Understanding Strategies and Challenges of Timestamp Tampering for Improved Digital Forensic Event Reconstruction

Céline Vanini
School of Criminal Justice
University of Lausanne
Lausanne, Switzerland
celine.vanini@unil.ch

Jan Gruber
Department of Computer Science
Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)
Erlangen, Germany
jan.gruber@fau.de

Christopher Hargreaves
Department of Computer Science
University of Oxford
Oxford, United Kingdom
christopher.hargreaves@cs.ox.ac.uk

Zinaida Benenson
Department of Computer Science
Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)
Erlangen, Germany
zinaida.benenson@fau.de

Felix Freiling
Department of Computer Science
Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)
Erlangen, Germany
felix.freiling@fau.de

Frank Breitinger*
Institute of Computer Science
University of Augsburg
Augsburg, Germany
frank.breitinger@uni-a.de

Abstract

Timestamps play a pivotal role in digital forensic event reconstruction, but due to their non-essential nature, tampering or manipulation of timestamps is possible by users in multiple ways, even on running systems. This has a significant effect on the reliability of the results from applying a timeline analysis as part of an investigation. We investigate the problem of users tampering with timestamps on a running ("live") system. While prior work has shown that digital evidence tampering is hard, we focus on the question of *why* this is so. By performing a qualitative user study with advanced university students, we derive factors that influence the reliability of successful tampering, such as the individual knowledge about temporal traces, and technical restrictions to change them. These insights help to assess the reliability of timestamps from individual artifacts that are used for event reconstruction and subsequently reduce the risk of misinterpretations.

CCS Concepts

• **Applied computing** → **Investigation techniques**; **Evidence collection, storage and analysis**; *System forensics.*

Keywords

Event reconstruction, Tampering, User study, Tamper resistance factors, Digital forensics investigation

ACM Reference Format:
Céline Vanini, Jan Gruber, Christopher Hargreaves, Zinaida Benenson, Felix Freiling, and Frank Breitinger. 2025. Understanding Strategies and Challenges of Timestamp Tampering for Improved Digital Forensic Event Reconstruction. In *Digital Forensics Doctoral Symposium (DFDS 2025), April 01, 2025, Brno, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3712716.3712727

1 Introduction

The dangers of evidence tampering, i.e., the intentional act of altering, concealing or falsifying evidence [21], are of great concern to law enforcement agencies since fictitious or fake traces can easily alter or sabotage a criminal investigation [4]. While there is a high awareness of the risks of tampering with physical evidence, tampering of digital evidence is much less understood. While some scholars consider tampering to be easier when dealing with evidence that is in digital form [1, 11], others claim that it is at least similarly difficult in special cases [22]. Understanding the risks of digital evidence tampering is particularly important for *timestamps* because, firstly, timestamps play a pivotal role in digital forensic event reconstruction to establish the order in which certain actions happened, and secondly, timestamp manipulation is a commonly applied indicator removal technique in security incidents [15, 23].

1.1 Related work

Despite work that has structured tampering activities under the heading of anti-forensics [5, 9, 10], and confirmed by the literature review of Neale [18], unfortunately, the understanding of digital evidence tampering in general and of timestamp tampering in particular is rather shallow. Previous research has primarily focused on specific technical contexts of timestamp tampering, such as manipulating file metadata on NTFS [8, 17, 19], or on technical approaches for timestamp tampering detection. These attempts aim to find inconsistencies between timestamps, e.g., the violation of general time rules [8], of causal relationships between timestamps [13, 27], or inconsistent relations to implicit timing information like sequence numbers [6]. Even if such inconsistencies are detected, it may still be unclear whether these are due to intentional tampering.

[*] The work was performed while being affiliated with the University of Lausanne.

DFDS 2025, Brno, Czech Republic

ACM ISBN 979-8-4007-1076-6/25/04
https://doi.org/10.1145/3712716.3712727

For example, if some timestamps have been set to January 1, 1970, explanations can vary from intentional "timestomping" [15] to an unintentional non-initialized Unix timestamp [12]. Thus, it is still necessary to have a solid understanding of adversarial (tampering) behaviors such that investigators can assess the effect of tampering on the meaning of evidence.

Some previous user studies [7, 16, 22] also focused on the tampering detection problem. In these experiments, participants had to produce convincing forgeries that should be taken as originals when analyzed by other participants in the study. This allowed to assess empirically how convincing forgeries were. Due to their empirical study design, these experiments did not attempt to investigate the difficulties of tampering with specific artifacts. Furthermore, the experiment setup considered the extreme case where a perpetrator has *full control* over every bit of the system, an approach we call *dead tampering*. Still, the quantitative insights from these works indicate that tampering may not be as easy as it can be expected, but it is highly unclear *why* this may be so.

