Gradually Improving the Forensic Process

Sebastian Neuner*, Martin Mulazzani*, Sebastian Schrittwieser† and Edgar Weippl*
*SBA Research
Email: {sneuner|mmulazzani|eweippl}@sba-research.org
†FH St. Pölten, Austria
Email: Sebastian.Schrittwieser@fhstp.ac.at

***Abstract*—At the time of writing, one of the most pressing problems for forensic investigators is the huge amount of data to analyze per case. Not only the number of devices increases due to the advancing computerization of everydays life, but also the storage capacity of each and every device raises into multi-terabyte storage requirements per case for forensic working images. In this paper we improve the standardized forensic process by proposing to use file deduplication across devices as well as file whitelisting rigorously in investigations, to reduce the amount of data that needs to be stored for analysis as early as during data acquisition. These improvements happen in an automatic fashion and completely transparent to the forensic investigator. They furthermore be added without negative effects to the chain of custody or artefact validity in court, and are evaluated in a realistic use case.**

***Keywords*-digital forensics, forensic process, file deduplication, file whitelisting**

I. Introduction

One of the major problems in digital forensics, and faced by investigators around the world on a daily basis, is the vast quantity of data to analyze in an average case. Commodity 3.5" SATA hard drives come with a maximum capacity of up to 8 terabytes per hard drive, while memory cards for smartphones and digital cameras can have up to 256 gigabytes. USB thumb drives have a current maximum capacity of two terabytes. This is expected to increase even further in the near future. Combined with the average number of devices per household, this can leave a forensic expert with tens of terabytes of data to not only acquire in time, but also to process and store securely. Even though this was predicted since before 2010 to become one of the challenges in our field [13], little has happened so far to adapt the forensic process as well as the recommendations for investigators on how to handle such vast amounts of data.

As such, this paper aims to enhance standards and recommendations like the RCF 3227 [7] or NIST SP-800-86 [16] to incorporate improved analysis techniques which reduce the amount of overhead, especially in the overall storage capacity needed during an investigation, but also to allow for automated, standardized analysis steps to reduce the manual workload for investigators. These techniques are most often not necessarily new, but have not yet received sufficient attention in the forensic literature and the forensic process.

The contributions of this paper are as follows:

- We propose an advanced forensic process for digital investigations, taking into account some of the most pressing limitations for investigators.
- We discuss different analysis techniques which scale well and can be used to limit backend storage requirements for analysts.
- We evaluate our process with an exemplary use case, and show that the overall storage requirement for that case can be decreased by 78%.

The remainder of this paper is organized as follows: Section II gives a brief background on the forensic process as well as existing guidelines for data acquisition and processing. Section III explains our proposed changes and improvements of the forensic process, which is then evaluated with an exemplary use case in Section IV. We discuss the results and limitations in Section V, before we conclude in Section VI.

II. Background

Due to the ever increasing computerization of the modern lifestyle, the number of devices as well as the individual storage capacity increases in forensic investigations. Two commonly referenced standards for the forensic process are RCF 3227 [7] as well as NIST SP-800-86 [16]. Both specify, among other things, the order of volatility for the importance of acquiring data in the correct sequence, the chain of custody for seamless documentation during acquisition and rigorously during data analysis, and numerous techniques for acquiring, processing and analyzing data of all kind. The increasing variety of device types, file systems and usage patterns, e.g. recently with the success of centralized online social networks or with cloud computing, keeps the forensic analyst furthermore constantly challenged.

NIST SP-800-86 recommends to use a *working copy* and to make a *backup copy* in case data gets accidentally written to the images or the images get tainted during analysis. This can put a heavy burden on analysts since they need to have twice the storage capacity for each case at hand at the time of acquisition, and for storage afterwards for as long as the investigation lasts. The acquisition usually results in copying the data twice, and verify integrity before and after each step. For a commodity multi-terabyte hard drive, this can take easily between 15 and 20 hours, just to complete one of these steps, and even worse for large-capacity RAID systems.

