achieving the goal of realistic, scenario-based disk image synthesis.

Another promising approach is the enhanced version of ForTrace presented by Wolf et al. (2024). The authors state that they could limit traces of the automation framework, re-designing ForTrace by removing the client-side agent component. They also implemented Optical Character Recognition (OCR) to handle tasks such as dealing with cookie banners, as well as image similarity measurements to recognize both expected and unexpected events. However, at the time of writing, the code was not yet available. That is why, in the following, we concentrate on the tool presented by Schmidt et al. (2023).

With this new tool addressing some of the main issues of VM control automation, exploring other disk image synthesis challenges is now possible.

2.3. *Challenges of scenario-based disk image generation*

With the availability of non-intrusive external VM control, it is possible to consider other hurdles in achieving realism. First, it is necessary to understand the overall process of generating a scenario-based disk image.

Hargreaves (2017) describe the stages of producing such a disk image as: constructing the scenario, storyboarding, and simulating the user's actions, which includes not just actions relating to the 'crime,' but also background activity. This high-level overview can then be broken down into more detail to consider specific challenges. Du et al. (2021) summarize the process of generating disk images as: first, designing the scenario and planning which artifact types are required; second, ensuring the availability of trained personnel to perform the activities over the specified time period; third, conducting the technical implementation of creating the VM, installing the operating system, adding users, etc.; fourth, executing the scripted set of actions; and ultimately, maintaining and indexing the corpus of synthetic disk images. In addition, Yannikos et al. (2014) specifically mention the part of the process that involves "a real-world scenario in which the required kind of data is typically created."

Besides the highlighted need for background activity, it is essential for storyboards in these scenarios to maintain narrative coherence. For example, Moch and Freiling (2012) report that students could identify inconsistencies, such as the use of Ubuntu Linux before Ubuntu was first released. Consequently, it is crucial not only for evidential actions but also for background activity to maintain a coherent narrative. One approach to achieve this is considering the persona of the simulated user, as much of the background activity would likely relate to their interests.

Taking the aforementioned considerations into account, an overview of an action-driven disk image generation process can be formulated as.

- outline the overall scenario,
- develop personas for the individual(s) involved,
- create a storyboard for both relevant actions and non-relevant actions,
- prepare the content of any media to be used in the scenario (email content, documents written, external files copied, etc.),
- carry out user actions, documenting the exact processes carried out to preserve the ground truth,
- retrieve the final disk image.

Considering the process outlined above, despite advancements in external VM control, numerous complex and time-consuming previous stages are still required to achieve narrative coherence.

Beyond the latest advancements in generative AI, especially with LLMs like GPT-4, which can process both image and text inputs to produce text outputs (OpenAI, 2023), there is prior research on AI tools being used for creative writing tasks (Elkins and Chun, 2020). Furthermore, Scanlon et al. (2023) propose using ChatGPT to assist with digital forensic scenario generation and provide examples for storyboard generation for an intellectual property theft scenario. Their approach involves generating detailed sets of actions, and creating 'character profiles' for people involved in the scenario as examples, but did not consider linking this with automation. There is, therefore, an existing example that shows leveraging the creative writing capabilities of LLMs for synthetic forensic data generation is possible, albeit without full investigation into the feasibility and effectiveness of this approach.

3. Methodology

Given that there is a proposal for the use of ChatGPT for scenario, storyboard, and persona generation (Scanlon et al., 2023), as well as a non-intrusive framework for VM control automation (Schmidt et al., 2023), this prompts research into whether an end-to-end, AI-assisted, scenario-based disk image synthesis is possible.

In this paper, we address this question by designing and building a proof-of-concept prototype. Our work has three stages, outlined below and further described in Sections 4, 5, and 6.

- **Requirements Analysis**: First, we collected requirements for the system design. However, we did not conduct stakeholder surveys or interviews due to the limited availability of such individuals. Instead, we considered requirements from existing literature that attempted to automate the creation of synthetic disk images and focused on those specifically related to the goals of this paper.
- **System Design and Prototype Implementation**: We designed the overall architecture of Re-imagen, a framework that meets the requirements of the previously conducted requirements analysis. We then developed prototype implementations of the core components of the architecture, again considering the extent to which they meet the derived requirements.
- **Demonstration**: In this final stage, we demonstrate the end-to-end process from scenario generation through to the final synthetic disk image.

4. Requirements

Much of the existing work concerned with synthetic disk image generation articulates the requirements for a corresponding synthesis system. Many of these requirements were first articulated by Moch and

Freiling (2009) and Moch and Freiling (2012), including the need to provide a complete installation of an operating system, offering extendable frameworks, and separating activity generation from activity execution. The need to generate background activity or 'noise' is discussed in Scanlon et al. (2017); Du et al. (2021); Göbel et al. (2022); and Schmidt et al. (2023). Additionally, Du et al. (2021) outline the need to mitigate traces of the automation framework within the disk image, also discussed by Göbel et al. (2022) and ultimately achieved by Schmidt et al. (2023).

While previous work has explored many requirements, neither the nature of the actions to be carried out nor the actions' coherence with the scenario has received much attention. It may be assumed that scenario-related actions inherently align with the scenario, but carrying out *background* activity that is consistent with the scenario and the fictitious user of the system has not been to be discussed in any of the literature reviewed.

Considering this deficiency and the present paper's proposal, additional requirements emerge that are specific to a set of actions generated by an LLM, such as the need to allow a human review of actions before execution, in accordance with responsible AI use through human-in-the-loop systems (Dignum, 2019). As the present research focuses on addressing a previously unconsidered part of the realism challenge, the requirements are specified as follows.

R1 The system shall generate user personas based on prompts.
R2 Generated activity schedules must be consistent with the usage pattern of the generated persona.
R3 Generated activity schedules must contain activity consistent with the generated persona.
R4 Generated activity schedules must be output in human-readable format for review.
R5 Generated activity schedules must be translated into machine-readable scripts for execution by the VM control automation.
R6 The output of the system will be a disk image containing traces of the activities carried out.

