Forensic Science International: Digital Investigation 32 (2020) 300924

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

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

DFRWS 2020 EU – Proceedings of the Seventh Annual DFRWS Europe

Tampering with Digital Evidence is Hard: The Case of Main Memory Images

Janine Schneider, Julian Wolf, Felix Freiling*

Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany

ARTICLE INFO

Article history:

ABSTRACT

Tampered digital evidence may jeopardize its correct interpretation. To assess the risks in a court of law, it is helpful to quantify the necessary effort to perform a convincing manipulation of digital evidence. Based on a sequence of controlled experiments with graduate students and digital forensics professionals, we study the effort to manipulate copies of main memory taken during a digital investigation. Confirming previous results on hard disc image tampering, manipulating main memory dumps can be considered hard in the sense that most forgeries were successfully detected. However, while the effort to detect a manipulation is generally bounded by the tampering effort, some forgeries fooled the analysts and caused analysis effort that was higher than the manipulation effort. The detection effort by graduate students, however, was generally higher than that of professionals. We study different manipulation and detection approaches and their success. Overall, tampering with main memory dumps appears to be harder than tampering with hard disc images but the probability to fool an analyst is higher too.

1. Introduction

Digital evidence is an increasingly important form of evidence in courts today. As with other types of evidence, the manipulation of digital evidence can lead to wrong conclusions. For example, assume an investigator finds a specific entry *x* in the browser history on the computer of a suspect. Commonly this will lead to the conclusion that some user of that computer has visited the website *x* in the past. But it may also be the case that the website was never visited by the computer, but rather the browser history was tampered with and evidence of that visit was planted by intent to lead an investigator to wrong conclusions. Questions of evidence manipulation reportedly surface in defense cases where the defendant claims that he or she did not commit the offense (such as performing a cyber-attack or downloading illegal documents) but an alternative hypothesis, like the actions of a Trojan horse is more plausible (Brenner et al., 2004).

The dangers of evidence tampering are not unique to digital evidence, but there appears to be a common belief that it is easier to tamper with digital evidence than to tamper with physical evidence. For example, Kumar et al. (2006, p.316) write that

> [digital evidence] is less tangible, highly volatile and relatively easier to be tampered compared to physical evidence.

Similarly, Caloyannides (2003, p. 89) states that digital data

> […] can be manipulated at will, and depending on the manipulator's sophistication, the alteration can be undetectable, regardless of digital forensics experts' competence and equipment.

This common belief appears to be rooted in the fact that digital evidence results from the *abstraction* of physical evidence like magnetization of a storage medium (Dardick et al., 2014; Paul, 2009) into a discrete and clearly separable sequence of zeros and ones. This discrete and digital nature of the evidence allows easier control: If done correctly, flipping a bit leaves no traces in that bit or the surrounding bits. So at least in theory

> […] it is much harder to destroy or tamper with a bloody knife than it is to modify metadata on a critical file. [Lin (2008), p.18]

The very nature of digital evidence has allowed digital forensic science to create new methods to protect the integrity of such evidence. Cryptographic hashes can be computed on bit strings of arbitrary length and through their unique properties of collision resistance be used as a shorthand to document the state of

* Corresponding author.
E-mail addresses: janine.schneider@fau.de (J. Schneider), julian.jw.wolf@fau.de (J. Wolf), felix.freiling@fau.de (F. Freiling).

https://doi.org/10.1016/j.fsidi.2020.300924

arbitrarily large amounts of data. But clearly, cryptographic hashes cannot prevent any type of tampering since manipulations can still occur either by performing tampering before the first hash is taken or by simply tampering with both the evidence *and* the documented hash.

So, in practical cases, the plausibility of evidence tampering needs to be assessed routinely. In analogy to physical evidence, the following line of reasoning is commonly performed: If there are no clear signs of evidence tampering, the effort of tampering is high, and neither motivation nor competence of a possible manipulative actor for performing the tampering task is evident, then it is improbable that the evidence was tampered with. To actually assess the risk of evidence tampering, it is therefore crucial to assess the competences needed and the effort necessary to perform specific manipulations of digital evidence.

1.1. Related work

The arguably longest tradition of work on understanding digital evidence tampering is in the context of *anti-forensics* (Harris, 2006), more specifically, the analysis of manipulated (or counterfeit) evidence in multimedia security, e.g., where methods of blind image forensics can be used to detect manipulations (Johnson et al., 2006; Lin et al., 2009). However, we are not aware of any literature with a similar intention focusing on non-multimedia files.

Apart from some preliminary experiments by Moch (2005, Sect. 6.6–6.8, p. 143), the only work that has systematically explored the effort to perform targeted evidence manipulation of more traditional (non-multimedia) data was performed by Freiling and Hösch (2018), who performed a controlled experiment with human investigators. In their study, graduate level students, all of them with basic digital forensics education from an earlier course, had to perform an evidence tampering task on a standard Ubuntu Linux disc. The tampering task was to manipulate the data on the hard disc image such that a forensic investigator will reach the conclusion that a particular website had been visited in a particular time period in the past. The manipulated disc images were called *forgeries*. Next to forgeries, Freiling and Hösch (2018) also prepared *originals*, i.e., system discs with which the website had actually been visited. Students then had to perform a forensic analysis of a random disc image and had to distinguish between original and forgery. Participants had to document their effort and fill out a pre-study questionnaire. In total, data from 14 participants was collected.

As a result, all forgeries were correctly detected and only one of the originals was falsely classified as a forgery. Their data showed that, on average, correctly classifying an original took considerably more effort than to correctly classify a forgery. Furthermore, the analysis effort was always bounded by the effort to create a forgery. All of this was achieved given full control by participants over the evidence, i.e., they could use any tool they wanted and manipulate any bit on the image in an offline fashion.