While the problems of digital forensics and the forensic process in particular are constantly discussed in the literature [26] [13], the amount of manual work is still not even close to an area where many investigators would like to have it. The vast spectrum of available software, a plethora of data formats and different devices and device types make it hard for fully automated approaches [10], even though commercial software often promises exactly that. With open source tools like *bulk_extractor* however, which is an automated open-source feature extractor, the community is heading in the right direction of fully automated analysis [14], as *bulk_extractor* scales to use the full capacity of the underlying machine, works without manual interaction once it is started and can be easily extended with plugins.

III. Improvements to the Forensic Process

In the first part of this section we explain techniques which can be leveraged to cope with the ever increasing case sizes, whereas we put them in the context of a case in the second part, especially where to apply them with respect to the existing recommendations.

A. Individual Improvements

The first proposed enhancement of the forensic process is already rigorously lived in practice: **not always are two physical copies needed**. While a working copy and a backup copy should always be in place, less stringent rule enforcement is needed when the data source drives do not need to get back to their owners or into production immediately. This is for example particularly the case for investigations by law enforcement, where the data sources itself are confiscated and no pressure exists to return them to the owner immediately. We do not have concrete numbers on how this is done in practice, but this can be very effective for reducing storage requirements. It is more like a logical enhancement to the standard processes, since it is not in all cases that the data needs to go back to production systems as soon as possible. Of course, for production systems where downtime is an issue and hard constraints exist that these systems stay online, a second copy is needed for backup. This is of relevance for e.g. all kind of server systems like mail servers or web servers. Sometimes it can be also enough to create an image of the current files in the file system, omiting the free space and possible file slack. This depends on the context of the investigation, and the actual questions to be answered.

Another strategy which is missing so far in the process descriptions is the rigorous use of **file whitelisting**. Files irrelevant to the investigation can be easily excluded in the early stages due to the use of cryptographic hash functions like MD5 or SHA-1, whereas files of particular interest can be identified if they are known a-priori to the investigator. In the forensic community, the most notable example for the former case is the NIST national software reference library (NSRL) with their reference data set (RDS) [20]. It uses default software installations of operating systems and end user software to derive a list of hash values on a file basis. The most recent version of the RDS 2.47 (as of December 2014) contains a total of 40 million hash values, for a total of close to 150 million files. An example for the latter is PhotoDNA, which computes a visual fingerprint for pictures and compares it with known pictures of child abuse. It was developed by Microsoft and Dartmouth University, and is used by large software companies like Facebook or Twitter. Most recently, a REST API was introduced to query the PhotoDNA database online[1]. Due to the availability of cheap storage and processing power, we argue that any investigator could and should set up their own list of hash values for files of interest. This could include all files from intra-company file shares, possibly malicious files from anti-virus quarantine, web pages (including pictures and thumbnails) or company-wide email attachements. Depending on the local privacy laws there are hardly any limitations on which files to include.

The improvement on the storage backend which this paper proposes is the creation of a **reduced working copy**. It is created as soon as all known, benign files are identified, as they can be safely excluded from the need to store them (except for its metadata). All other files are stored according to the file system metadata, and additionally all portions of the free space are extracted and stored as well. At worst, this can be a very large fraction of the original drive capacity. At best, a vast majority of files can be excluded in a fully automated process and without any interaction of the investigator needed. Since this process is strictly monotonous (the resulting working copy can only be at most the capacity of the drive), the resulting working copy will always be smaller than the full capacity of the storage drive. All further analysis steps can be done on this reduced working copy, and the original drive(s) can be locked securely away as the backup. If the drive(s) need to go back into production use, a second copy is to be created using a bitwise copy. The second large improvement on the storage backend is the rigorous use of deduplication, at the very least across devices within each case. This step should also include the application of **fuzzy hashing** [29], since files which are similar but not the same until the very last bit cannot be identified using cryptographic hash functions. While the most commonly found fuzzy hash functions are *ssdeep* [17] and *sdhash* [28], there is still no common ground which is the best for specific use cases, and specialized similarity hash functions are still an active field of research [6], like for example *mrsh-v2* [5] which can identify file fragments.

Hashing each file per device by default can be used to **identify the same files across devices** easily, and reduce the need for storing them multiple times. This is likely to further reducing the number of files that need inspection of any kind, and save storage at the investigators backend due to deduplicate. In particular with the use of cloud storage solutions like Dropbox or iCloud, many devices nowadays share local files which are kept synchronous across devices.