5. System design

In this section, we elaborate on our proposed overall system architecture for the Re-imagen framework. The workflow allows us to create a structured description for the use of a computer by a fictitious user with an LLM, translate this description to instructions for VM control automation, and ultimately retrieve synthetic forensic data.

Re-imagen consists of three core components: the Prompt Helper Module, the Translation Module, and the VM Instruction Module. The primary means of communication between the components are the Activity Description Script and VM Interaction Script. We refer to the employed VM control automation tool interacting with the VM for synthetic data creation as the Automatable VM Module. In this context, it is important to note that while our prototype implementation utilizes the GUI interaction tool by Schmidt et al. (2023), the conceptual system design allows for the analogous use of different tools, such as ForTrace (Göbel et al., 2022; Wolf et al., 2024). Although our focus is on disk image synthesis, the retrieval of further forensic data, like main memory dumps or network traffic, is feasible as well.

Fig. 1 summarizes the entire workflow of Re-imagen. Note that it omits the manual preparatory actions performed by the instructor, such as the initial configuration of the VM in accordance with the scenario as well as any further preparations required by the chosen automation tool.

The next sub-sections describe the core components we developed and the system's means of communication.

5.1. Prompt Helper Module

The Prompt Helper module supports the interaction between the instructor and the LLM to yield a coherent result. It ensures that the LLM ultimately outputs a suitable Activity Description Script (see Section 5.2). The obligatory steps in this action-generation process are the creation of.

1. A persona for the user,
2. A schedule of this persona for a specified time period,
3. A corresponding Activity Description Script for a selected device.

All this information can be generated during the LLM interaction, where the instructor can either have the LLM create the requisite details or provide partial information to direct the LLM. Interacting with the LLM, a persona that describes the user of a system is first created, e.g., their full name, age, gender, location, language, interests and hobbies, occupation, and technological proficiency. Second, a schedule for this persona is generated for a particular time period (e.g., for a specific day) that contains the usage of a device, such as a private computer. This overall schedule helps to situate the usage of the device for which synthetic forensic data is generated into the persona's overall daily routine. Therefore, it could be helpful when creating consistent forensic data across multiple devices or personas. However, in this paper, we focus on generating forensic data for a single persona and device. Ultimately, the LLM outputs an Activity Description Script for a selected device in accordance with the persona's overall schedule.

It should be noted that in each step, the Prompt Helper Module provides the option for the instructor to customize the queries to the LLM. For example, in the persona creation stage, the instructor is able to

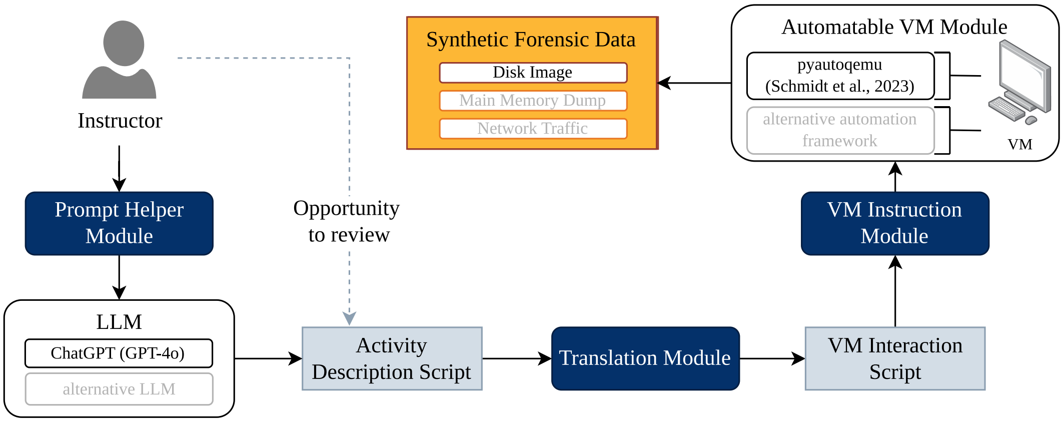


Fig. 1. Overall workflow of Re-imagen describing: System modules (dark blue), communication files between different components (light blue), integrated tools, including LLMs and VM control automation components (white), and final output (orange). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

define the desired characteristics of the persona instead of purely relying on all characteristics generated by the LLM. Moreover, it allows for the incorporation of arbitrary, concrete activities into the time schedule as specified by the instructor.

5.2. *Activity Description Script*

The Activity Description Script is the final output generated during LLM interaction. It contains the activities and corresponding timestamps related to a persona's usage of a specified device in a structured format. However, before further processing, the instructor may review and adjust the script, for example to check for any undesirable actions that have been generated, or to review for biased assumptions or stereotypes that have emerged from the LLM. The Activity Description Script is a JSON file that only uses absolute time values for readability. Furthermore, it abstracts from concrete, system-specific implementations of activities, e.g., it would only contain the activity of shutting down the computer at a particular time. It would not include the concrete steps that need to be performed as they can vary considerably depending on the system, e.g., the GUI elements to be interacted with.

5.3. *Translation Module*

The Translation Module converts the reviewed LLM output (the Activity Description Script) to a VM Interaction Script (see Section 5.4) suitable for the desired system. To do so, it accesses the system specifications including operating system, number of users, user passwords, or programs installed. Moreover, to map the activities of an Activity Description Script to instructions of a VM Interaction Script, it must be aware of the concrete commands available to the VM Instruction Module for the specific system.

User activities might be mapped to multiple instructions for the VM Instruction Module. This is the case when the activities translate to successive VM instructions that are carried out immediately after each other. It can also occur if multiple concrete implementations of an activity are possible on the system, or if the way the activity is performed depends on its temporal placement relative to other activities.

Furthermore, the Translation Module performs some basic checks on the validity of the Activity Description Script, such as timestamps being in chronological order and computer-on and computer-off events encapsulating other types of events.