1.2. Research goal and contributions

While the results of Freiling and Hösch (2018) show that the specific task of tampering with hard disc images is "hard" (i.e., none of the forgeries fooled the analysts) and the effort to create forgeries is "high", it is not clear whether this is true for all tampering tasks and any type of digital evidence. So, in this paper we ask the question, whether the situation is different when main memory images and not hard disc images are the object of digital evidence tampering.

Given the relative immaturity of main memory analysis (Ligh et al., 2014) over hard disc analysis (Carrier, 2005), it is not clear whether the results of Freiling and Hösch (2018) carry over to that other domain of digital evidence: After all, the field of main memory acquisition is naturally more diverse, tool support for main memory analysis is much less established, and there is a more evident lack of trained personnel than there is in the field of hard disc acquisition and analysis. All these aspects can be positive and negative from the viewpoint of an analyst. We therefore expect that the sophistication and expertise of the humans involved might make a larger difference, and as such, level the fields between manipulation and detection.

In this paper, we consider the following tampering task: In the memory dump of a Kali Linux Laptop, plant evidence of a fictitious previous network connection. In this context, we study the effort to perform an evidence manipulation and the effort to detect manipulations. We do this by running two controlled experiments:

1. A tampering experiment analogous to Freiling and Hösch (2018) with graduate level students.
2. A tampering detection experiment with professionals that had to repeatedly classify multiple main memory images as original or forgery.

The main insights of these experiments are as follows:

- The first experiment was performed with graduate students with knowledge in digital forensics. Of 22 analyzed memory dumps, all but two were correctly classified. Interestingly, the only mistakes made were two forgeries which were incorrectly classified as originals.
- In the repetitive detection experiment with professionals, the classification task was performed with similar quality than the graduate students: Of 183 analyses performed, 159 were correct. 16 analyses resulted in forgeries being wrongly classified as originals and 8 analyses resulted in originals being wrongly classified as forgeries.
- The effort needed by the professionals for performing the classification task appears to be considerably less than the effort used by the graduate students. However, due to a mishap in the execution of the experiment, the measured effort between students and professionals was not comparable.
- The average effort for the correct classification of an original was generally higher than the effort for the correct classification of a forgery. However, there was no clear correlation between manipulation and analysis effort. In fact, the linear regression of the results indicates that higher manipulation effort leads to lower analysis effort.
- The repetitive classification experiment did not exhibit clear signs of speedup over multiple tasks of the same category.

1.3. Paper outline

We describe the scenario and the research questions in Section 2 followed by a description of the experimental design in Section 3. We then report on the results of the individual experiment (Section 4) and the repetitive groups experiment (Section 5). Section 6 concludes the paper. We assume some basic knowledge in the tools and techniques of main memory forensics. As a possible introduction please consult Ligh et al. (2014).

2. Scenario and research questions

2.1. Main memory image manipulation

The focus of our experiments is the detection of tampered main memory images into which incriminating evidence is planted. It is

inspired by cases of corrupt investigators who drop a packet of drugs while performing a search of the premises of a "suspect" (Flynn, 2019). In the realm of digital evidence such actions correspond to evidence tampering before the first cryptographic hash is documented.

The fictitious scenario was as follows:

In October 2018, the police identifies a server on the Internet, which runs an illegal website selling drugs. From wiretapping, the police observes a couple of ssh connections and thereby identifies another machine belonging to a person called Werner Weber. The police suspects Werner Weber to be the administrator of the website and obtains a search warrant for his house. During the search, the police finds Werner's laptop and acquires a main memory dump of that machine using LiME. Directly after, electricity fails on the server and the laptop. Both computers shut down, making their encrypted discs inaccessible. The only evidence remaining is the memory dump of Werner Weber's laptop.

The tampering risks involved in this scenario arise in case traces of a ssh connection to the illegal server are found in the memory dump. If they are found, they can be either the result of a *real* ssh connection having taken place in the past. Or alternatively, these traces could have been planted by the investigator in charge of handling the evidence.

2.2. Research questions

In this paper, we focus on the situation with *full control* (as in previous work (Freiling and Hösch, 2018)), i.e., tampering and analysis tasks are performed in an offline scenario and there is no restriction in the tools used. This means that main memory acquisition was not performed during the experiments but rather memory dumps were provided to participants as files.

In this context, we ask the following questions:

- Given a comparable scenario to previous work (Freiling and Hösch, 2018), in what way is main memory image tampering similar or different to hard disc image tampering, both regarding difficulty to produce good forgeries and the bound on analysis effort?
- What makes good (i.e., wrongly classified) forgeries good? Which factors influence their success?
- What factors influence the speed of the tampering analysis? Does it help if similar detection tasks have been performed before? (positive effect of training?)
- What techniques do people apply to produce and (successfully) detect manipulations of main memory dumps?

3. Experimental design

The experiment is divided into three parts:

- (Part A) A tampering task conducted with students
- (Part B) An individual analysis task also conducted with students
- (Part C) A repetitive group analysis task with teams of professionals

3.1. General considerations

To control for the influence of previous knowledge, all participants had to fill in a questionnaire in advance. In Part A and B, the questionnaire was also used to gather information about the motivation, experience, and demographic data of the participants. Additionally, all participants were required to log their effort which they invested in the study. Participants of Part A and B should log their effort in a project diary whereby only continuous efforts of at least 30 min had to be reported. The efforts of Part C were logged automatically.

Since Part B was an individual student analysis and Part C was a professional group analysis, we define two effort values. On the one hand, the tampering task and individual analysis effort in parts A and B are called *person effort*. The person effort is given in minutes. On the other hand, the group analysis effort is called *team effort*. The team effort is given in seconds.

For anonymization each participant or team had to create an individual/team pseudonym. During the experiments, an exchange of information between participants or teams was not allowed. All participants had to sign or acknowledge a permission that their data could be collected and processed for research purposes.

The main memory dumps were created using LiME and a virtual machine running Kali Linux 2018.4 x64.