In contrast to the worst case assumptions often made in the literature, in practice adversaries do not operate under idealized circumstances. When accessing a compromised system, perpetrators usually have less control because they are under time pressure, lack knowledge of alternative methods, or need to modify the system while logged in remotely. Also, while less experienced users may be able to perform actions such as changing a value in a database or editing some text in a file, they may not be capable of booting to an alternative environment and performing low-level manipulations.

In such situations, adversaries are forced to manipulate data on the system they are currently using, a process we call *live tampering*. Unlike dead tampering, live tampering is arguably not only more realistic than dead tampering (e.g., remote access scenario, or full volume encryption), it also introduces new challenges, as the act of tampering itself generates traces on the system being manipulated. When these traces are directly embedded in the manipulated data itself, we refer to them as *first-order traces*. For example, manipulating browser evidence tends to result in contradictory information in the browser history and browser cache [7]. In addition to first-order traces, tampering actions can also leave indirect indicators, which we call *second-order traces*, such as traces of anti-forensic tool usage.

1.2 A qualitative look at live tampering

In this paper, we report on the results of a user study in live tampering. While prior work has shown that digital evidence tampering is hard, we focus on the question of *why* this is so and therefore have chosen to apply qualitative research methods, i.e., questionnaires and semi-structured interviews. Our general goal was to understand how study participants chose their strategies and allocate their resources while solving a live timestamp tampering task. To investigate this, we conducted a user study with 10 advanced university students, who tampered with a live system based on a fictitious scenario, in which an adversary attempts to swap two events to cover their tracks. Not all adversaries are specialist hackers or advanced persistent threats (APT) in practice, so our protagonist was assumed to be a regular user rather than a sophisticated adversary. As we will show, the exploration of the dynamics of such tampering also leads to understanding the difficulty of tampering with specific artifacts. This can help develop further strategies for reliable event reconstruction, since methods for representing the uncertainty of traces, e.g., the C-Scale ('Strength of Evidence scale') [3], include an estimate of the number of sources that agree, but also the difficulty of tampering with those sources.

Our focus was on the following research questions:

RQ1 **Approach to tampering:** What strategies do adversaries employ in planning and executing tampering with the temporal order of events?

RQ2 **Awareness and precautions of traces left by the manipulation**: How do adversaries deal with (new) traces stemming from their manipulations?

RQ3 **Barriers to the tampering process:** What makes an artifact more difficult to tamper with compared to another?

Overall, this work provides the following contributions: (1) we present the design, implementation, and assessment of the first user study on tampering with timestamps on a running system (live tampering), (2) we provide clear indications that adversaries differentiate between first-order and second-order traces and adapt their tampering strategy accordingly, (3) we identify strategies of live tampering that involve the opportunistic application of tampering actions along the hierarchical abstraction layers, which indicates the mental application of a rational *tampering budget*, leading to a concentration on artifacts being easier to manipulate, and (4) we establish an understanding of the reliability of tampering indicators and derive factors that influence the tamper resistance of an artifact.

For additional details and arguments, we refer the reader to the extended version of this paper [25].

2 User study design

This section opens with a description of the tampering task scenario, followed by a detailed explanation of the user study, which consisted of four phases: (1) a pre-tampering questionnaire, (2) a tampering task, (3) a post-tampering questionnaire, and (4) a set of semi-structured interviews.

2.1 Participants

The user study was conducted during an advanced lecture on digital forensics at the Friedrich-Alexander Universität (FAU) in the fall of 2023, with 35 registered students. Of these, 10 students completed phases (2) and (3), and three of them additionally participated in the interviews (4).

2.2 Scenario of the tampering task

The user study assumes that a perpetrator who has full control over a running system (administrator) wants to cover the tracks of their deeds by swapping two events E_2 and E_3, as depicted in Figure 1, i.e., make an examiner believe that the download of illegal material (E_3) occurred before the perpetrator became aware of its illegality (E_2). More details about the scenario can be found in Appendix A.

In the real world, swapping events may impact the criminal intent ('Mens rea'). Here, the updated sequence ($E_1 - E_3 - E_2$) suggests that the criminal liability of the downloaded material was unknown to the suspect when the web browsing was conducted. This particular type of tampering allows us to learn more than

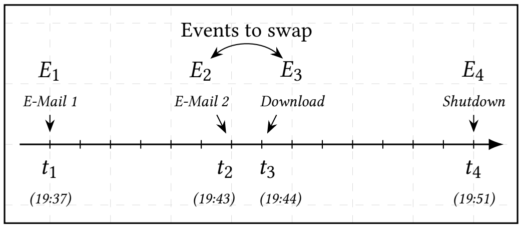


Figure 1: Artificial sequence of events $E_1 - E_4$ imagined for the tampering task.

demanding trace deletion or addition, as we are interested in the resilience of artifacts and the probability that they are taken into account.

