Data Corpora for Digital Forensics Education and Research

York Yannikos, Lukas Graner, Martin Steinebach, Christian Winter

Chapter 21

DATA CORPORA FOR DIGITAL FORENSICS EDUCATION AND RESEARCH

York Yannikos, Lukas Graner, Martin Steinebach and Christian Winter

Abstract Data corpora are very important for digital forensics education and research. Several corpora are available to academia; these range from small manually-created data sets of a few megabytes to many terabytes of real-world data. However, different corpora are suited to different forensic tasks. For example, real data corpora are often desirable for testing forensic tool properties such as effectiveness and efficiency, but these corpora typically lack the ground truth that is vital to performing proper evaluations. Synthetic data corpora can support tool development and testing, but only if the methodologies for generating the corpora guarantee data with realistic properties.

This paper presents an overview of the available digital forensic corpora and discusses the problems that may arise when working with specific corpora. The paper also describes a framework for generating synthetic corpora for education and research when suitable real-world data is not available.

Keywords: Forensic data corpora, synthetic disk images, model-based simulation

1. Introduction

A digital forensic investigator must have a broad knowledge of forensic methodologies and experience with a wide range of tools. This includes multi-purpose forensic suites with advanced functionality and good usability as well as small tools for special tasks that may have moderate to low usability. Gaining expert-level skills in the operation of forensic tools requires a substantial amount of time. Additionally, advances in analysis methods, tools and technologies require continuous learning to maintain currency.

In digital forensics education, it is important to provide insights into specific technologies and how forensic methods must be applied to perform thorough and sound analyses. It is also very important to provide a rich learning environment where students can use forensic tools to rigorously analyze suitable test data.

The same is true in digital forensics research. New methodologies and new tools have to be tested against well-known data corpora. This provides a basis for comparing methodologies and tools so that the advantages and shortcomings can be identified. Forensic investigators can use the results of such evaluations to make informed decisions about the methodologies and tools that should be used for specific tasks. This helps increase the efficiency and the quality of forensic examinations while allowing objective evaluations by third parties.

The paper provides an overview of several real-world and synthetic data corpora that are available for digital forensics education and research. Also, it highlights the potential risks and problems encountered when using data corpora, along with the capabilities of existing tools that allow the generation of synthetic data corpora when real-world data is not available. Additionally, the paper describes a custom framework for synthetic data generation and evaluates the performance of the framework.

2. Available Data Corpora

Several data corpora have been made available for public use. While some of the corpora are useful for digital forensics education and research, others are suited to very specific areas such as network forensics and forensic linguistics. This section presents an overview of the most relevant corpora.

2.1 Real Data Corpus

A few real-world data corpora are available to support digital forensics education and research. Garfinkel, *et al.* [7] have created the Real Data Corpus from used hard disks that were purchased from around the world. In a later work, Garfinkel [5] described the challenges and lessons learned while handling the Real Data Corpus, which by then had grown to more than 30 terabytes [5]. As of September 2013, the Real Data Corpus incorporated 1,289 hard disk images, 643 flash memory images and 98 optical discs. However, because this corpus was partly funded by the U.S. Government, access to the corpus requires the approval of an institutional review board in accordance with U.S. legislation. Ad-

ditional information about the corpus and its access requirements are available at [6].

A smaller corpus, which includes specific scenarios created for educational purposes [25], can be downloaded without any restrictions. This smaller corpus contains:

- Three test disk images created especially for educational and testing purposes (e.g., filesystem analysis, file carving and handling encodings).
- Four realistic disk image sets created from USB memory sticks, a digital camera and a Windows XP computer.
- A set of almost 1,000,000 files, including 109,282 JPEG files.
- Five phone images from four different cell phone models.
- Mixed data corresponding to three fictional scenarios for educational purposes, including multiple network packet dumps and disk images.

Due to the variety of data it contains, the Real Data Corpus is a valuable resource for educators and researchers in the areas of multimedia forensics, mobile phone forensics and network forensics. To our knowledge, it is the largest publicly-available corpus in the area of digital forensics.