3.2. The tampering task (part A)

First, we studied the effort required to tamper with a main memory dump. Therefore, we created a tampering task where a group of participants were given a clean main memory image, i.e., one from which the server from the scenario described above was never accessed. The participants received the main memory image via an individual download link against a receipt. They also received the ssh key to access the server and the sudo password for the administration of the server. There were no restrictions regarding the tampering actions performed upon the image, i.e., participants were allowed to tamper with the image in any way they liked. The only requirement was to fulfill the following task:

- Tamper with the main memory image so that an analyst comes to the following conclusions:
  - There was definitely an active network connection to the server at the time of main memory backup or before that.
  - Administrative commands were executed on the server over this network connection.

Besides the submission of the tampered image the participants were also required to submit a description of their tampering approach and a project diary documenting their person effort. Fig. 1 visually summarizes the design of the tampering task.

3.3. The individual detection experiment (part B)

In the individual detection experiment, study participants were asked to examine a randomly chosen main memory image and to decide whether it was an original or a forgery. This is sketched schematically in Fig. 2. The random image pool was created through randomly drawing 15 forgeries from Part A and adding the 5 original images 3 times.

The participants got the main memory image via an individual download link against a receipt. They also received the kernel module used to generate the image, a volatility profile, the ssh key and the sudo password of the server. The investigation order requested answers to the following questions:

- Was there an active network connection from Werner Weber's computer to the server with a given IP address (the same one issued during the tampering task)?
- If yes: Was the server administered via this network connection, i.e., by executing some sudo command (like web server start/stop)?

The participants were told to answer the investigative questions

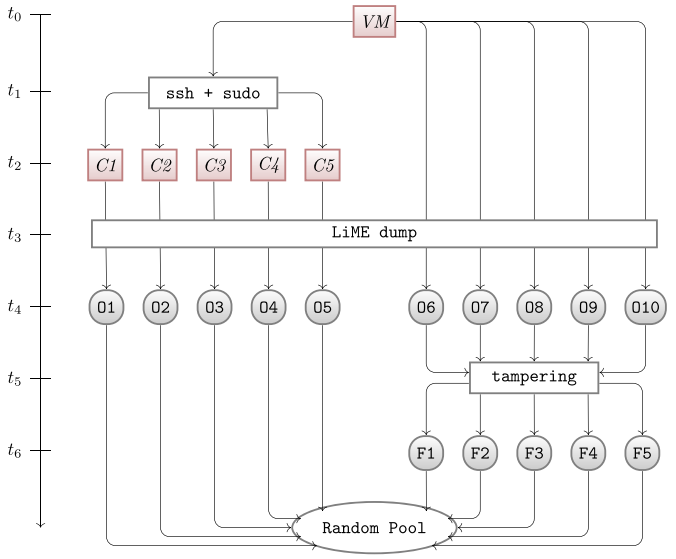

Fig. 1. Design of Part A of the experiment (tampering task). The timeline (left) depicts critical points: time were the Kali Linux VM was set up to simulate the suspect's laptop (t_0); a connection to the webserver via ssh was established (t_1) resulting in 5 originals (t_2) of which memory dumps were taken (t_3); original memory dumps (O) were added to the pool or handed out to the participants (t_4); dumps without ssh connections were tampered with (t_5) resulting in a set of forgeries (F) which were also added to the pool (t_6).

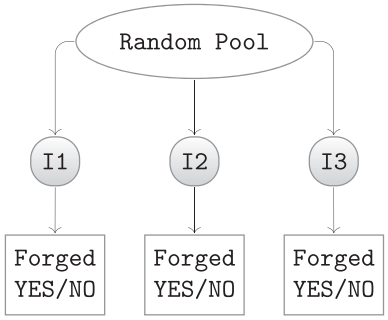

Fig. 2. Design of individual detection experiment (Part B). Participants drew an image (I) from the random pool and had to classify it as original or forgery.

in the form of a brief forensic report. As in Part A, participants were also required to log their individual (person) effort for the analysis of the image.

3.4. Repetitive detection experiment (part C)

The repetitive detection experiment was conducted in teams of participants, where each team tried to analyze as many images as possible in a specified time period. The teams received a sequence of main memory dumps (called challenges). These were the elements of the random pool created in Part A of the experiment. Overall, the pool contained 20 originals and 20 forgeries. For each challenge, participants should decide if the traces of an ssh connection were planted or not. The participants were allowed to use any tool they liked to analyze the memory dump regarding the question, whether the memory dump showed any traces of active network connections between the laptop and the server.

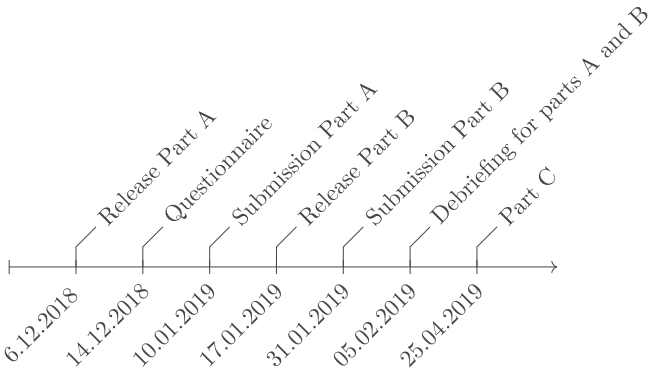

Fig. 3. Timeline of experiment (parts A, B and C).

4. Results of the tampering task and the individual detection experiment

We now report on the execution and results of parts A, B and C of the experiment. The overall timeline is given in Fig. 3.

4.1. Context and participants

To perform Part A and B of the experiment, an advanced course on digital forensics at Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) was used as setting. The course was held between October 2018 and February 2019 with roughly 30 students. From the pre-study questionnaire, we knew that all participants but one were computer science majors that had advanced technical skills and had successfully participated in an introductory course on digital forensics with an extensive file system forensic analysis part before October 2017. Furthermore, most of the participants were master students, with a moderate motivation to submit the tasks. Most of them stated that they would invest 20 to 30% of their weekly work time for the course. Participation in the experiment was mandatory, but the result was not graded and therefore, not part of the grade of the course. None of them had any experience in main memory forensics.