2.3 Pre-tampering questionnaire

The user study started with a pre-tampering questionnaire aiming at gathering a set of background information about participants. The questionnaire included a total of 15 single/multiple-choice(s) questions centered around the respondents' experience. The questions concerned their teaching curriculum such as their course of study or graduate degree (Q1-Q4, Q10, and Q11), their experience with digital forensics lectures (Q8-Q9), and practical work (Q12-Q14). The remaining questions aimed at capturing their workload and motivation (Q5, Q6), as well as the effort they were willing to put into the study (Q7, Q15).

2.4 Tampering task

Participants were provided with the full virtual machine image of a Windows 10 Home workstation (VM turned off, administrator account credentials provided) after events $E_1 - E_4$ had been performed. They were asked to act like the suspect and swap events E_2 and E_3. All actions had to be performed on the running system (working on the disk images directly was prohibited) and the time boundaries E_1 and E_4 had be to respected (E_2, E_3 had to remain within E_1, E_4). Participants were given two weeks to complete the tampering task. They had to return a logbook containing notes of their actions and the VM itself.

2.5 Post-tampering questionnaire

Participants who returned a forged VM were asked to fill out a post-tampering questionnaire, which aimed at capturing *how* the students went about the task. It comprised 22 questions and was separated into three sections referring to a different time of the tampering task: *before*, *during*, and *after*. Questions relating to *before* and *after* focused on the assessment of specific artifacts[1] that the respondent knew or manipulated (Q1-Q10 and Q15-Q22). For example, some questions asked participants which artifacts they manipulated and to rank them according to the perceived difficulty of tampering. The middle part targeted performed activities, e.g., how certain actions were performed (Q11-Q14). Respondents were asked to place themselves at these times and provide detailed insights into their experiences, decisions, and observations. They were encouraged to use their logbook to answer the questions.

2.6 Semi-structured interviews

Respondents were invited to take part in semi-structured interviews where three individuals accepted. These face-to-face interviews followed a generic outline with a set of open-ended questions articulated around the preliminary findings from the post-tampering questionnaire. The intention was to extract further insights into their experiences during the tampering task and qualitatively evaluate the difficulty of tampering with different artifacts.

2.7 Ethics

The ethics commission at the university in which the user studies and interviews were conducted does not handle non-medical studies for explicit approval, and therefore instead the experimental protocol followed their general data protection and ethics rules. Participation in the study was voluntary and integrated into the course as a non-graded exercise.

2.8 Data analysis

We adopted an iterative approach for the analysis of qualitative data derived from the questionnaires, logbooks, and interview transcriptions. Open-ended responses were inductively coded to extract thematic patterns or trends [20]. The logbooks were not coded as they primarily described the tampering process undertaken by each participant but were used to improve our understanding of their manipulations.

2.9 Limitations

The user study has three limitations: (1) it included only 10 participants which is small and may not sufficiently capture the variability of behaviors and strategies; (2) all participants were students sharing similar backgrounds and experiences; and (3) only one tampering scenario was developed which is only one possible instance of manipulation, and therefore gives only answers to the research question from the viewpoint of the scenario. An adversary may have different strategies such as hiding data or artifact wiping [5]. Given these limitations, it is important to understand that our goal was a qualitative and explorative study which allows for a detailed examination of each case. It can be seen as a first step towards better understanding strategies, challenges, and artifacts, as well as revealing new research directions such as factors influencing the trustworthiness of timestamps. While some may argue that identical backgrounds are a disadvantage, our study found that participants followed different strategies which would be less informative with a large/diverse group. Nevertheless, we acknowledge that a larger-scale study would add value and therefore we provide detailed descriptions of our tampering task.[2]

[1] We compiled a list of relevant, existing Win10 artifacts based on the *Plaso* documentation [14], i.e., existing parsers which can be found in Appendix B.

[2] All details to repeat this study are publicly available here [24].

3 Results: Tampering Preparation

This and the following two sections summarize the results we extracted from the questionnaires and interviews. We begin with insights on how participants prepared for the tampering task.

3.1 Participant background, experience, and knowledge

We anticipated that participants have differences in prior knowledge of (Windows) forensics which was confirmed through the post-tampering questionnaire (Q3). Based on their responses, participants can be categorized into two knowledge levels: novices (Participants 5, 6, 8, and 9) who had limited Windows forensics experience, and semi-experienced participants, who had a generic knowledge of browser-related artifacts. With few exceptions, participants consistently marked Windows Registry sources, the `$USNjrnl`, and the Thunderbird Global Database as unknown. In contrast, most were familiar with Firefox-related data, Thunderbird's Inbox file, the `$MFT`, the `$RECYCLE.BIN`, and Windows event logs. During the interviews, we learned that most participants regularly used Linux.