[1] Online at http://www.microsoft.com/en-us/photodna

However, the file system metadata of all copies needs to be preserved. Across cases and in the near future, efficient and privacy-preserving mechanisms will be needed to share hash value lists between multiple parties. Even though there are current mechanisms available to facilitate private set intersection [12], i.e. using zero-knowledge proofs [8], it is not yet known if they can be used for digital forensics and handle millions of hash values in practice. File system- as well as enhanced analysis of file **metadata** should be used in this step to compare file timestamps, EXIF metadata or other information sources, to identify data sources and sinks and to reconstruct the flow of information across devices (and users). In very large environments with thousands of computers and users, this can be challenging.

Finally, the process should include the acquisition from various **online accounts** and the retrieval of the associated data and metadata using forensic methods. Online services like Facebook, Twitter, Apple or Google Services have hundreds of million of users, and these online accounts are often tied to smartphones. While these companies have mechanisms in place to aid law enforcement, this source of information is not available for foreign civil law suits or other third party investigations. Even though there have been recently proposed approaches to acquire the data without the explicit aid of the service operators, e.g. using APIs [15] or based on observed network traffic [11] [22], they have yet not been incorporated in the standard processes. Cloud computing [2] can pose another, although related type of problem for digital forensics. Compared to online services and SaaS platforms, acquisition in IaaS cloud services is more related to the standard forensic process in direct comparison [19]. An obstacle is often how the investigator can access these services, and whether or not the credentials needed for authentication can be obtained from the suspects, the hard drives or by other means. In most cases user consent is needed, and even though Great Britain is among the few countries that can convict a suspect if he/she is not releasing a password, this is not commonly found elsewhere. And even if the user consents, these APIs are (and probably never will be) similar to allow the creation of a uniform forensic interface.

B. Improved Forensics Process

Our improved steps for automated data analysis so far only enhance the current standards, in particular NIST SP-800 86. While RFC 3227 stops after the data acquisition, NIST SP-800 86 states specific steps to reduce the amount of files and data to analyze, i.e. using the NIST NSRL hash value collection. However, fuzzy hashing and cross-device checking are not mentioned, as well as the importance of online accounts for data storage and online services. It only exemplifies the use of multiple sources for data gathering, within a confined scope.

The core improvement in this paper is the parallelized calculation and evaluation of hash values, and the reduced working copy. As before, the data should be acquired according to the order of volatility, and using a hardware write blocker in place to prevent manipulations (accompanied by rigorous documentation). Before the image is created, file system metadata is parsed and all files in the file system hashed numerous times, including cryptographic hash functions like SHA-1 and fuzzy hash functions like *ssdeep* or *sdhash*. These hash values are then stored in some form of database and automatically evaluated with the proposed improvements: known, benign files are excluded using e.g. the NIST NSRL dataset, and multiple copies of the same file are detected across devices. Similarity hash values are used to detect similar files and present a set of candidates that seem related. This information can be embedded and enriched within an automatic timeline creation from file system metadata in the acquisition steps. Deleted files where the data has not yet been overwritten should be extracted and hashed similar to the other files. Furthermore, known malicious files can be found using hash value black listing. In the future, additional hash value calculations can be added as well as additional hash value sources. This can include novel fuzzy hash functions, other cryptographic hash functions like the upcoming SHA-3 or new hashing methodologies like sector hashing as proposed in [34].

After the automatic exclusion of files, the remaining files, folders and regions of free space are copied into the reduced working copy. Depending on the context of the analysis, this is expected to be sufficient for many cases. The use of cryptographic hash functions allows the argumentative exclusion of known files, since for each and every file there is a line of argumentation why this file was removed and ignored in further analysis steps. The final step is the optional extraction of online credentials from browsers, stored passwords or artefacts from online data services like e.g. Dropbox or iCloud. The entire process is visualized in Figure 1. Please note that the individual processing steps can be run concurrently: hashing the files may happen on the same byte stream as extracting the file system metadata, thus reducing the amount of read requests to the hard drive to the original bytewise copy as used today in digital forensics. Also, the extraction of online account information is considered optional, thus the different representation in the figure.