Overall there were 30 students in the course of which 23 participated in Part A of the experiment and submitted a tampered main memory dump. 22 students participated in Part B and submitted the investigation report. The intersection between the set of participants in Part A and Part B consisted of 19 students.

4.2. Tampering effort

Overall, 23 participants produced tampered images in Part A of the experiment. The resulting statistics are shown on the left hand side of Table 1. The right hand side contains the corresponding figures from the previous tampering experiment on hard disc drives Freiling and Hösch (2018). In comparison, the effort for tampering with main memory images was considerably higher in all parameters (on average almost twice as large).

4.3. Tampering success (part B)

For Part B, 15 of the 23 forgeries produced in Part A were selected for the random pool. The random pool additionally contained 15 originals. Thus, the random pool consisted of 30 memory images and the probability of drawing an image from each class in the pool was equal.

Overall, 22 students participated in Part B of the experiment. The results of their analyses together with the individual analysis

Table 1
Tampering effort measured in minutes for Part A (left) and comparison with HDD manipulation results (Freiling and Hösch, 2018) (right).

	our work	HDD experiment
	(Part A, $n = 23$)	(Phase 1.A (Freiling and Hösch, 2018), $n = 13$)
Min/max	300/2300	120/1260
Average	927	532
Stddev.	466	325
Median	810	450

effort are summarized in Table 2 within a confusion matrix. In total, 20 participants correctly classified their image from the random pool (12 originals and 8 forgeries). The only false classifications were 2 forgeries that were falsely classified as originals. The table shows that on average it needed more effort to correctly classify an original than to correctly classify a forgery. Within the study on the manipulation of hard disc drive it was the other way round (harder to correctly classify a forgery than an original). The detection rate, however, was equally good.

4.4. Relations between tampering and analysis effort

Considering the above numbers, the average person effort for correctly detecting a forgery (375 min) is lower than the average effort to create a forgery (927 min). This is analyzed in more detail in Fig. 4 where the tampering effort is plotted in relation to the detection effort for all 10 forgeries that were used in Part B of the experiment. Interestingly, the figure shows that the larger the tampering effort, the lower the detection effort. This contrasts with the findings of Freiling and Hösch (2018) where the trend was exactly opposite.

4.5. Factors influencing manipulation and analysis success

We compared the two participants who created successful forgeries with those who were not successful based on the data from the pre-study questionnaire.

The two participants who created successful forgeries were computer science master students. Both of them were highly motivated and stated that they were willing to invest 20–30% of their weekly work time in the course. One of them received a very good grade in the preliminary course. The other one got a good grade. Both of them planned to invest 2 to 3 h weekly for the course exercises. In comparison to the other participants neither of those two had more experience with main memory analysis than the others, was more motivated or was willing to invest more weekly hours in the course. However, both participants invested more time to produce their forgery than the other participants did.

The two participants who did a wrong classification were both computer science students. One of them was a bachelor, the other one a master student. Both of them stated that they are highly motivated. The bachelor participant was willing to invest 60–70% of the weekly work time into the course, the master participant 20–30%. The bachelor participant was the only participant which has not attended the introductory course on digital forensics. The master participant had a very good grade in the preliminary course. They stated that they planned to invest 4 and 2 weekly hours to process the exercises. However, the master participant did not participate in the tampering task and invested only 60 min in the individual detection experiment. This is in contrast to the average and indicates that experience helps to avoid falling for a forgery.

Table 2
Results of Part B of our experiment (individual detection, person effort measured in minutes).

	classified as original	classified as forgery
original	12	0
person effort range	150–795	–
person effort average	426.25	–
person effort median	385	–
forgery	2	8
person effort range	60–240	20–870
person effort average	150	375
person effort median	150	270

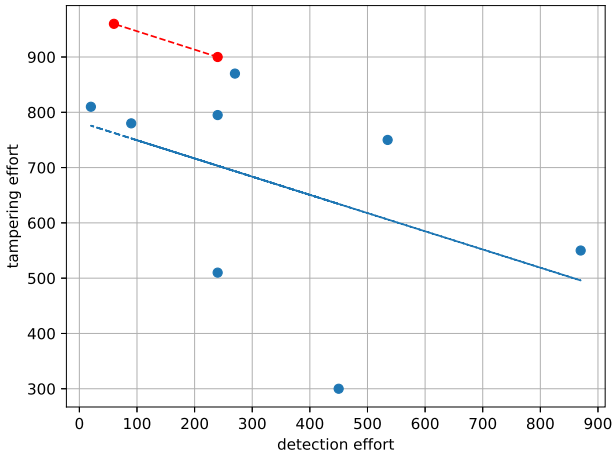

Fig. 4. Relation between tampering and detection effort for correctly (blue) and incorrectly (red) classified forgeries with a linear regression line for both cases.

4.6. Tampering approaches

For Part A of the experiment the participants were required to document their tampering approaches. The resulting documentation shows different tampering techniques. For example, some participants created a similar reference image with true traces of ssh connections, then compared and copied evidence from the reference to the forgery. Other participants created and customized completely new images, manipulated existing data structures or inserted data (e.g., ssh keys) by hand.

Overall, the following three classes of tampering approaches can qualitatively be formed from the data:

- Participants created a system which was similar to the fictitious scenario, took a memory dump from that system and did a sweep over the image to correct obviously wrong data. We call this approach *Simulate-and-Sweep*.
- Participants modified the existing structures of the given image by hand. We call this approach *Manual-Fix*.
- The Manual-Fix method was sometimes combined with an additional technique which inserted suspicious data at random or specific free locations in the dump and hope that investigator falls for it. We call this method *Manual-Fix++*.

Fig. 5 shows the relation between the tampering and detection effort for the different tampering approach classes together with the geographic center ("average location") per class. The geographic center was calculated by plotting the average detection effort over the average tampering effort for the class.