3.2 Participant initial thoughts and designs

The first question of the post-tampering questionnaire invited the participants to reflect on their initial strategies, before starting the task. All participants agreed that accomplishing the task of making it appear as if E_3 happened before E_2 required modifying one of the two events. The common strategy was therefore to select a fixed event that would act as a pivot point for re-arranging other events. Interestingly, we observed that half chose E_2 (opening the second email) as their reference event while the other half preferred E_3 (browsing and downloading with Firefox). Participants within groups using the same pivot point expressed similar decisive factors in their choice to re-arrange the other event:

Level of knowledge: Participants 4, 6, and 7, who chose Thunderbird (E_2) as their pivot point, mentioned their greater familiarity with Firefox—gained from their introductory lecture—as a key factor in their decision of manipulating Firefox. In contrast, Participant 10 chose Firefox as the pivot point, having limited prior knowledge and having heard that Thunderbird would be easier to manipulate.

Volume of linked artifacts: The volume of linked artifacts refers to the number of artifacts associated with E_2 and E_3 that would need to be modified. All participants who selected E_3 as their pivot point agreed that Thunderbird appeared easier to manipulate than Firefox because it involved modifying a smaller volume of data. Four participants expressed concerns about the higher number of files to manipulate with the Firefox strategy, which they felt increased the likelihood of errors. Participant 2 noted: "My idea was to change as little as possible to minimize the potential for errors". Similarly, Participants 1 and 9 were worried about generating excessive second-order traces.

Correlation with remotely stored information: Another important consideration was the limited control over external data storage. This concern originated from the possibility that an unaltered "true copy" of the targeted data might exist in a remote location. This concern dissuaded Participants 3, 5, and 6 of the E_2 pivot point group from manipulating Thunderbird. For instance, Participant 5 indicated that the "full analysis of other sources than just the machine [would] probably give the real course of events away". These sources include an ISP, a DNS, or a mail server.

Maintaining internal artifact consistency: This factor refers to the relationship between the organization of an artifact and the data it contains. For instance, SQLite databases organize their entries following a specific allocation strategy. Hence, tampering with timestamps in entries may be exposed when looking at the order of (raw) entries and identifiers in the database. Participant 9 highlighted this issue: manipulating Firefox artifacts can be "identified by logical inconsistencies such as timestamps not matching the order in which events are listed in a cache/log file".

3.3 Preparation

In the post-tampering questionnaire, 9 out of 10 students reported that they had undertaken preparation beforehand. We learned that the types of preparation steps varied from one participant to another. Some mentioned gathering information through literature, forums, and/or tools review, while others took a hands-on, 'learning by doing' approach within the VM.

To evaluate the influence of preparation on their knowledge of Windows artifacts, we compared responses from Q3 and Q15. We observed that informed participants showed minimal changes in knowledge, while novices experienced significant gains, reaching similar overall scores. The former relied on literature reviews, testing and AI-based inquiries, focusing on artifacts related to the event to modify (Firefox- or Thunderbird-related artifacts). Overall, all participants, except Participant 9, achieved a comparable baseline knowledge before commencing the task.

4 Results: Tampering Actions

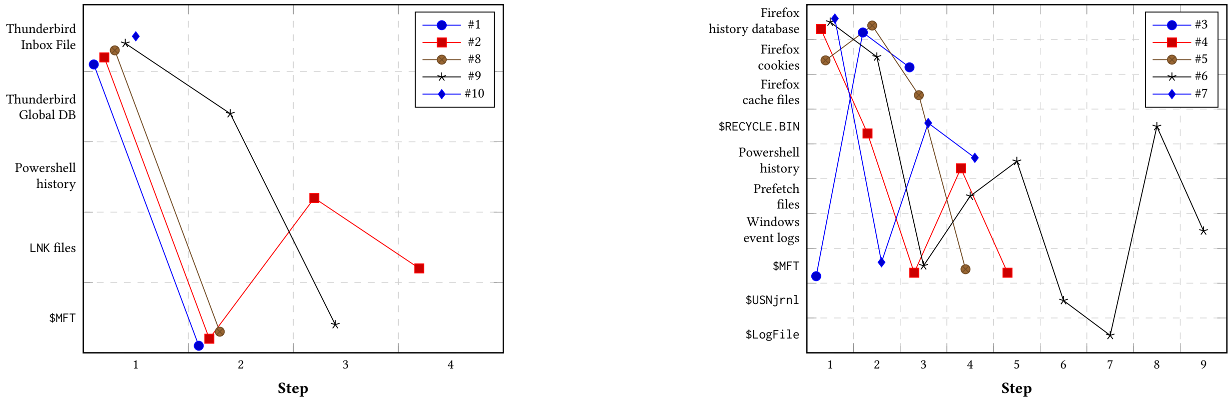
This section describes the execution of the tampering task: the tampering process itself, as well as the handling of second-order traces.

4.1 Tampering approaches

As discussed before, all participants shared the common strategy to decide on a reference event between E_2 and E_3 that is used as a pivot point to swap the other event.