Most importantly, all of the steps discussed so far have the ability to run automatically and present their findings in an understandable format to the investigator as well as in machine readable form for further analysis steps. The computational overhead is very likely to be negligible compared to the additional insights using automated analysis as well as the possible reduction in the number of files and file fragments needing manual inspection.

IV. Evaluation

Since there are no common grounds or standardized use cases regarding digital forensic investigations, we exemplify our improvements in our evaluation with the scenario described in the following. We believe that the described scenario represents a large fraction of use cases, and exemplifies our

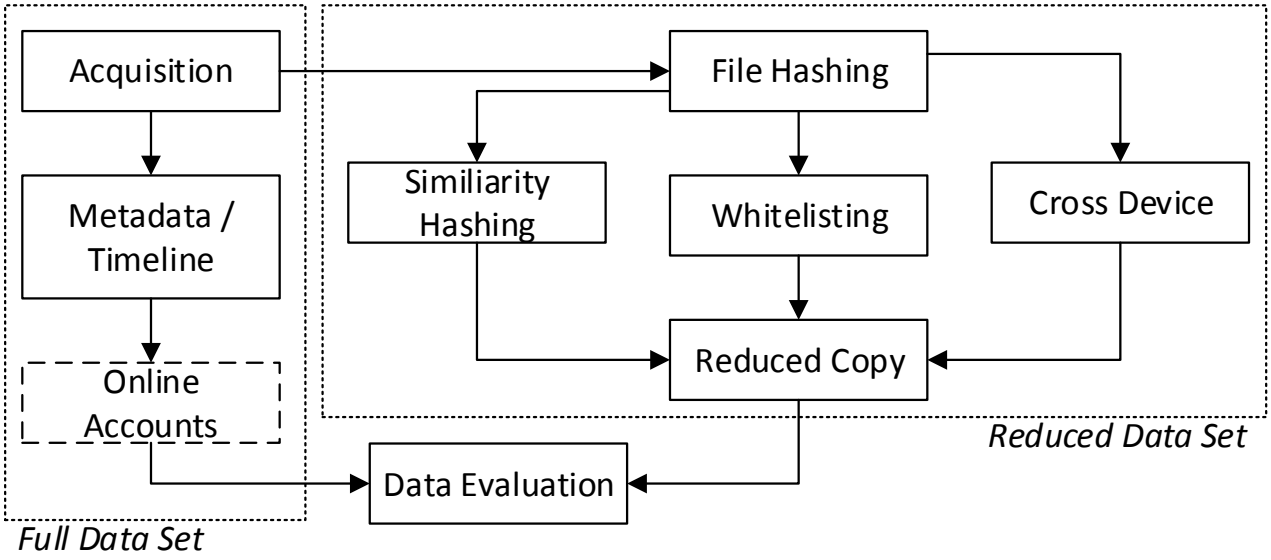

Fig. 1. Improved Steps for the Forensic Process

improvements. Please note that the specifics we used in this scenario may seem somewhat arbitrary, but they were derived during our ongoing informal discussions with law enforcement officials as well as forensic investigators. We believe that this scenario aims in the right direction of the current standard use case for forensic investigations and can be considered a valid, although biased sample for the typical kind of case with a rather high prevalence in the field.

A. Design of our evaluation

We consider some form of malicious online activity as the initial reason for an investigation. The investigator is tasked with the acquisition of a relatively small number of devices from the following set, all devices which can be found in a modern household: computers or notebooks, smartphones respectively tablets, external storage devices like USB thumb drives or external hard drives, and lastly digital cameras. Furthermore numerous accounts at online services, e.g. Facebook, Google, Flickr or Twitter (just to name a few), but omitted for brevity in our evaluation.

For our evaluation we used the following setup. We consider the investigated user to have the following devices in use: Two computers, whereas one computer is a Desktop PC and one computer is a Laptop. The windows PC is based on Windows 8 which uses roughly 160,000 files. We consider an additional total of 50,000 files to be from the user, temporary working files and installed software. As described by Rowe et al. [31], commonly found hard drives include 18% Microsoft related system files, 25% graphics (e.g. camera images), 4.7% documents (e.g. spreadsheets, presentations, etc.) and 4.3% executables to name the most important types. For mobility reasons the user has a Laptop computer with files daily mirrored with the Desktop computer and therefore these corpus' share 80% of the same files. The user uses an Android smartphone with about 13,000 multimedia files such as images, photos, videos and music files as described by Lessard et al. [18], 2000 files which are related to different installed Apps (assuming about 300 files per App) and 20,000 files which are either related to running Google services or related to the Android operating system itself.