2.2 DARPA Intrusion Detection Data Sets

In 1998 and 1999, researchers at MIT Lincoln Laboratory [12, 13] created a simulation network in order to produce network traffic and audit logs for evaluating intrusion detection systems. The simulated infrastructure was attacked using well-known techniques as well as new techniques that were specially developed for the evaluation. In 2000, additional experiments were performed involving specific scenarios, including two DDoS attacks and an attack on a Windows NT system. The data sets for all three experiments are available at [11]; they include network traffic data in `tcpdump` format, audit logs and filesystem snapshots.

The methodologies employed in the 1998 and 1999 evaluations were criticized by McHugh [16]. McHugh states that the evaluation results miss important details and that portions of the evaluation procedures are unclear or inappropriate. Additionally, Garfinkel [4] points out that the data sets do not represent real-world traffic because they lack complexity and heterogeneity. Therefore, this corpus has limited use in network forensics research.

2.3 MemCorp Corpus

The MemCorp Corpus [22] contains memory images created from several virtual and physical machines. In particular, the corpus contains images extracted from 87 computer systems running various versions of Microsoft Windows; the images were extracted using common memory imaging tools.

The corpus includes the following images:

- 53 system memory images created from virtual machines.
- 23 system memory images created from physical machines with factory default configurations (i.e., with no additional software installed).
- 11 system memory images created from machines under specific scenarios (e.g., after malware was installed).

This corpus supports education and training efforts focused on memory analysis using tools such as the Volatile Framework [23]. However, as noted by the corpus creator [22], the corpus does not contain images created from real-world systems or images from operating systems other than Microsoft Windows, which reduces its applicability. The creator of the MemCorp Corpus provides access to the images upon request.

2.4 MORPH Corpus

Several corpora have been created in the area of face recognition [8]. Since a large corpus with facial images tagged with age information would be very useful for multimedia forensics, we have picked a sample corpus that could be a valuable resource for research (e.g., for detecting of illegal multimedia content like child pornography).

The MORPH Corpus [20] comprises 55,000 unique facial images of more than 13,000 individuals. The ages of the individuals range from 16 to 77 with a median age of 33. Four images on average were taken of each individual with an average time of 164 days between each image.

Facial images annotated with age information are useful for developing automated age detection systems. Currently, no reliable methods (i.e., with low error rates) exist for age identification. Steinebach, *et al.* [21] have employed face recognition techniques to identify known illegal multimedia content, but they did not consider age classification.

2.5 Enron Corpus

The Enron Corpus introduced in 2004 is a well-known corpus in the area of forensic linguistics [9]. In its raw form, the corpus contains

619,446 email messages from 158 executives of Enron Corporation; the email messages were seized during the investigation of the 2001 Enron scandal. After data cleansing, the corpus contains 200,399 messages. The Enron Corpus is one of the most referenced mass collections of real-world email data that is publicly available.

The corpus provides a valuable basis for research on email classification, an important area in forensic linguistics. Klimt and Yang [10] suggest using thread membership detection for email classification and provide the results of baseline experiments conducted with the Enron Corpus. Data sets from the Enron Corpus are available at [3].

2.6 Global Intelligence Files

In February 2012, WikiLeaks started publishing the Global Intelligence Files, a large corpus of email messages gathered from the intelligence company Stratfor. WikiLeaks claims to possess more than 5,000,000 email messages dated between July 2004 and December 2011. As of September 2013, almost 3,000,000 of these messages have been available for download by the public [24]. WikiLeaks continues to release new email messages from the corpus on an almost daily basis.

Like the Enron Corpus, the Global Intelligence Files would provide a valuable basis for research in forensic linguistics. However, we are not aware of any significant research conducted using the Global Intelligence Files.

2.7 Computer Forensic Reference Data Sets

The Computer Forensic Reference Data Sets maintained by NIST [19] is a small data corpus created for training and testing purposes. The data sets include test cases for file carving, system memory analysis and string search using different encodings.

The corpus contains the following data:

- One hacking case scenario.
- Two images for unicode string searches.
- Four images for filesystem analysis.
- One image for mobile device analysis.
- One image for system memory analysis.
- Two images for verifying the results of forensic imaging tools.