Figure 2 illustrates the sequence of manipulations performed within the groups using the same pivot point. On the y-axis are artifacts that were manipulated and/or removed by students within each group, ordered according to their hierarchical position in the abstraction layers of the system (higher levels: application to lower levels: file system) [2]. The x-axis illustrates the sequence of manipulation steps undertaken by participants.

While the sequence of manipulations is intrinsically linked to the conflicting goal of dealing with second-order traces, as further discussed in Section 4.2, the graphs show a visible trend of descent through abstraction layers among participants. After choosing their pivot point, all students started by manipulating artifacts directly related to the unfixed event to re-arrange, at the application level. For instance, all participants in the Firefox pivot point group started with the manipulation of timestamps within Thunderbird's Inbox file. We then observe that further manipulations concerned artifacts in lower layers, starting with artifacts at the OS layer and addressing file system artifacts (apart from the `$MFT`) last.



(a) Thunderbird group

(b) Firefox group

Figure 2: Manipulation sequences showing participants as connected lines over time.

In regards to specific timestamp manipulations, participants attempted to maintain consistency across manipulations to fit the scenario, e.g., avoiding instances where a picture's download time precedes the visit to the corresponding web page. To achieve consistency, most participants added or removed (according to the strategy) constant time offsets.

4.2 Dealing with second-order traces

All participants showed concern about second-order traces. This carefulness is reflected in their strategy design (Section 3.2), choice of tooling, and sequence of manipulations. Overall, three distinct approaches emerged from the findings.

Clandestine approach: Participants using the clandestine approach proactively minimized the creation of second-order traces, rather than focusing on their removal after they had been generated. Three participants anticipated the creation of second-order traces by carefully choosing the tools they would use in the task. This included preferring tools that hypothetically leave fewer traces (e.g., command prompt over Powershell), benign-looking tools like DB Browser for SQLite, avoiding installing anti-forensic software, and using portable versions loaded on a USB stick.

Tampering-focused approach: This approach concerns two participants who pertained to the act of tampering itself without considering the generation of second-order traces, as "it would be practically impossible and would continue indefinitely". The only action undertaken by both participants was the update of the file system metadata of Thunderbird's Inbox file and the transfer of files to the $RECYCLE.BIN.

Mixed approach: The mixed approach involves participants who were mindful of second-order traces generation and actively engaged in recursively removing them from the system. Participants began by focusing on their main target (Firefox- or Thunderbird-related artifacts). They then adopted a wrapping-the-onion method, systematically working to erase not only the second-order traces of their tampering but also the subsequent layers that emerged from their efforts to conceal this tampering, and so on, in an iterative process. They also showed sophisticated efforts to minimize the generation of second-order traces by employing methods such as tool name obfuscation and downloading scripts from a web server.

5 Results: Tampering Difficulties

Participants experienced failures that forced them to adjust their strategy which can be divided into two classes: (1) practical challenges and (2) perceptions. On top of that, most participants expressed having encountered technical difficulties when manipulating specific artifacts, which are discussed in Section 6.3.

5.1 Practical challenges in tampering

These aspects cover the range of issues that the participants faced when working on the task, such as the installation or use of tools. When facing such practical challenges, we can see that participants were forced to make compromises on certain aspects of their strategy. For instance, Participant 3 encountered difficulties when installing Powershell modules, forcing them to install external software and deviating from their original plan of minimizing suspicions. Another participant had their external software recognized as a virus by Windows Defender and was forced to obfuscate its name and download it via a personal web server.

5.2 Perceptions of knowledge and relevance

Some participants adjusted their strategy in response to soft factors like their knowledge level and the perceived relevance of manipulating a specific artifact. For instance, Participant 6 chose not to alter the Windows event logs due to the absence of clear indications of their actions. Similarly, Participant 4 did not manipulate these logs, lacking the knowledge about where to find pertinent information and how to modify it.

6 Discussion and reflection

This section revisits the research questions in light of our findings. To generalize our results, we also discuss the technical aspects identified that affect the tamper resistance of specific artifacts.

6.1 Adversary's perspective

RQ1: What strategies do adversaries employ in planning and executing tampering with the temporal order of events? When tasked with swapping two events, we observed that all tamperers followed the same strategy: deciding on a reference event and using it as a pivot point to re-order the other event. The sequence of manipulations in re-ordering this unfixed event appeared to be closely related to the placement of each connected artifact in the abstraction layers. Only a few participants manipulated traces in the lower layers along this hierarchy, which from a forensic perspective, is relevant. Despite planning, several participants encountered unexpected difficulties during their manipulation process or reported various factors influencing their ability to tamper with certain artifacts. Those are either inherent to the intrinsic nature of the targeted artifact, such as its complexity, or the operating system/setting in which the artifact resides, e.g., the availability of tools to facilitate the manipulation in that environment.