5.4. *VM Interaction Script*

In contrast to the Activity Description Script, the VM Interaction Script contains times, commands, and all required arguments for the execution of activities on the VM. Its focus is not on readability but on rapid processing and smaller file sizes. For this format, a CSV file is used, in which either absolute times or relative times with a start time are specified.

The purpose of the VM Interaction Script is to provide a level of abstraction between general activity descriptions and the actual implementation of the VM control automation. This allows using different automation tools during the actual generation of forensic data. Instead of providing concrete commands for the automation tool, the VM Interaction Script includes references to the activity implementations for a specific setup, for instance, a single-user Windows 10 Home computer with Firefox as the default web browser. These activity implementations are supplied within a separate activity repository in the VM Instruction Module (see Section 5.5) containing detailed implementations compatible with the selected VM control automation tool.

Upon choosing the GUI interaction tool *pyautoqemu* by Schmidt et al. (2023), which enables scripting fine-grained actions like keyboard inputs and mouse clicks, the exact steps for completing a task, such as performing a Google search in Firefox, are stored in the activity repository rather than the VM Interaction Script. Should another VM control automation tool be selected, we merely need add implementations of the abstract activities for that automation tool to the activity repository.

5.5. *VM Instruction Module*

The VM Instruction Module controls the VM using a VM control automation tool (such as *pyautoqemu* by Schmidt et al. (2023)), the VM Interaction Script produced by the Translation Module, and an activity repository for the chosen interaction tool and specific system. By executing the concrete system-specific commands with the details supplied in this script, the VM Instruction Module interacts with the Automatable VM Module to create activity on the VM. Furthermore, the VM Instruction Module is responsible for logging the performed activities. It logs the actual activity initiation time to address possible deviations from the VM Interaction Script. This is required primarily for two reasons: First, it is not possible for the LLM to predict the concrete time needed to perform an activity on the specific system. Second, if an activity from the Activity Description Script is translated to successive instructions, they all hold the same timestamp in the VM Interaction Script, making an additional logging mechanism necessary for ground truth provision. The VM Instruction Module also creates a screenshot of what the VM displays after an activity is completed to allow for further validation of the intended actions taking place.

When the script is complete, the VM Instruction Module also directs the retrieval of the final output (the synthetic forensic data) from the Automatable VM Module offering different output formats, such as E01 or raw disk image files.

6. Demonstration

In this section, we demonstrate the functionality of the Re-imagen framework based on two examples using a prototype implementation.

6.1. *Prototype implementation*

Our prototype implementation uses ChatGPT with GPT-4o as the LLM and *pyautoqemu* (Schmidt et al., 2023) as the VM control automation tool. Within the developed system, we implemented the Translation Module as well as the VM Instruction Module in Python 3.11, the latter working purely with absolute time values. For each entry in the VM Interaction Script, the VM Instruction Module logs the time that the activity was initiated and creates a screenshot of the VM display. A more fine-grained logging mechanism is also conceivable. For the time being, the Prompt Helper module has been realized as query templates that support the web interface interaction with ChatGPT and are the result of multiple iterations of prompt engineering.

The prototype targets a single-user Windows 10 Home system with Firefox 126.0.1 as the default browser. The GUI-based interactions that are implemented are for turning the computer on and off, creating text documents with Notepad, conducting Google searches, and subsequently visiting one to three of the search results.

The Translation Module was implemented as follows: For the activity of turning the computer on, the Translation Module added two sequential activities to the VM Interaction Script, namely turning the computer on and logging in a single user with a provided password. For turning the computer off, we implemented a sequence of corresponding mouse–click interactions with the Windows menu. For the Google search sessions, we implemented a conditional translation of activities. The initial query of a session must open the browser, after which the query can be entered without any further interaction with the GUI. For all subsequent searches, we randomly chose between entering the new search term in the same tab as the previous one or opening a new tab.

Since the results of Google searches are dynamic, it is not possible to create a visual template of the search results for interaction. To implement browsing to these search results, we enhanced the GUI interaction tool by Schmidt et al. (2023) with basic OCR capabilities using Python Tesseract (version 0.3.10). This allowed for OCR either on the entire VM screen or on specified sections of the screen only.

In the following subsections, we provide two examples that cover all necessary steps from LLM interaction to retrieving a synthetic disk image. We have set up a dedicated Ubuntu 22.04 LTS machine to run our experiments on. The first example involves the use of a personal computer over three days, including activities such as turning the computer on and off, as well as using Firefox for Google searches. In the second example, we define a time frame on a specified day during which the LLM should schedule activities. We use this example to illustrate the currently implemented activities of our prototype system, which include browsing to Google search results and creating text documents. Both resulting disk images were analyzed with Autopsy (version 4.21.0). Indented text in both examples indicates inputs to and outputs from ChatGPT, with the latter being emphasized.

6.2. *Example 1 - Maximilian*

In our first example, we requested a persona for

> Maximilian, a 32-year-old accountant living in Berlin, Germany, who is a native English speaker.

For persona creation, we provided the first name, age, location, and occupation to ChatGPT in our query. The result included additional information such as that

> *he balances his professional life with a variety of interests, including photography, cooking, reading, and cycling.*

Subsequently, we had ChatGPT generate an overall schedule for this persona and an Activity Description Script for turning the computer on and off, covering the period from Friday, May 31, 2024, to Sunday, June 2, 2024. We then requested an enrichment of the resulting script with

> extensive Google search activities considering the persona [...] and the influence the different days of the week and holidays have on the behavior of this user.