This corpus provides a small but valuable reference set for tool developers. It is also suitable for training in forensic analysis methods.

3. Pitfalls of Data Corpora

Forensic corpora are very useful for education and research, but they have certain pitfalls.

- **Solution Specificity:** While a corpus is very valuable when developing methodologies and tools that solve research problems in digital forensics, it is difficult to find general solutions that are not somehow tailored to the corpus. Even when a solution is intended to work in general (with different corpora and in the real world), research and development efforts often slowly adapt the solution to the corpus over time, probably without even being noticed by the researchers. For example, the Enron Corpus is widely used by the forensics linguistics community as a single basis for research on email classification. It would be difficult to show that the research results based on this corpus apply to general email classification problems.

  This could also become an issue if, for instance, a general methodology or tool that solves a specific problem already exists, and another research group is working to enhance the solution. Using only one corpus during development increases the risk of crafting a solution that may be more effective and efficient than previous solutions, but only when used with that specific corpus.

- **Legal Issues:** The data in corpora such as Garfinkel's Real Data Corpus created from used hard disks bought from the secondary market may be subject to intellectual property and personal privacy laws. Even if the country that hosts the real-world corpus allows its use for research, legal restrictions could be imposed by a second country in which the research that uses the corpus is being conducted. The worst case is when local laws completely prohibit the use of the corpus.

- **Relevance:** Data corpora are often created as snapshots of a specific scenarios or environments. The data contained in corpora often loses its relevance as it ages. For example, network traffic from the 1990s is quite different from current network traffic – a fact that was pointed out for the DARPA Intrusion Detection Data Sets [4, 16]. Another example is a data corpus containing data extracted from mobile phones. Such a corpus must be updated very frequently with data from the latest devices if it is to be useful for mobile phone forensics.

Figure 1. Generating synthetic data based on a real-world scenario.

- **Transferability:** Many data corpora are created or taken from specific local environments. The email messages in the Enron Corpus are in English. While this corpus is valuable to forensic linguists in English-speaking countries, its value to researchers focused on other languages is debatable. Indeed, many important properties that are relevant to English and used for email classification may not be applicable to Arabic or Mandarin Chinese.

  Likewise, corpora developed for testing forensic tools that analyze specific applications (e.g., instant messaging software and chat clients) may not be useful in other countries because of differences in jargon and communication patterns. Also, a corpus that mostly includes Facebook posts and IRC logs may not be of much value in a country where these services are not popular.

4. Synthetic Data Corpus Generation

Aside from methodologies for creating synthetic data corpora by manually reproducing real-world actions, little research has been done related to tool-supported synthetic data corpus generation. Moch and Freiling [17] have developed Forensig2, a tool that generates synthetic disk images using virtual machines. While the process for generating disk images has to be programmed in advance, the tool allows randomness to be introduced in order to create similar, but not identical, disk images. In a more recent work, Moch and Freiling [18] present the results of an evaluation of Forensig2 applied to student education scenarios.

A methodology for generating a synthetic data corpus for forensic accounting is proposed in [14] and evaluated in [15]. The authors demonstrate how to generate synthetic data containing fraudulent activities from smaller collections of real-world data. The data is then used for training and testing a fraud detection system.

5. Corpus Generation Process

This section describes the process for generating a synthetic data corpus using the model-based framework presented in [27].

Figure 1 presents the synthetic data generation process. The first step in generating a synthetic data corpus is to define the data use cases. For

example, in a digital forensics class, where students will be tested on their knowledge about hard disk analysis, one or more suitable disk images would be required for each student. The students would have to search the disk images for traces of malware or recover multimedia data fragments using tools such as Foremost [1] and Sleuth Kit [2].

The disk images could be created in a reasonable amount of time manually or via scripting. However, if every student should receive different disk images for analysis, then significant effort may have to be expended to insert variations in the images. Also, if different tasks are assigned to different students (e.g., one student should recover JPEG files and another student should search for traces of a rootkit), more significant variations would have to be incorporated in the disk images.

The second step in the corpus generation process is to specify a real-world scenario in which the required kind of data is typically created. One example is a computer that is used by multiple individuals, who typically install and remove software, and download, copy, delete and overwrite files.