RQ2: How do adversaries deal with (new) traces stemming from their manipulations? Adversarial actions generate new artifacts. It is particularly difficult to maintain a comprehensive overview of all newly created artifacts, and attempting to manipulate every subsequent artifact can become an endless endeavor. In addition, the manipulation may create inconsistencies, such as relative sequences or implicit timestamps, that can be detected. Of the approaches used by participants, the mixed approach is the most sophisticated, where participants not only anticipated the creation of second-order traces but also recursively removed the traces of their deeds. For the tamperers, this is a never-ending conflict of goals between the manipulation of n-order traces associated with the event to re-order and reducing the generation of $n + 1$-order traces originating from the manipulation process itself.

6.2 Artifacts

RQ3: What makes an artifact more difficult to tamper with compared to another? Findings suggest that several artifacts are more "tamper-proof" compared to others based on the technical challenges and difficulties faced by our tamperers: (1) the correlation with remotely stored information, maintaining internal artifact consistency and the volume of linked artifacts (defined in Section 3.2 and combined here as a *implicit time information* factor), as well as additional challenges such as (2) the placement in the abstraction layers (see Section 4.1), (3) the existence of integrity checks, (4) the assigned permissions, (5) encryption, and (6) the availability of software to edit artifacts on the system. These technical aspects are complemented by soft factors such as perception or knowledge which depend on the experience/sophistication of an adversary.

6.3 From tamper-proof to resistance factors

As artifacts differ in their suitability to be manipulated, this means that they have special features (or factors) that make them easy or difficult to manipulate. These factors are examined briefly in this section and are discussed in more detail in our previous work [26].

Permissions: Various operations on Windows are protected via User Account Control (UAC) and require Administrator privileges. Consequently, one factor is the level of permissions required to modify an artifact. On many system configurations, including the one in this study, the user is an Administrator of the system in question. Therefore, in many cases, the UAC interface presents a little barrier to accessing the protected files, other than clicking 'Allow'. On the other hand, running a command as Admin may trigger other events or be logged.

Integrity checks: An artifact might have embedded mechanisms used to verify that data has not been altered or corrupted. For example, email signatures are generated over the content of the email, which may include time information. Modifying this information may result in an invalid signature and trigger suspicion. This becomes even more challenging when monitoring systems such as auditd (Linux) are used.

Software availability: Manipulations require some sort of tool, which may be a text editor, regedit (both available on most Windows systems), or more specialized tools such as a hex editor or database modification tool. New tools may require an installation creating artifacts of their existence. Some may qualify as anti-forensics tools according to Conlan et al. [5] while others may be less suspicious. If no tool is available, an adversary may have to reverse engineer an artifact, which requires sophisticated knowledge.

Placement within the software stack: This factor refers to the level at which an artifact or process is positioned within the hierarchical layers of software architecture. As discussed in Section 6.1, this impacts the modification or accessibility of an artifact.

Implicit timing information: In addition to timestamps, manipulations may lead to logical inconsistencies within an artifact. For instance, a database appends new entries with an increasing ID which means potential timestamps in a column should also increase. This implicit timing information may not be known to the tamperer and can now be integrated into digital forensic timelines [6]. In addition, implicit timing information may also be evidence that cannot be controlled due to its residence on an external source.

Encryption/format: The artifact requiring manipulation may be in a proprietary format or even encrypted (this is related to software availability) impeding a modification. Considerations include the type of encryption software implementation, as well as whether the keys are available, recoverable, or not.

The possibility of evidence tampering should be considered during the investigation, encouraging examiners to look for inconsistencies. The factors presented here directly impact the tamper resistance of traces and offer a vast potential to improve the interpretation of tampering because they provide clear guidance in determining the reliability of artifacts and the likelihood of them being changed. Given that adversaries have a finite amount of resources leading to a confined *tampering budget*, they will likely fail to produce perfect forgeries. Consequently, these tamper resistance factors can be used to evaluate artifacts that contain such discrepancies and reconstructing what may have caused them.

7 Conclusion

There is great interest in concealing crime. One way to do this is to tamper with digital traces afterward. This tampering poses a significant threat to the reliability of forensic event reconstruction. Our user study sheds light on previously unexplored aspects of live system tampering. Through a user study involving 10 graduate students tasked with swapping two past events, we identified a general tampering strategy which is to decide on one event that would act as a pivot point to re-arrange the other event. We also concluded that manipulations generate new traces that need to be hidden or manipulated as well, resulting in an endless cycle of manipulations. Compared to dead tampering, this conflict of goals between tampering and removing the traces of tampering increases the difficulty of creating perfect forgeries. Furthermore, we generalized our results and derived factors that influence the tamper resistance of artifacts, such as embedded integrity checks and artifact placement within the software stack. These factors need more discussion, however, they guide practitioners about the reliability of an artifact especially if two contradicting artifacts are found. We believe that the qualitative findings from our study on live tampering will improve the understanding of criminal efforts to conceal their activities and aid in their reconstruction.