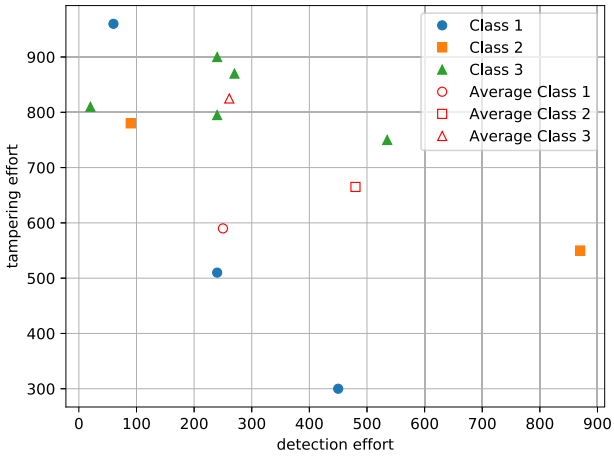

Fig. 5. Relation between tampering and detection effort for different tampering approach classes (1: Simulate-and-Sweep, 2: Manual-Fix, 3: Manual-Fix++) with average locations ("centers", in unfilled red).

4.6.1. Simulate-and-Sweep

The Simulate-and-Sweep (Class 1) consists of all tampering approaches, where a new memory dump was created that fits the fictitious scenario. Six participants created a new dump that fitted the scenario and based on what was known about the system where the dump was created. This approach makes it easier to control the results and easier to keep consistency. Furthermore, there are only minor modifications to the new dump necessary, e.g. tampering the IP addresses and timestamps.

One of the forgeries classified as original was created using this technique. Two forgeries in this class were not analyzed.

4.6.2. Manual-Fix

The Manual-Fix class (class 2) consists of all tampering approaches, where existing data was manually modified. 17 participants modified existing data structures in the dump. This mostly involved changing existing bash history entries, process lists and network sockets, but also less obvious things like ARP cache entries and filenames. This approach is easy to perform because appropriate existing entries must only be found and replaced. If this approach is used to tamper with the image one needs to consider the length and position of data like strings and IP addresses.

4.6.3. Manual-Fix++

The Manual-Fix++ Class (Class 3) is a subset of the Manual-Fix Class in which data was inserted at random or specific free locations in the dump. 10 of the 17 participants from the Manual-Fix class additionally added data at random and free locations in the dump. This included the password, ssh keys and commands that could have been executed on the server and can be expected to be found in a legitimate dump. This approach was mostly used by students who simulated the given scenario, created memory dumps and performed analysis of their simulation. Often two dumps were created for this, one before and one after connecting to a target via ssh. Those dumps were then compared regarding traces left in main memory. In some cases Manual-Fix++ was used to fool quick and random examination, for example string analysis, without previous simulation. One student explicitly cited previous research done by Davidoff (2008) regarding cleartext passwords that can be found in the main memory.

This approach was used to create one of the two false negatives and thus caused a forgery classified as an original.

Table 3
Detection-to-tampering ratios for different tampering approaches and the number of convincing forgeries (success) per class. Manual-Fix++ is a subset of Manual-Fix.

Method	*n*	ratio	success
Simulate-and-Sweep	6	0.42	1
Manual-Fix	17	0.72	0
→ Manual-Fix++	10	0.32	1

4.6.4. Comparison

Using the coordinates of the average position, we computed a *detection-to-tampering ratio* by dividing the x value (detection effort) by the y value (tampering effort). Ideally (for the analyst), large effort in tampering results in low effort in detection. Therefore, larger values of this ratio indicate that a class is "more successful" from the viewpoint of an attacker and therefore better to create good forgeries. The ratios for the individual approaches are given in Table 3. This shows that the "best" method appears to have been the Manual-Fix method. The potential "dirty" improvement of the Manual-Fix++ approach did not pay off and apparently lead to easier detection. However, the two forgeries that fooled the analysts came from those two classes with lowest detection-to-tampering ratio.

4.7. Detection approaches

In Part B of the experiments, the participants were required to report their forgery detection approaches.

Forgeries created with the Simulate-and-Sweep method appeared very consistent to the analysts, especially when looking at network and user activities. When analysts identified this type of forgery as such, this happened because expected information were missing, traces of unanticipated tool usage could be found and especially because of time inconsistencies between apparently manipulated and non-manipulated data structures. In one case the tampering included changing timestamps in the bash history, which then created an inconsistency with the process start time, which was not changed. This was paired with a wrong server operating system. In another case the forgery was created using onboard virtual machine tools instead of LiME. This lead to missing kernel module traces.

Manual-Fix and Manual-Fix++ forgeries were detected through inconsistencies in the bash history and the process list such as double execution of commands, time and date inconsistencies, inconclusive command sequences and incomplete entries. Furthermore, the analysts exposed the forgeries because of missing ssh keys, missing network connections and missing inodes. One forgery was exposed because the analyst recognized that the source port and the destination port in the found network connection was reversed. Furthermore, no typical local source port number was used and the representation of spaces in the hex dump of the bash history was wrong. The Manual-Fix++ approach failed because the analysts assessed the random inserted information as data that is stored in the main memory without any context.

Since the consistency with the given scenario was given through using the original dump, the major problem was the inconsistent manipulation of the given data structures. This led to the detection of the forgeries. The analysts mainly used volatility, some kind of hexeditor (hexedit, hexdump, GHex, wxHexEditor, bless) and some plain-text data searching tool (grep, strings).

5. Results of repetitive detection experiment

5.1. Context and participation

Part C of the experiment was performed as part of the forensic

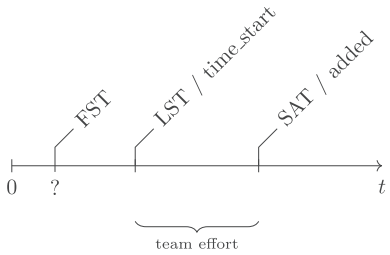

Fig. 6. Definition of *team effort* per challenge based on timestamps in database: *first shown time* (FST) was not recorded, therefore we used *last shown time* (LST) and *submission added time* (SAT) to approximate team effort.

rodeo during the DFRWS EU 2019 conference in April 2019 in Oslo, Norway. During the event, the participants had to group into teams and received a sequence of "alternative" memory dumps of Werner Weber's laptop that corresponded to different things that could have happened in the past. These memory dumps were the elements of the random pool created during Part A and individually analyzed in Part B.