The ChatGPT-generated Activity Description Script for the three days included 33 search queries across nine computer sessions. The search terms in the script were consistent with the persona, as they referred to topics such as photography or cycling route research (see Table 1). The different days of the week seemed to have been considered in generating activity, as there was no activity on the private computer during typical work hours on Friday. An excerpt of the script is shown in Fig. 2.

```
[{"time":"2024-05-31T16:20:30+02:00",
    "activity":"computer_on"},
 {"time":"2024-05-31T16:25:10+02:00",
    "activity":"google_search",
    "search_term":"photography competitions Berlin 2024"},
 {"time":"2024-05-31T16:29:50+02:00",
    "activity":"google_search",
    "search_term":"local photography exhibitions"},
 ...
 {"time":"2024-05-31T16:53:30+02:00",
    "activity":"computer_off"}]
```

Fig. 2. Extract from an Activity Description Script for ‘Maximilian’, generated with GPT-4o.

```
2024-05-31T16:20:30+02:00,start_computer
2024-05-31T16:20:30+02:00,login_single_user,12345678
2024-05-31T16:25:10+02:00,
    firefox_simple_initial_google_search_session,
    photography competitions Berlin 2024
2024-05-31T16:29:50+02:00,
    firefox_open_new_tab_and_search,
    local photography exhibitions
[...]
2024-05-31T16:53:30+02:00,shutdown_via_menu
```

Fig. 3. Extract from the VM Interaction Script that corresponds to the Activity Description Script in Fig. 2.

To be able to conduct a complete test run, we set up a Windows 10 Home VM with one user, named *Max*, with the password *12345678*. We then translated the Activity Description Script to a VM Interaction Script (see Fig. 3) with the prototype of the Translation Module and supplied the latter script to the VM Instruction Module. The VM Instruction Module ran between 7:10 on May 31, 2024, and 18:10 on June 2, 2024, and executed the given activities contained in the VM Interaction Script. The results were validated based on the automatically captured screenshots and the generated log file. Finally, we analyzed the exported disk image using Autopsy and were able to find the resulting digital forensic traces related to the search terms.

In Appendix A, we provide an excerpt from the ChatGPT response for persona generation.

6.3. *Example 2 - Catherine*

In the second example, we requested ChatGPT to create a persona for

> Catherine Smith, a 43-year-old chef living in Vienna, Austria, who is a native English speaker.

Table 1
Excerpts from the ChatGPT-generated personas and selected corresponding Google search queries.

Persona	Google Searches
Full Name: Maximilian Hunter; *Gender*: Male; *Age*: 32; *Location*: Berlin, Germany; *Language*: English (native), German (proficient); *Occupation*: Accountant; *Hobbies & Interests*: Photography - often explores Berlin to capture its urban beauty, cooking and experimenting with fusion recipes combining German and British cuisine, reading with a preference for historical fiction and mystery novels, participates in local cycling clubs, enjoys weekend bike tours, tech enthusiast	photography competitions Berlin 2024; how to prepare for a photo competition; best DSLR settings for competition; Berlin outdoor activities June 2024; cycling routes near me; Sunday breakfast recipes
Full Name: Catherine Smith; *Gender*: Female; *Age*: 43; *Location*: Vienna, Austria; Language: English (native); *Occupation*: Chef; *Hobbies & Interests*: Culinary arts and experimenting with new recipes, wine tasting and pairing, attending and hosting cooking workshops, exploring local markets and sourcing fresh ingredients, traveling to discover new cuisines, reading culinary magazines and food blogs, gardening and growing herbs, yoga and mindfulness practices	latest culinary trends 2024; summer vegetable recipes; food blogs to follow 2024; wine recommendations for dinner parties; how to organize a cooking workshop; promote cooking workshop online

The resulting, ChatGPT-generated persona included further characteristics, such as an interest in

> *culinary arts and experimenting with new recipes, wine tasting and pairing, as well as attending and hosting cooking workshops.*

For this example, we asked ChatGPT to generate a schedule for June 1, 2024, and mentioned that the persona

> only uses her computer after work, between 22:52 and 23:58.

Subsequently, we requested the generation of an Activity Description Script, also including

> the activity of the user creating text documents on their system[, which] could be any type of text, for example, a diary, or an email draft.

The resulting script encompassed nine Google searches and the creation of two text documents. These activities aligned with the specified persona, involving searches related to cooking, wine recommendations, and organizing a workshop, as detailed in Table 1. Notably, the creation of a text document titled *recipe_notes* followed two searches on cooking. Additionally, the creation of a text document titled *workshop_outline*, which contained a seven-step agenda with descriptions for each point, followed three Google searches concerning workshop organization.

Analogous to the previous example (see Section 6.2), we set up a VM with one user, named *Catherine*. The VM Instruction Module ran between 22:52 and 23:58 on June 1, 2024, executed the Google searches with consecutive browsing to search results, and created the text documents.

Appendix B contains excerpts from the ChatGPT responses for persona, daily schedule, and Activity Description Script generation, as well as an extract of the translated VM Interaction script. It also includes Firefox history artifacts, identified on the disk image by Autopsy, showing persona-consistent background activity generated between 22:52 and 23:58 on June 1, 2024.

7. Evaluation

Revisiting the requirements described in Section 4, this section discusses the extent to which the work carried out addresses those requirements.

> R1: The system shall generate user personas based on prompts

Section 6 has shown the details of providing a prompt to ChatGPT using a helper template that ultimately results in profiles containing full name, age, gender, location, language, interests and hobbies, occupation, and technological proficiency. The LLM adds many additional details that enrich the profile. These details do not need to be created manually by the instructor constructing the scenario.

While the two examples provided are a limited set, the diversity of generated persona characteristics is quite broad suggesting many more available options and variety possible. In this paper, relationships with friends and family members were not considered, which was discussed by Scanlon et al. (2023). However, this would require only a minor update to the Prompt Helper Module now that the overall concept has been demonstrated.

> R2: Generated activity schedules must be consistent with the usage pattern of the generated persona

Section 6 has shown that for the individual personas, the activity generated by ChatGPT was within the qualitatively described time bounds. In the first example, this was relatively broad, merely specifying that the computer was used on three particular days. This resulted in multiple computer sessions per day, selected by ChatGPT, with activities encapsulated by a computer on and computer off event. In the second example, the time frame during which the persona should use the computer was mentioned explicitly.