Use of AI writing assistance

At least one author of this paper used ChatGPT-4 and the Grammarly plugin to assist in correcting typographical and grammatical errors and refining the phrasing of certain sentences. All recommendations were thoroughly evaluated and modified when needed before being integrated into this paper.

Acknowledgments

We wish to thank the students from the course on “Advanced Forensic Computing” at FAU for their participation. We also thank the anonymous reviewers for their helpful comments on previous versions of the paper. This work was supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Research and Training Group 2475 “Cybercrime and Forensic Computing” (grant number 393541319/GRK2475/2-2024).

References

[1] Michael A. Caloyannides. 2003. Digital "evidence" and reasonable doubt. *IEEE Security & Privacy* 1, 6 (2003), 89–91. https://doi.org/10.1109/MSECP.2003.1266366

[2] Brian D. Carrier. 2003. Defining Digital Forensic Examination and Analysis Tool Using Abstraction Layers. *International Journal of Digital Evidence* 1, 4 (2003), 1–12. http://www.utica.edu/academic/institutes/ecii/publications/articles/A04C3F91-AFBB-FC13-4A2E0F13203BA980.pdf

[3] Eoghan Casey. 2020. Standardization of forming and expressing preliminary evaluative opinions on digital evidence. *Forensic Science International: Digital Investigation* 32 (March 2020), 200888. https://doi.org/10.1016/j.fsidi.2019.200888

[4] William J. Chisum and Brent E Turvey. 2000. Evidence dynamics: Locard's exchange principle & crime reconstruction. *Journal of Behavioral Profiling* 1, 1 (2000), 1–15.

[5] Kevin Conlan, Ibrahim Baggili, and Frank Breitinger. 2016. Anti-forensics: Furthering digital forensic science through a new extended, granular taxonomy. *Digital Investigation* 18 (Aug. 2016), S66–S75. https://doi.org/10.1016/j.diin.2016.04.006

[6] Lisa M. Dreier, Céline Vanini, Christopher J. Hargreaves, Frank Breitinger, and Felix Freiling. 2024. Beyond timestamps: Integrating implicit timing information into digital forensic timelines. *Forensic Science International: Digital Investigation* 49 (2024), 301755. https://doi.org/10.1016/j.fsidi.2024.301755

[7] Felix Freiling and Leonhard Hösch. 2018. Controlled experiments in digital evidence tampering. *Digital Investigation* 24 (March 2018), S83–S92. https://doi.org/10.1016/j.diin.2018.01.011

[8] Michael Galhuber and Robert Luh. 2021. Time for Truth: Forensic Analysis of NTFS Timestamps. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES 21)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3465481.3470016

[9] Simson Garfinkel. 2007. Anti-forensics: Techniques, detection and countermeasures. In *2nd International Conference on i-Warfare and Security*, Vol. 20087. 77–84.

[10] Ryan Harris. 2006. Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital Investigation* 3, Supplement (2006), 44–49. https://doi.org/10.1016/j.diin.2006.06.005

[11] Xiaodong Lin. 2018. *Introductory Computer Forensics: A Hands-on Practical Approach*. Springer International Publishing. https://doi.org/10.1007/978-3-030-00581-8

[12] Farhad Manjoo. 2001. Unix Tick Tocks to a Billion. *Wired* (2001). https://www.wired.com/2001/09/unix-tick-tocks-to-a-billion/

[13] Andrew Marrington, Ibrahim Baggili, George Mohay, and Andrew Clark. 2011. CAT Detect (Computer Activity Timeline Detection): A tool for detecting inconsistency in computer activity timelines. *Digital Investigation* 8 (2011), S52–S61. https://doi.org/10.1016/j.diin.2011.05.007 The Proceedings of the Eleventh Annual DFRWS Conference.

[14] Joachim Metz. 2024. Welcome to the Plaso documentation. https://plaso.readthedocs.io/en/latest/

[15] MITRE ATT&CK. 2020. MITRE ATT&CK v15.1, Indicator Removal: Timestomp. https://attack.mitre.org/versions/v15/techniques/T1070/006/

[16] Christian Moch. 2015. *Automatisierte Erstellung von Übungsaufgaben in der digitalen Forensik*. Ph. D. Dissertation. University of Erlangen-Nuremberg. https://d-nb.info/1068781181

[17] Alji Mohamed and Chougdali Khalid. 2019. Detection of Timestamps Tampering in NTFS using Machine Learning. *Procedia Computer Science* 160 (Jan. 2019), 778–784. https://doi.org/10.1016/j.procs.2019.11.011

[18] Christopher Neale. 2023. Fool me once: A systematic review of techniques to authenticate digital artefacts. *Forensic Science International: Digital Investigation* 45 (June 2023), 301516. https://doi.org/10.1016/j.fsidi.2023.301516

[19] David Palmbach and Frank Breitinger. 2020. Artifacts for Detecting Timestamp Manipulation in NTFS on Windows and Their Reliability. *Forensic Science International: Digital Investigation* 32 (April 2020), 300920. https://doi.org/10.1016/j.fsidi.2020.300920

[20] Johnny Saldaña. 2021. *The coding manual for qualitative researchers.* SAGE Publications.