The memory dumps were presented to the teams as a sequence of challenges via a mellivora capture-the-flag platform running on a webserver (Nakiami). During the experiment, 40 challenges had to be solved sequentially which were drawn randomly from the pool. The challenges were numbered from 1 to 40. For every challenge, the team needed to give one of two answers:

- YES: Means that the image is a forgery, i.e., traces of network connections to the server are forged evidence
- NO: Means that the image is an original, i.e., traces of network connections to the server are true evidence

The teams got immediate feedback whether their answer was correct or not. The participants were also told to record a brief (1000 characters) justification of their answers. To increase motivation, correct answers scored points and a prize was set out for the winning teams.

Overall 15 teams with 66 members participated in Part C of the experiment. Within a pre-study online questionnaire, most of them stated that they were average experienced professionals or academics. The experiment started at 7 pm and ended at 8:45 pm, thus the teams had 105 min to solve the challenges.

5.2. Definition of effort

Originally we had planned to make the effort used to analyze and detect forgeries comparable between experiment parts B and C. For this, we defined the following timestamps (see Fig. 6, all timestamps were measured relative to the begin of the experiment):

- The *first shown time* (FST) defines the point in time when the challenge was shown the first time to the team.
- The *submission added time* (SAT) defined the point in time that the result of the challenge was submitted.

Unfortunately, the database did not record the first time that the challenge was shown to the team but the final timestamp of the challenge being shown to the team which we defined as the *last shown time* (LST). We defined the time needed by the team to solve a challenge as the difference between LST and SAT for that challenge and called this the *team effort* of that challenge for a particular team. The *overall team effort* is the sum of all team efforts per team over all challenges.

Table 4
Results of team 4, 5, 9, 10, 11 of Part C of the experiment (team effort given in seconds).

	classified as original	classified as forgery
original	84	8
team effort range	0–2610	3–643
team effort average	198.89	162.75
team effort median	133	173
forgery	16	75
team effort range	2–182	3–1852
team effort average	52.81	192.05
team effort median	8	123

While we recorded the number of team members per team, we did not prohibit teams from requesting and solving challenges in parallel. We therefore could not reconstruct the number of team members who actually worked on a particular challenge at a given point in time. Therefore, the team effort per challenge does not imply any person effort per challenge. Measurements of effort between parts B and C of the experiment were therefore incomparable.

In our analysis, we restricted ourselves to the results of teams that took part during the full time of the experiment (some teams started late and some stopped to submit answers early), that solved sufficiently many challenges and that clearly were not solving most of the challenges by guessing. This resulted in data from five teams (teams 4, 5, 9, 10 and 11).

Team 4 turned out to be the winning team scoring most points in the smallest time. Overall, the five teams had 23 team members and analyzed 183 challenges until the end of the experiment.

5.3. Tampering success (part C)

Table 4 shows the results from the teams as a confusion matrix. Similar to the results of Part B of the experiment (see Table 2), the participants of Part C also had a very high correct detection rate: 159 out of 183 challenges were correctly solved (84 out of 92 originals and 75 out of 91 forgeries). Conversely, 8 originals were classified as forgeries and 16 forgeries were classified as originals. Therefore, twice as many forgeries were falsely classified than originals. The team effort values are given for information purposes only. In general, the analysis effort seems to be lower for the group of professional analysts than for the student group. This could be due to the fact that the group of professionals did not had to write a report, but also to greater experience.

5.4. Success of tampering approaches

We now have a look at those forgeries that were incorrectly classified as originals. These were challenges numbered 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 34, 35, and 38 (without double false negatives).

Four of those challenges belonged to the Simulate-and-Sweep class, four of them belonged to the Manual-Fix class and six to the Manual-Fix++ class. Interestingly, one of the Simulate-and-Sweep class forgeries was the same one that was also falsely classified as original in Part B of the experiment. In contrast, the wrongly classified Manual-Fix++ class forgery from Part B of the experiment was not among the wrongly classified forgeries in Part C.

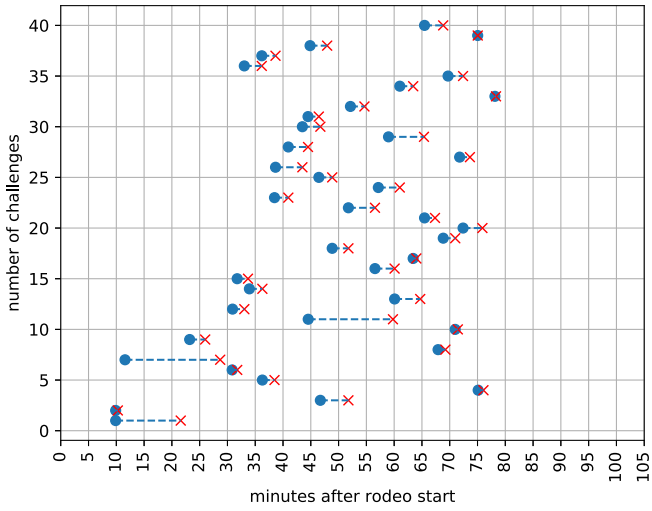

Fig. 7. LST and SAT times per challenge for team 4. LST is depicted as blue circles and SAT is depicted as red x.

5.5. Evidence of parallel processing and guessing

Even though sequential processing of challenges was mandated by the rules of the experiment, it was not technically enforced. The effects of parallel processing of challenges clearly shows up in the data. For example, Fig. 7 plots the LST and SAT values for all challenges over the course of the experiment, exhibiting a large amount of parallel processing to optimize the team effort (team 4 had 3 members). In contrast, Fig. 8 shows the mostly sequential challenge processing of team 11, which resulted in considerably more time to solve all challenges than team 4.