In addition to the mentioned computation devices (computers, smartphone) the specified setup includes two digital cameras with 2000 photos in total, splitted across three SD cards and several external storage devices used for backup. Those external storage devices include two external hard drives with half a terabyte and one terabyte in capacity, and three USB thumb drives from various manufacturers and with different capacities. These external hard drives contain the backed up files from the Desktop PC as well as the notebook, respectively. The Desktop PC was used for backing up the files from the cameras and the smartphone, meaning that these files are found in the backup on the one terabyte hard drive as well. The USB thumb drives include an additional 20,000 files which are unique with respect to the other devices. Finally data is spread over the computers and the smartphone via a cloud service (e.g. Dropbox) and kept in sync with a remote copy. Therefore a large number of the user files are available on Desktop PC, Laptop as well as the smartphone. Please note that most of the specific numbers were chosen at random, they would differ in reality due to the different user's age distribution, usage patterns, personal preferences, professional background and other factors. The overall capacity and number files for each device can be seen in Table 1.

Device	Storage Capacity	# of files	used capacity
Windows 8 PC	1tb	210k	250gb
Windows 8 Notebook	500gb	190k	180gb
Android Smartphone	32gb	35k	15gb
SD Cards	$\{8\|8\|16\}$gb	2k	10gb
external hard drive	$\{500gb\|1tb\}$	400k	430gb
USB thumb drive	$\{4\|8\|16\}$gb	20k	32gb
Sum:	**3.16tb**	**857k**	**917gb**

TABLE I
OVERVIEW OF DEVICES AND STORAGE CAPACITIES

B. Evaluation of the improved forensic process

The regular forensic process would need to acquire and copy each device at least once, resulting in the need to store roughly a little more than three terabytes of data only for the device images. If a backup copy is needed, this adds up to 6.2 terabytes of storage capacity needed. Overall, this would also mean extracting and analyzing 857,000 files. In the improved forensic process, however, an overall list for the entire case and thus all the devices is created which contains file names and hash values of all the unique files. This list also includes all metadata for the deduplicated files. To further reduce the number of files to be extracted for the working copy, the content is compared to available software reference lists like the NSRL. These steps allow to drastically reduce the numbers. The first reduction is caused by having redundant device contents on multiple devices. The Desktop computer and the Laptop both have their backup exclusively on their external USB backup drives. 80% of the Laptop user files (30,000) are duplicates from the Desktop PC, leaving 20% or 6,000 unique files as difference between the user files on the Desktop PC and the Laptop (due to cloud sync and working copies). As such, and starting with the acquisition on the Desktop PC, 210,000 files are to be extracted from the Desktop PC while the acquisition of the Laptop deduplicates the operating files and most of the user files. Therefore 184,000 files are duplicates and not added to the reduced working copy.

One cloud service is in use which synchronizes files over the Desktop PC, the Laptop computer as well as the smartphone, including the pictures of the user. A typical Desktop PC contains about 7.6% camera images as of [31], which would be in this particular case roughly 16,000 pictures on the PC. This is a superset of the pictures from the smartphone, including the audio and video files outside of the synced folders, leaving 22,000 files on the smartphone to be included in the reduced working copy. The 2,000 pictures on the digital cameras (stored on three different SD cards) were already synced to the Desktop and are thus duplicates in this case.

The last step is the removal of commonly found files e.g. using the NIST NSRL RDS. According to Rowe [30], 32% of a typical hard drive can be matched with files contained in the RDS set. This reduces the 210,000 files from the Desktop PC to roughly 143,000 files, and the files uniquely found on the Laptop to 4,080 files. Table II illustrates files to be extracted per source in the corpus. Grayscale areas mark the proportion of files that have to be extracted from that specific source, whereas white areas are duplicates that do not have to be taken into account for the created reduced working copy.