> R3: Generated activity schedules must contain activity consistent with the generated persona

Section 6 has shown that the web browsing is consistent with the personas generated, as are the text files created, which is one of the major contributions of this work. Selected Google searches conducted are summarized in Table 1 to highlight the differences in activity generated in each of the individual cases. The searches of the two users show clear links between their personas, occupations, locations, and interests.

This research has used only Google searches and text document creation at this stage to demonstrate this coherence and consistency, but other activities can be incorporated in the future, allowing more extensive use of LLMs for content generation to be tested. In turn, more generative activities will also be included, for example, creating communication via emails. But generating this type of content falls well within the capabilities of ChatGPT.

The limited support for different activity types is in part due to limitations of the GUI automation tool, which at the time of writing was missing features, such as identifying whether an action was performed successfully or handling unexpected system behavior.

> R4: Generated activity schedules must be output in human-readable format for review

Since the activities are generated by an LLM, there is a risk that undesirable activities may be produced. This requirement was designed to separate script generation from execution and to allow manual review. An extract of the generated human-readable set of actions is shown in Fig. 2. Given the simple JSON format, it is easy to identify whether the actions generated are undesirable, and, if so, to remove or modify them. However, when large sets of actions are generated, a manual review may become cumbersome. This could prompt automated checking to flag undesirable activity, e.g., based on keywords, in addition to manual review.

> R5: Generated activity schedules must be translated into machine-readable scripts for execution by the GUI automation

The use of the VM control automation meant that a Translation Module was needed to convert the LLM-generated actions into scripts. These were then interpreted for execution by, in this case, *pyautoqemu*. The modular nature of our proposed system architecture means that it is possible to use an alternative automation framework if another is preferred. An example VM Interaction Script is shown in Fig. 3 and represents the machine-readable, system-specific, enriched version of the human-readable, more generic Activity Description Script shown in Fig. 2 earlier.

> R6: The output of the system will be a disk image containing traces of the activities carried out

The overall system architecture requires minimal human prompting to create a persona, and then a series of actions aligned with that persona. These are automatically translated into VM control instructions which are executed by the framework. The resulting virtual hard disk is imaged using *ewfacquire* to create an E01 disk image with the generated activity. The Autopsy view of a resulting disk image for the 'Catherine' example is shown in Fig. 4 within Appendix B. It contains parts of the resulting Firefox history corresponding to the Google searches automatically generated and the browsing to search results. While more work is required to expand the range of activities, the prototype demonstrates the potential of our approach in terms of producing background activity in scenario-based synthetic disk images that have increased believability, richness, and coherence.

8. Limitations and further work

In this section, we reflect upon aspects of this work that require further attention. First, we discuss the endeavor to define and measure the realism of synthetic forensic data. Second, we elaborate on improvement possibilities for our prototype comprising a brief review of challenges in the use of ChatGPT and limitations of the VM control automation tool.

8.1. Definition and measurement of realism

The discussion of related work in Section 2.1 showed that the term ‘realistic data’ is still ill-defined. In this paper, we took a pragmatic approach and restricted our evaluation to the qualitative assessment of background activity coherence. Many properties of realism are currently expressed qualitatively, e.g., the time between actions being consistent with what is possible with real usage.

However, there are a number of quantitative metrics that have been used on synthetic disk images in previous work. Du et al. (2021) used Plaso to generate timelines of the generated disk images and considered the total events extracted and their variety (e.g. EVT, REG, WEBHIST). The number of files was also considered. However, it is unclear which values or distributions of values actually qualify a disk image to be called *realistic*.

To challenge the usefulness of these metrics, we computed several of them on a total of eight disk images that the authors had created over the past years for teaching. The results are shown in Table 2. The first column is a unique identifier derived from the name of the corresponding scenario. The type of operating system used is given as ‘type,’ ‘files’ refers to the number of files listed by fiwalk (Garfinkel, 2012), ‘events’ is the number of events listed by log2timeline (The Plaso (log2timeline) authors, 2024), and ‘timespan’ is an approximation of the lifetime of the system, specifically the days between earliest and latest `ctime` timestamp in fiwalk.

The table presents the typical short lifetime of exercise systems, with only two disk images (MTW and VC) showing timespans of considerable length: Disk image MTW was based on a disk image produced by an instrumented VM that had performed random application actions within a daily schedule for considerable time. Although this image was taken from a different project and not originally intended to be used for teaching purposes, it provided a suitable starting point for a manually executed story. Disk image VC also comprised a timespan of more than 200 days. However, in the fiwalk output, we also observed timestamp anomalies with multiple timestamps dating between 1970–1984 and 2046–2094 (which were ruled out for timespan calculation). These anomalies may be due to the use of ForTrace (Göbel et al., 2022) for synthesizing this disk image. Apart from MTW and VC, all disk images were created manually.

Examining the number of files and events, we cannot observe a useful pattern distinguishing manually-created images from automatically-created ones. It is also unclear how these quantitative measures behave over the lifetime of a system: Does more equate to better? Is there ‘too much’ to be normal?

While Du et al. (2021) used Plaso and Procmon for metrics, we concur with the authors that quantitative metrics of realism are a significant challenge as they state that “further work is also required in this area and an automated way of comparing the traces left by a real-life action against the machine-generated set [is needed].”

Exploring the possibilities of more qualitative metrics, one could consider measuring not individual events but rather higher-level *activities* like those included in the Activity Description Script of our architecture. One could, for example, measure the total number of activities, their variety, the time between them (whether they are representative of real-world disk images), their longevity, and semantic coherence. An potential but naturally fuzzy metric, for which we admittedly have no data, is the number of human work hours spent in creating these disk images.

While these metrics may allow some aspects of realism to be assessed, quantitative metrics are currently a challenge for the benefits this paper offers, such as the richness of personas and coherent activities. We conjecture that no single set of quantitative metrics will be able to capture semantic plausibility as we have tried to create with LLMs.