One major difficulty in evaluating the team results stems from effects of guessing. Fig. 9 shows the distribution of the correct and incorrect answers given by team 10 over the time of the experiment. This clearly shows that the closer the experiment came to an end, the more random the results of the challenges were. This is in contrast to a similar plot for the winning team 4 (see Fig. 10) that shows no sign of guessing. Because of their excellent strategy and the effects of parallel processing, team 4 had much more time to solve the challenges than the other four teams and gave only one

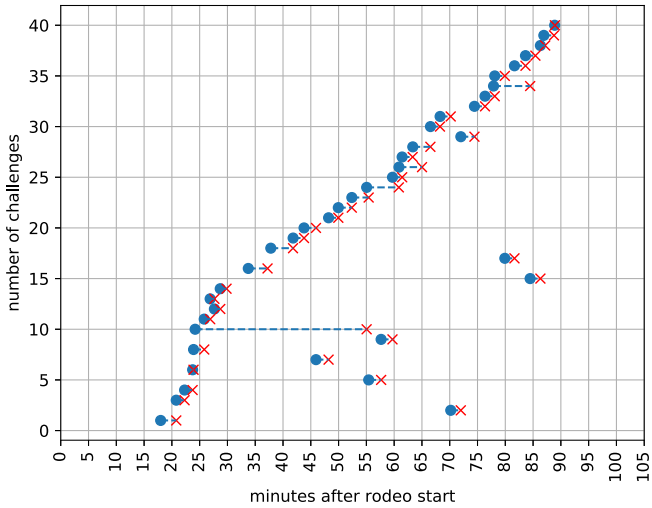

Fig. 8. LST and SAT times per challenge for team 11. LST is depicted as blue circles and SAT is depicted as red x.

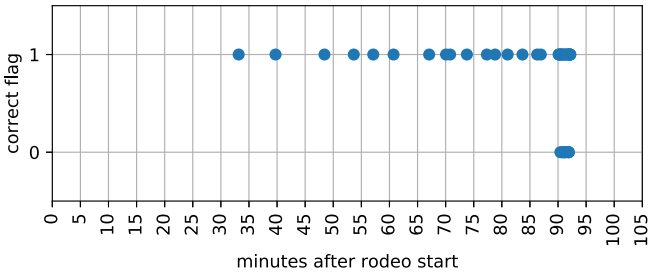

Fig. 9. Distribution of correct and incorrect answers over time for team 10.

incorrect answer at the beginning of the experiment. Even this answer ("7365" instead of "YES" or "NO") appears to be an initial test answer.

The two above observations convinced us that the definition of the effort for Part C of the experiment is incomparable to that of Part B.

5.6. Effect of repetition

One of the major research questions in Part C of the experiment was to study the effect of the repetition. Fig. 11 and Fig. 12 shows the evolution of the team effort per challenge over time for teams 4 and 9. They show that team 4 apparently quickly developed a solving strategy while team 9 clearly showed an improvement of team effort over the time of the experiment.

5.7. Detection approaches

After the experiment, only teams 4 and 10 reported their detection approaches. Both teams stated that they had used volatility to analyze the main memory dumps. They mainly used the three volatility modules linux_netstat, linux_netscan, and linux_bash. Both used a similar detection strategy, which team 4 described as follows:

- For `linux_bash`: "We were looking for a ssh connection (with the id admin, port 43023 (not 22 anyway))."
- For `linux_netstat`: "We looked if there was an active connection to the IP address 131.188.31.249."
- For `linux_netscan`: "When it worked, we were checking if the results were consistent with the results found with `linux_netstat`."

Team 10 stated that they used `linux_netstat` to detect active connections and additionally analyzed the process list.

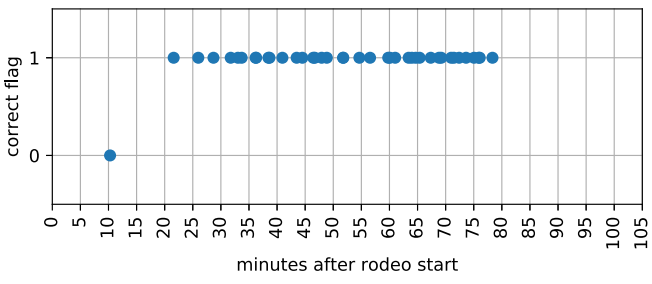

Fig. 10. Distribution of correct and incorrect answers over time for team 4.

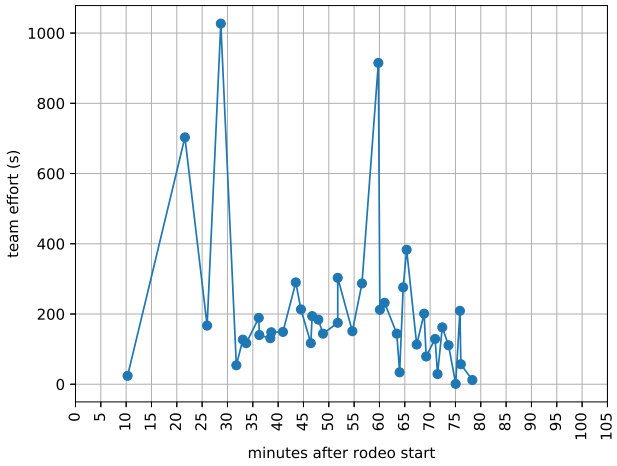

Fig. 11. Team effort per challenge for submission added time (SAT) for team 4. Effort is measured in seconds.