V. RESULTS

Considering the use case described in the previous section, the reduced working copy will finally contain roughly 189,000 files. This means a reduction in the number of files by 78% compared to the full dataset. On average, each file in our case has 1.1 megabytes, which means an overall storage reduction in size for the working copy of 709 gigabytes. This reduction can be projected on several other factors which are increased such as time, CPU requirements and memory consumption. However, those calculated numbers highly depend on the underlying data, which means a bigger amount of commonly found and duplicated data can lead to an even higher reduction in disk space consumption. The percentages of the overall reduction in files and storage space are represented in Figure 2.

A. Discussion

The improvements described above are totally transparent to the forensic analyst and can be applied during the data acquisition in an automated fashion. Therefore we argue that our proposed improved forensic workflow can be beneficial for the vast majority of forensic cases, and can be applied very easily to existing tools and workflows. However, results highly depend on the data that has to be evaluated. A case containing a high number of unique files, distributed over several devices and cloud services, is unlikely to get an improvement factor as above. If the number of unique files is low, or the unique files are large in size, the resulting improvement will be lower. Since there are no studies in the literature describing the uniqueness and distribution of files in a typical forensic investigation (or user environment per se), it is not possible to determine a specific number. The overall reduction also depends on the used whitelist, and while the NIST RDS is a good start, it might be beneficial to leverage additional sources for file whitelisting.

B. Related Work

While commonly found file systems like FAT32, NTFS [9] or HFS+, as well as new-and-upcoming ones like ReFS [32] do not support in-place data deduplication, ZFS [4] and btrfs [27] already support deduplication. Btrfs for example supports batch deduplication with tools available online[2] and eventually will implement in-band deduplication. This is similar to our proposed cross-device data deduplication, but has two major drawbacks: for one it requires that the analysis platform is able to work with btrfs or ZFS, which are currently only supported by Linux. Both deduplicate with a fine sub-file granularity. While it has been shown recently that the overlap of sub-file hash values is fairly small [34], the use of strong cryptographic hash functions on the entire

[2] *duperemove*, online at https://github.com/markfasheh/duperemove

Desktop PC	68%
Laptop	2%
external USB devices	5%
SD card / camera	
Cloud	
Smartphone	63%

TABLE II
FILE EXTRACTION DISTRIBUTION PER SOURCE IN CORPUS.