[21] Chris W. Sanchirico. 2004. Evidence Tampering. *Duke Law Journal* 53, 4 (2004), 1215–1336.

[22] Janine Schneider, Julian Wolf, and Felix Freiling. 2020. Tampering with Digital Evidence is Hard: The Case of Main Memory Images. *Forensic Science International: Digital Investigation* 32 (2020), 300924. https://doi.org/10.1016/j.fsidi.2020.300924

[23] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. 2020. *MITRE ATT&CK: Design and philosophy.* Technical Report MP180360R1. The MITRE Corporation.

[24] Céline Vanini, Jan Gruber, Christopher J. Hargreaves, Zinaida Benenson, Felix Freiling, and Frank Breitinger. 2024. Guidelines and Questionnaires for a User Study on Live Timestamp Tampering in Digital Forensic Event Reconstruction. https://doi.org/10.48657/ydrk-qa98 (Version 1.0).

[25] Céline Vanini, Jan Gruber, Christopher Hargreaves, Zinaida Benenson, Felix Freiling, and Frank Breitinger. 2024. Strategies and Challenges of Timestamp Tampering for Improved Digital Forensic Event Reconstruction (extended version). https://doi.org/10.48550/arXiv.2501.00175 arXiv:2501.00175 [cs.CR]

[26] Céline Vanini, Chris Hargreaves, and Frank Breitinger. 2024. Evaluating tamper resistance of digital forensic artifacts during event reconstruction. https://doi.org/10.48550/arXiv.2412.12814 arXiv:2412.12814 [cs.CR]

[27] Svein Yngvar Willassen. 2008. Finding Evidence of Antedating in Digital Investigations. In *Proceedings of the 2008 Third International Conference on Availability, Reliability and Security (ARES '08)*. IEEE Computer Society, USA, 26–32. https://doi.org/10.1109/ARES.2008.149

A Scenario details

The following case scenario and task were given to the participants:

> Albert A. is accused of the illegal possession of “rhinoceros” images. In November 2023, the police seized his computer in his home and found several rhino images. Albert A. uses his computer at home for private purposes. Albert A. claims that he came across these images by accident, not knowing that they were illegal. In contrast, the prosecution claims that Albert A. knew that rhino images were illegal before he downloaded the images.
>
> You are given the full computer (shut down virtual machine) of Albert A.'s computer after actions $E_1 - E_4$ have been finished. After completing E_3, Albert A. thinks that sequence $E_1 - E_2 - E_3$ does not look good. He wants to switch actions E_2 and E_3. Play the role of Albert A. Boot the system and manipulate the

computer such that “it looks as if” E_3 happened before E_2. Results will be analyzed by experts assessing the sequence of actions E_1, E_2, and E_3.

In this synthetic scenario:

- E_1 (at time t_1): An email is received which asks the receiver, whether he has already seen ‘Rhinocerotidae’, which is ‘really, really hot material’.
- E_2 (at time t_2): Another email is received in which the sender clearly states that ‘Rhinocerotidae’ is a term referring to illegal material. The user opens the message and views the attachment.
- E_3 (at time t_3): The user opens a browser and issues a search query for ‘Rhinocerotidae’, visits the Wikipedia website on ‘Rhinoceros’, and downloads several rhino images.
- E_4 (at time t_4): The user shuts down the computer.

B Windows 10 Forensic Artifacts

Before designing the questionnaire, we compiled a list of relevant, existing Windows 10 artifacts based on the *Plaso* documentation [14]. This included artifacts within the registry and other user application or OS-related artifacts such as LNK files or the `$RECYCLE.BIN`. In addition, we added several artifacts that we deemed relevant regarding our tampering scenario, but are currently not considered by Plaso such as Thunderbird’s Inbox file and Global database (message index system). The complete list can be found below in Table 1.

Table 1: Catalog of Windows artifacts derived from Plaso parsers.

Layers	Sources
Application	Files internal metadata Firefox cache files Firefox cookies Firefox history and downloads database Microsoft Edge cache files Microsoft Edge history and downloads database OneDrive synchronization logs Thunderbird Inbox file Thunderbird Global database
OS	Amcache (registry) Bam (registry) Jumplists LNK files OpenSavePIDMRU / LastVisitedPIDMRU (registry) Prefetch files `setupapi.dev.log` Shellbags (registry) ShimCache (registry) USB/USBSTOR (registry) UserAssist (registry) Windows Event Logs Windows timeline database

File system	`$LogFile`
	`$MFT`
	`$RECYCLE.BIN`
	`$USNjrnl`