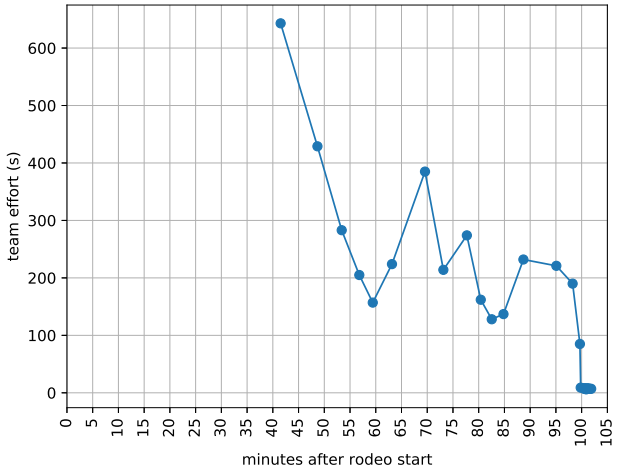

Fig. 12. Team effort per challenge for submission added time (SAT) for team 9. Effort is measured in seconds.

6. Summary and conclusions

We performed a sequence of experiments to understand the effort and factors influencing the tampering with digital evidence. The domain of evidence consisted of main memory images instead of hard disc images that had been investigated in previous work (Freiling and Hösch, 2018). Next to the production of forgeries (Part A), the experiment consisted of an individual detection experiment (Part B) and a repetitive team detection experiment (Part C).

Overall, the results confirmed the findings on the hard disc tampering experiment, namely that tampering is hard in the sense that the probability that a manipulation is detected is high. However, tampering with main memory dumps appears to be an entirely different playing turf than hard disc images as can be seen from the strange linear regression trend that relates manipulation effort to tampering effort. This trend was inverse to what has been seen in the work on hard disc tampering (Freiling and Hösch, 2018) and what can generally be expected.

The different tampering approaches were similar to those documented by Freiling and Hösch (2018) for hard disc images, where Manual-Fix appears to be better than other approaches to slow down analysts, but detection success is obviously also influenced by experience and training of the analyst.

Unfortunately, due to the small numbers of participants our results are not statistically significant but they can be regarded as qualitative insights into different forgery approaches. The repetitive detection experiment was an attempt to increase the number of participants, but unfortunately the circumstances of the experiment left many variables uncontrolled. To execute a tampering experiment with more participants in a controlled environment in the future, it is necessary to determine a suitable and realistic tampering task first. Experimenting with other situations of tampering is left to future work.

The experimental data of part B and C of the experiment can be downloaded at https://faui1-files.cs.fau.de/public/publications/SWF20-DFRWS-EU-2020-Data.xlsx for further analysis. All data regarding the memory images (originals and forgeries) is available online at https://www.cs1.tf.fau.de/dfrws-eu-2019-forensic-rodeo/ for future exercises. The ground truth is available upon request to instructors from the authors.

Acknowledgments

We wish to thank all participants of the experiment and the DFRWS 2019 Forensic Rodeo for their patience and participation. We also thank Mattia Epifani for his support in the preparation of the DFRWS 2019 Forensic Rodeo, and Jens Schlumberger, Andreas Dewald and the anonymous reviewers for their helpful comments on previous versions of the paper. 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/GRK2475/1–2019).

References

Brenner, S.W., Carrier, B., Henninger, J., 2004. The Trojan Horse Defense in Cybercrime Cases', Santa Clara High Technology Law Journal, vol 21, 1. http://digitalcommons.law.scu.edu/chtlj/vol21/iss1/1.

Caloyannides, M.A., 2003. ‘Digital “evidence” and reasonable doubt’. IEEE Security & Privacy 1 (6), 89–91. https://doi.org/10.1109/msecp.2003.1266366.

Carrier, B., 2005. File System Forensic Analysis. Addison-Wesley.

Dardick, G.S., Endicott-Popovsky, B., Gladyshev, P., Kemmerich, T., Rudolph, C., 2014. Digital evidence and forensic readiness (dagstuhl seminar 14092). Dagstuhl Reports 4 (2), 150–190. http://drops.dagstuhl.de/opus/volltexte/2014/4549.

Davidoff, S., 2008. Cleartext Passwords in Linux Memory. Massachusetts institute of technology, pp. 1–13.

Flynn, M., 2019. A florida Cop Planted Meth on Random Drivers, Police Say. One Lost Custody of His Daughter. July 11 (The Washington Post).

Freiling, F.C., Hösch, L., 2018. Controlled experiments in digital evidence tampering. Digit. Invest. 24 https://doi.org/10.1016/j.diin.2018.01.011. S83–S92.

Harris, R., 2006. Arriving at an anti-forensics consensus: examining how to define and control the anti-forensics problem. Digit. Invest. 3 (Suppl. 1), 44–49.

Johnson, M.K., Farid, H., 2006. Exposing digital forgeries through chromatic aberration. In: Voloshynovskiy, S., Dittmann, J., Fridrich, J.J. (Eds.), ‘MM&Sec’, ACM, pp. 48–55. https://doi.org/10.1145/1161366.1161376.

Kumar, V., Srivastava, J., Lazarevic, A., 2006. Managing Cyber Threats: Issues, Approaches, and Challenges, Massive Computing. Springer US. https://books.google.de/books?id=zTEoHdW9qDQC.

Ligh, M.H., Case, A., Levy, J., Walters, A., 2014. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. John Wiley and Sons.

Lin, X., 2018. Introductory Computer Forensics: A Hands-On Practical Approach. Springer. https://books.google.de/books?id=thh5DwAAQBAJ.

Lin, Z.C., He, J.F., Tang, X., Tang, C.K., 2009. Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recogn. 42 (11), 2492–2501. https://doi.org/10.1016/j.patcog.2009.03.019.

Moch, C., 2015. Automatisierte Erstellung von Übungsaufgaben in der digitalen Forensik. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, Technische Fakultät.

Nakiami, [n.d.]. Github - nakiami/mellivora: mellivora is a CTF engine written in PHP. https://github.com/Nakiami/mellivora.

Paul, G.L., 2009. Foundations of Digital Evidence. Amer Bar Assn.