Capturing semantic consistency and realism will likely involve approaches from different domains, such as creative writing where users often rate collaborative fiction in terms of ‘coherence’ on a 5-point scale at the end of a generated story (Swanson and Gordon, 2008).

8.2. Prototype improvements

While this paper has demonstrated the potential of our approach to enhance scenario-based disk image synthesis by emulating user behavior based on and integrated with LLM-generated actions and content, there is still significant work required in various areas.

The first area is enhancements to the prototype of our Re-imagen framework. Aside from the support of a broader range of activities and system types, these involve a more expressive intermediate language for VM Interaction Scripts. Such a language could allow nesting, grouping, expressing more complex actions like ‘browse the web for *x* amount of time,’ or introducing dependencies. Enhancements to the Prompt Helper Module could entail using a wrapper for the ChatGPT API instead of templates for interaction with the web interface, or letting the instructor influence concrete activities more flexibly.

Moreover, it would be useful to support generating longer-running images incrementally. At the same time, functionality for consistency checking or flagging undesired activities should be prepared since human oversight will become increasingly difficult with longer simulation timespans.

Another area needing improvement is the GUI tool we used, which is acknowledged by its authors (Schmidt et al., 2023). Examples are the

Table 2
Comparison of metrics from manually created educational disk images: ‘files’ refers to the number of files listed by fiwalk, ‘events’ is the number of events listed by log2timeline, ‘timespan’ is measured in days between earliest and latest `ctime` timestamp in fiwalk.

ID	type	created	files	Events	timespan	comment
WW	Linux	2016	314104	975221	12	
KK	Linux	2017	334020	950417	2	
MTW	Windows	2020	639975	2808016	609	based on a long-running image with regularly executed random actions
MTL	Linux	2020	521547	1154879	16	
NM	Windows	2021	552883	2286776	31	
VC	Windows	2022	84610	1193988	228	built using ForTrace (Göbel et al., 2022)
ASK	Linux	2023	271451	951413	9	
ASB	Windows	2023	728713	2787882	11	

initial manual effort needed to provide templates for every action or the current lack of capabilities to recognize the success of an action or handle unexpected events, such as popups. However, the OCR capabilities we implemented to enhance the tool might be a starting point for addressing some of these challenges, such as interaction with cookie banners. Alternatively, it may be considered to evaluate and switch to the agentless version of ForTrace proposed by Wolf et al. (2024), once its code becomes available.

During our experiments, we also encountered inherent and considerable limitations to automation frameworks that may be practically unsolvable, namely reverse Turing tests (such as CAPTCHAs) occasionally employed on websites like Google and the challenge of including social media usage without conflicting with rules prohibiting the deployment of bots.

Previous work on utilizing LLMs, such as ChatGPT, to facilitate digital forensic tasks has highlighted the issue of hallucinations (Scanlon et al., 2023; Henseler and van Beek, 2023; Michelet and Breitinger, 2024). In light of this issue, we aimed to design our system with the current strengths and limitations of LLMs such as GPT-4o in mind, utilizing it to assist with tasks related to creative writing. Other parts of the process, such as the incorporation of system specifics that need to be considered to implement the actions on a system, are performed outside of the LLM interaction phase. In addition, the instructor can review LLM output before it is used for data synthesis. Moreover, we decided to implement web browsing sessions through Google searches instead of having the LLM directly generate URLs the user browses to, as those are prone to be invalid due to such hallucinations. However, this issue needs to be considered when aiming to enhance the activity descriptions generated with an LLM.

9. Conclusion

The availability of digital forensic datasets, especially scenario-based ones, remains relatively limited. However, there has been a growing effort to enhance the automated generation of synthetic disk images, with arguments highlighting their significance in numerous use cases (Göbel et al., 2023). Considering properties such as coherent background activity, long-term usage of a system, and volume and frequency of general activity, is crucial to generate synthetic forensic data that is representative of real-world investigations. The use of LLMs to create personas and corresponding background activity at scale is a potential way to meet these properties.

Our work has implemented and demonstrated a system capable of addressing these needs, arguably increasing the realism of synthetic disk images for those who require these properties within their automatically generated datasets. While there is still room for improvement, we believe our proposed approach represents a significant advancement over current methods that either lack automatic background activity generation or produce random, bulk activity. The potential for future work is extensive. For example, it could involve using the Re-imagen framework to run multiple VMs for various personas over extended periods. This could be used to create a repository of rich base images that can serve as a starting point for digital forensics teaching scenarios. Therefore, our approach has the potential to significantly address the persistent dataset availability problem in the digital forensics field.

CRediT authorship contribution statement

Lena L. Voigt: Conceptualization, Investigation, Methodology, Software, Validation, Writing - Original Draft, Writing - Review and Editing. **Felix Freiling:** Conceptualization, Investigation, Methodology, Writing - Original Draft, Writing - Review and Editing, Supervision. **Christopher J. Hargreaves:** Conceptualization, Investigation, Methodology, Writing - Original Draft, Writing - Review and Editing, Supervision.

Acknowledgments

This work has been supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Research and Training Group 2475 “Cybercrime and Forensic Computing” (grant number 393 541 319/GRK2475/2-2024).