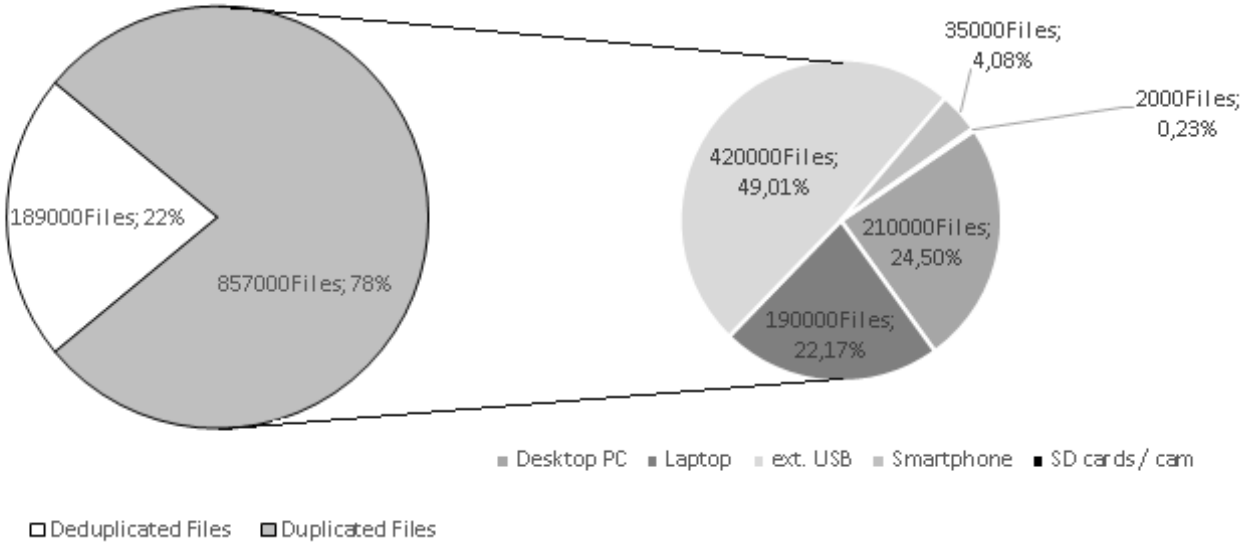

Fig. 2. File reduction in the reduced working copy

file has the benefit of using existing file whitelists like the NIST RDS. For btrfs, the process of deduplication so far has to be started manually by the analyst. As such, our proposed incorporation of data deduplication in the forensic process can be considered a generalization in comparison to those file systems.

Numerous forensic models have been discussed in the literature so far, and an ad hoc overview is given in [25]. While the initial work in [3] was superseded by the forensic standards by IETF and NIST, [24] defined a set of qualities that should be part of any forensic model. Related to the recent success of online social networks, various approaches have been presented to allow data acquisition [15] [1] and visualization of social networking data [21]. Visualization in general received a lot of attention in the ”golden age of digital forensics” (coined by Simson Garfinkel), and was considered an important stepping stone in the field of semi-automated forensic analysis [33], [23].

C. Limitations

The limitations of our approach lie in the unknown general interpretation of the forensic processes and the forensic practice in the field. It is unclear if or how often forensic investigators create two distinct copies for an investigation, and whether or not file whitelisting is used in practice. Secondly, the underlying data is case-specific, and as such it is hard to calculate concrete numbers, and unfortunately our improved process can neither provide a lower nor an upper bound. However, and since computational power is rather cheap and even notebooks nowadays comprise of multi-core CPUs, the overhead is expected to be negligible.

Another limitation is that other common problems in digital forensic analysis are not touched by our approach. If data is encrypted, or entire hard drives are protected, there is no access to the data and as such our improved process is not applicable. The same is true for remote data if it is not accessible, it cannot be used for the investigation.

D. Future Work

After the presentation and a thorough discussion of our improvements we plan to standardize them in a RFC. This RFC can be an important enhancement for several use cases, e.g. to prove the line of action as a consultant to a customer, or to prove the standardized actions as an expert witness in

court. We also plan to find and describe standardized forensic use cases for further forensic research which are generally enough to cover the majority of use cases, but vague enough to be open for future problems. We also plan to implement specific steps of our process within open source toolkits as well as a standalone framework for file deduplication and for creating the reduced working copy.

VI. Conclusion

In this paper we showed how an improved forensic process can be used to reduce the amount of storage requirement for forensic investigations using file whitelisting and cross-device deduplication. While metadata of duplicate files has to be preserved, our process is particularly useful in cases where the focus of the investigation lies on referenced files in the file system. We described an exemplary use case where file deduplication and file whitelisting was used to save 78% of storage capacity, or 700 gigabytes. Overall we hope that our improved process will lead to interesting discussions in the community and an improved standard forensic process in the near future.

Acknowledgements

The research was funded by COMET K1, FFG - Austrian Research Promotion Agency and by FFG grant 846070: SpeedFor.

References

[1] N. Al Mutawa, I. Baggili, and A. Marrington. Forensic analysis of social networking applications on mobile devices. *Digital Investigation*, 9:S24–S33, 2012.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[3] V. Baryamureeba and F. Tushabe. The enhanced digital investigation process model. In *Proceedings of the Fourth Digital Forensic Research Workshop*. Citeseer, 2004.

[4] J. Bonwick. Zfs deduplication, 2009. URL: https://blogs.oracle.com/bonwick/entry/zfs_dedup.

[5] F. Breitinger and H. Baier. Similarity preserving hashing: Eligible properties and a new algorithm mrsh-v2. In *Digital forensics and cyber crime*, pages 167–182. Springer, 2013.

[6] F. Breitinger and V. Roussev. Automated evaluation of approximate matching algorithms on real data. *Digital Investigation*, 11:S10–S17, 2014.

[7] D. Brezinski and T. Killalea. Rfc 3227: Guidelines for evidence collection and archiving. *Internet Engineering Task Force*, 2002.

[8] J. Camenisch and G. M. Zaverucha. Private intersection of certified sets. In *Financial Cryptography and Data Security*, pages 108–127. Springer, 2009.

[9] B. Carrier. *File system forensic analysis*. Addison-Wesley Professional, 2005.

[10] A. Case, A. Cristina, L. Marziale, G. G. Richard, and V. Roussev. Face: Automated digital evidence discovery and correlation. *digital investigation*, 5:S65–S75, 2008.

[11] M. Cohen. Pyflag–an advanced network forensic framework. *Digital investigation*, 5:S112–S120, 2008.

[12] E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *Financial Cryptography and Data Security*, pages 143–159. Springer, 2010.

[13] S. L. Garfinkel. Digital forensics research: The next 10 years. *digital investigation*, 7:S64–S73, 2010.

[14] S. L. Garfinkel. Digital media triage with bulk data analysis and bulk_extractor. *Computers & Security*, 32:56–72, 2013.

[15] M. Huber, M. Mulazzani, M. Leithner, S. Schrittwieser, G. Wondracek, and E. Weippl. Social snapshots: Digital forensics for online social networks. In *Proceedings of the 27th annual computer security applications conference*, pages 113–122. ACM, 2011.

[16] K. Kent, S. Chevalier, T. Grance, and H. Dang. Guide to integrating forensic techniques into incident response. *NIST Special Publication (SP) 800–86*, 2006.

[17] J. Kornblum. Identifying almost identical files using context triggered piecewise hashing. *Digital investigation*, 3:91–97, 2006.

[18] J. Lessard and G. Kessler. Android forensics: Simplifying cell phone examinations. 2010.

[19] B. Martini and K.-K. R. Choo. An integrated conceptual digital forensic framework for cloud computing. *Digital Investigation*, 9(2):71–80, 2012.

[20] S. Mead. Unique file identification in the national software reference library. *Digital Investigation*, 3(3):138–150, 2006.

[21] M. Mulazzani, M. Huber, and E. Weippl. Social network forensics: Tapping the data pool of social networks. In *Eighth Annual IFIP WG*, volume 11, 2012.

[22] C. Neasbitt, R. Perdisci, K. Li, and T. Nelms. Clickminer: Towards forensic reconstruction of user-browser interactions from network traces. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1244–1255. ACM, 2014.

[23] J. Olsson and M. Boldt. Computer forensic timeline visualization tool. *digital investigation*, 6:S78–S87, 2009.

[24] S. Peisert, M. Bishop, S. Karin, and K. Marzullo. Toward models for forensic analysis. In *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on*, pages 3–15. IEEE, 2007.

[25] M. M. Pollitt. An ad hoc review of digital forensic models. In *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on*, pages 43–54. IEEE, 2007.

[26] G. G. Richard III and V. Roussev. Next-generation digital forensics. *Communications of the ACM*, 49(2):76–80, 2006.

[27] O. Rodeh, J. Bacik, and C. Mason. Btrfs: The linux b-tree filesystem. *ACM Transactions on Storage (TOS)*, 9(3):9, 2013.

[28] V. Roussev. Data fingerprinting with similarity digests. In *Advances in digital forensics vi*, pages 207–226. Springer, 2010.

[29] V. Roussev. An evaluation of forensic similarity hashes. *digital investigation*, 8:S34–S41, 2011.

[30] N. C. Rowe. Testing the national software reference library. *Digital Investigation*, 9:S131–S138, 2012.

[31] N. C. Rowe and S. L. Garfinkel. Finding anomalous and suspicious files from directory metadata on a large corpus. In *Digital Forensics and Cyber Crime*, pages 115–130. Springer Berlin Heidelberg, 2012.

[32] S. Sinofsky. Building the next generation file system for windows: Refs, 2012. URL: http://blogs.msdn.com/b/b8/archive/2012/01/16/building-the-next-generation-file-system-for-windows-refs.aspx.

[33] S. Teelink and R. F. Erbacher. Improving the computer forensic analysis process through visualization. *Communications of the ACM*, 49(2):71–75, 2006.

[34] J. Young, K. Foster, S. Garfinkel, and K. Fairbanks. Distinct sector hashes for target file detection. *Computer*, (12):28–35, 2012.