A. Example 1 - Maximilian

A.1 ChatGPT-generated Persona Description

Full Name: Maximilian Hunter; Gender: Male; Age: 32; Location: Berlin, Germany; Language: Native English speaker, proficient in German; Occupation: Accountant. […]

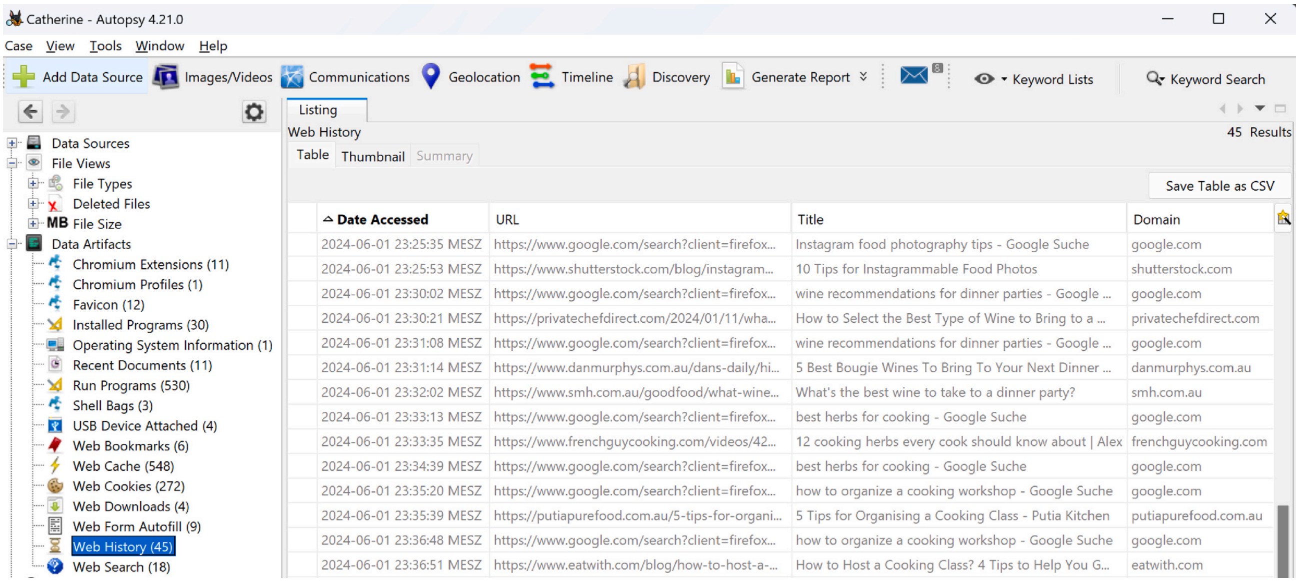
Summary: Maximilian Hunter is a 32-year-old accountant living in Berlin, Germany. Originally from the UK, he is a native English speaker and proficient in German. With a background in accounting and finance, he balances his professional life with a variety of interests, including photography, cooking, reading, and cycling. Maximilian is tech-savvy, adept in using both accounting and photo editing software, and maintains an active online presence. His unique blend of financial expertise and creative pursuits make him a well-rounded individual with a diverse set of skills and interests.

B. Example 2 - Catherine

B.1. ChatGPT-generated Persona Description

Full Name: Catherine Smith; Gender: Female; Age: 43; Location: Vienna, Austria; Language: English (native); Occupation: Chef. […]

Summary: Catherine Smith is a 43-year-old chef residing in Vienna, Austria. Originally from an English-speaking country, she has brought her passion for culinary arts to her adopted home. With a background from Le Cordon Bleu and a penchant for wine, Catherine has carved out a niche for herself in the vibrant culinary scene of Vienna. Her interests extend beyond cooking; she enjoys traveling, gardening, and practicing yoga. Online, Catherine is an active member of culinary communities, always on the lookout for the latest trends and inspirations. She manages her digital life with a moderate level of IT proficiency, favoring straightforward and practical technology. Known for her innovative fusion dishes and commitment to fresh, local ingredients, Catherine continues to delight and inspire those around her with her culinary creations.



Web History 45 Results

Date Accessed	URL	Title	Domain
2024-06-01 23:25:35 MESZ	https://www.google.com/search?client=firefox...	Instagram food photography tips - Google Suche	google.com
2024-06-01 23:25:53 MESZ	https://www.shutterstock.com/blog/instagram...	10 Tips for Instagrammable Food Photos	shutterstock.com
2024-06-01 23:30:02 MESZ	https://www.google.com/search?client=firefox...	wine recommendations for dinner parties - Google ...	google.com
2024-06-01 23:30:21 MESZ	https://privatechefdirect.com/2024/01/11/wha...	How to Select the Best Type of Wine to Bring to a ...	privatechefdirect.com
2024-06-01 23:31:08 MESZ	https://www.google.com/search?client=firefox...	wine recommendations for dinner parties - Google ...	google.com
2024-06-01 23:31:14 MESZ	https://www.danmurphys.com.au/dans-daily/hi...	5 Best Bougie Wines To Bring To Your Next Dinner ...	danmurphys.com.au
2024-06-01 23:32:02 MESZ	https://www.smh.com.au/goodfood/what-wine...	What's the best wine to take to a dinner party?	smh.com.au
2024-06-01 23:33:13 MESZ	https://www.google.com/search?client=firefox...	best herbs for cooking - Google Suche	google.com
2024-06-01 23:33:35 MESZ	https://www.frenchguycooking.com/videos/42...	12 cooking herbs every cook should know about | Alex	frenchguycooking.com
2024-06-01 23:34:39 MESZ	https://www.google.com/search?client=firefox...	best herbs for cooking - Google Suche	google.com
2024-06-01 23:35:20 MESZ	https://www.google.com/search?client=firefox...	how to organize a cooking workshop - Google Suche	google.com
2024-06-01 23:35:39 MESZ	https://putiapurefood.com.au/5-tips-for-organi...	5 Tips for Organising a Cooking Class - Putia Kitchen	putiapurefood.com.au
2024-06-01 23:36:48 MESZ	https://www.google.com/search?client=firefox...	how to organize a cooking workshop - Google Suche	google.com
2024-06-01 23:36:51 MESZ	https://www.eatwith.com/blog/how-to-host-a-...	How to Host a Cooking Class? 4 Tips to Help You G...	eatwith.com

Fig. 4. The resulting disk image `catherine.E01` displayed in Autopsy, with persona-consistent background activity generated between 22:52 and 23:58 on June 1, 2024.

B.2. ChatGPT-generated Activity Description Script for June 2, 2024

```
[{"time":2024-06-01T22:52:48 + 02:00″,
    “activity":"computer_on"}, …
  {"time":"2024-06-01T23:35:14 + 02:00″,
    “activity":"google_search”,
    “search_term”: “how to organize a cooking
      workshop"}, …
  {"time":"2024-06-01T23:45:00 + 02:00″,
    “activity":"create_text_document”,
    “file_name":"workshop_outline”,
    “content":"Cooking Workshop Outline:
    \n1. Introduction and Welcome\n
      - Brief introduction of myself and my culinary
      background\n
      - Overview of the workshop agenda and objectives\n
      - Icebreaker activity to get participants
      engaged\n\n …
      7. Closing Remarks and Take-home Recipes\n … "},
  {"time":"2024-06-01T23:58:00 + 02:00″,
    “activity":"computer_off"}]
```

B.3. VM Interaction Script for June 1, 2024

```
2024-06-01T22:52:48 + 02:00,start_computer
2024-06-01T22:52:48 + 02:00,login_single_user,12345678
[…]
2024-06-01T23:35:14 + 02:00,
  firefox_open_new_tab_and_search,
  how to organize a cooking workshop
2024-06-01T23:45:00 + 02:00,notepad_create_document,
  workshop_outline,"Cooking Workshop Outline:
    1. Introduction and Welcome … "
2024-06-01T23:58:00 + 02:00,shutdown_via_menu
```

B.4. Artifact overview from autopsy

Fig. 4 presents the resulting disk image catherine. E01 displayed in Autopsy.

References

Breitinger, F., Jotterand, A., 2023. Sharing datasets for digital forensic: a novel taxonomy and legal concerns. Forensic Sci. Int.: Digit. Invest. 45, 301562.

Dignum, V., 2019. Responsible artificial intelligence - how to develop and use AI in a responsible way. Artificial intelligence: foundations, theory, and algorithms. Springer.

Du, X., Hargreaves, C., Sheppard, J., Scanlon, M., 2021. TraceGen: user activity emulation for digital forensic test image generation. Forensic Sci. Int.: Digit. Invest. 38, 301133.

Elkins, K., Chun, J., 2020. Can GPT-3 pass a writer's turing test? Journal of Cultural Analytics 5 (2).

Garfinkel, S., Farrell, P., Roussev, V., Dinolt, G., 2009. Bringing science to digital forensics with standardized forensic corpora. Digit. Invest. 6, 2–11.

Garfinkel, S.L., 2012. Digital forensics XML and the DFXML toolset. Digit. Invest. 8, 161–174.

Göbel, T., Baier, H., Breitinger, F., 2023. Data for digital forensics: why a discussion on 'how realistic is synthetic data' is dispensable. Digital Threats: Research and Practice 4 (3), 1–18.

Göbel, T., Maltan, S., Türr, J., Baier, H., Mann, F., 2022. ForTrace – a holistic forensic data set synthesis framework. Forensic Sci. Int.: Digit. Invest. 40, 301344.

Grajeda, C., Breitinger, F., Baggili, I., 2017. Availability of datasets for digital forensics – and what is missing. Digit. Invest. 22, 94–105.

Hargreaves, C., 2017. Digital forensics education: a new source of forensic evidence. Forensic Science Education and Training: A Tool-kit for Lecturers and Practitioner Trainers 73–85.

Henseler, H., van Beek, H., 2023. ChatGPT as a copilot for investigating digital evidence. In: LegalAIIA@ ICAIL, pp. 58–69.

Horsman, G., Lyle, J.R., 2021. Dataset construction challenges for digital forensics. Forensic Sci. Int.: Digit. Invest. 38, 301264.

Michelet, G., Breitinger, F., 2024. ChatGPT, Llama, can you write my report? an experiment on assisted digital forensics reports written using (local) large language models. Forensic Sci. Int.: Digit. Invest. 48, 301683.

Moch, C., Freiling, F.C., 2009. The forensic image generator generator (Forensig2). In: 2009 Fifth International Conference on IT Security Incident Management and IT Forensics. IEEE, pp. 78–93.

Moch, C., Freiling, F.C., 2012. Evaluating the forensic image generator generator. In: Digital Forensics and Cyber Crime: Third International ICST Conference, ICDF2C 2011, Dublin, Ireland, October 26-28, 2011, Revised Selected Papers 3. Springer, pp. 238–252.

OpenAI, 2023. GPT-4 technical report. https://cdn.openai.com/papers/gpt-4.pdf.

Scanlon, M., Breitinger, F., Hargreaves, C., Hilgert, J.N., Sheppard, J., 2023. ChatGPT for digital forensic investigation: the good, the bad, and the unknown. Forensic Sci. Int.: Digit. Invest. 46, 301609.

Scanlon, M., Du, X., Lillis, D., 2017. EviPlant: an efficient digital forensic challenge creation, manipulation and distribution solution. Digit. Invest. 20, 29–36.

Schmidt, L., Kortmann, S., Hupperich, T., 2023. Improving trace synthesis by utilizing computer vision for user action emulation. Forensic Sci. Int.: Digit. Invest. 45, 301557.

Swanson, R., Gordon, A.S., 2008. Say anything: a massively collaborative open domain story writing companion. In: Interactive Storytelling: First Joint International Conference on Interactive Digital Storytelling, ICIDS 2008 Erfurt, Germany, November 26-29, 2008 Proceedings 1. Springer, pp. 32–40.

The Plaso (log2timeline) authors, 2024. GitHub - log2timeline/plaso: super timeline all the things. https://github.com/log2timeline/plaso.

Wolf, D., Göbel, T., Baier, H., 2024. Hypervisor-based data synthesis: on its potential to tackle the curse of client-side agent remnants in forensic image generation. Forensic Sci. Int.: Digit. Invest. 48, 301690.

Yannikos, Y., Graner, L., Steinebach, M., Winter, C., 2014. Data corpora for digital forensics education and research. In: Advances in Digital Forensics X: 10th IFIP WG 11.9 International Conference, Vienna, Austria, January 8-10, 2014, Revised Selected Papers 10. Springer, pp. 309–325.