The third step is to create a model to match this scenario and serve as the basis of a simulation, which is the last step. A Markov chain consisting of states and state transitions can be created to model user behavior. The states correspond to the actions performed by the users and the transitions specify the actions that can be performed after the preceding actions.

5.1 Scenario Modeling using Markov Chains

Finite discrete-time Markov chains as described in [26] are used for synthetic data generation. One Markov chain is created for each type of subject whose actions are to be simulated. A subject corresponds to a user who performs actions on a hard disk such as software installations and file deletions. The states in the Markov chain correspond to the actions performed by the subject in the scenario.

In order to construct a suitable model, it is necessary to first define all the actions (states) that cause data to be created and deleted. The transitions between actions are then defined. Following this, the probability of each action is specified (state probability) along with the probability of each transition between two actions (transition probability); the probabilities are used during the Markov chain simulation to generate realistic data. The computation of feasible transition probabilities given state probabilities can involve some effort, but the process has been simplified in [28].

Next, the number of subjects who perform the actions are specified (e.g., number of individuals who share the computer). Finally, the details of each possible action are specified (e.g., what exactly happens during a download file action or a delete file action).

5.2 Model-Based Simulation

Having constructed a model of the desired real-world scenario, it is necessary to conduct a simulation based on the model. The number of actions to be performed by each user is specified and the simulation is then started. At the end of the simulation, the disk image contains synthetic data corresponding to the modeled real-world scenario.

5.3 Sample Scenario and Model

To demonstrate the synthetic data generation process, we consider a sample scenario. The purpose for generating the synthetic data is to test how different file carvers deal with fragmented data. The real-world scenario involves an individual who uses an USB memory stick to transfer large amounts of files, mainly photographs, between computers.

In the following, we define all the components in a model that would facilitate the creation of a synthetic disk image of a USB memory stick containing a large number of files, deleted files and file fragments. The resulting disk image would be used to test the ability of file carvers to reconstruct fragmented data.

- **States:** In the sample model, the following four actions are defined as Markov chain states:

    1. *Add Document File*: This action adds a document file (e.g., PDF or DOC) to the filesystem of the synthetic disk image. It is equivalent to copying a file from one hard disk to another using the Linux `cp` command.
    2. *Add Image File:* This action adds an image file (e.g., JPEG, PNG or GIF) to the filesystem. Again, it is equivalent to using the Linux `cp` command.
    3. *Write Fragmented Data:* This action takes a random image file, cuts it into multiple fragments and writes the fragments to the disk image, ignoring the filesystem. It is equivalent to using the Linux `dd` for each file fragment.
    4. *Delete File:* This action removes a random file from the filesystem. It is equivalent to using the Linux `rm` command.

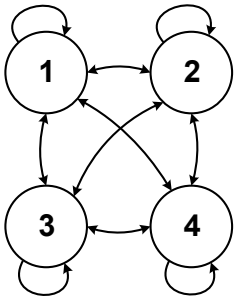


Figure 2. Markov chain used to generate a synthetic disk image.

- **Transitions:** Next, the transitions between the actions are defined. Since the transitions are not really important in the scenario, the Markov chain is simply constructed as a complete digraph (Figure 2). The state numbers in the Markov chain correspond to the state numbers specified above.

- **State Probabilities:** Next, the probability π_i of each action (state) i to be performed during a Markov chain simulation is specified. We chose the following probabilities for the actions to ensure that a large number of files and file fragments are added to the synthetic disk image and only a maximum of about half of the added files are deleted:

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_4) = (0.2\,, 0.2\,, 0.4\,, 0.2).$$

- **State Transition Probabilities:** Finally, the feasible probabilities for the transitions between the actions are computed. The framework is designed to compute the transition probabilities automatically. One possible result is the simple set of transition probabilities specified in the matrix:

$$P = \begin{bmatrix} 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \end{bmatrix}$$

  where p_{ij} denotes the probability of a transition from action i to action j.

6. Corpora Generation Framework

The framework developed for generating synthetic disk images is implemented in Java 1.7. It uses a modular design with a small set of core

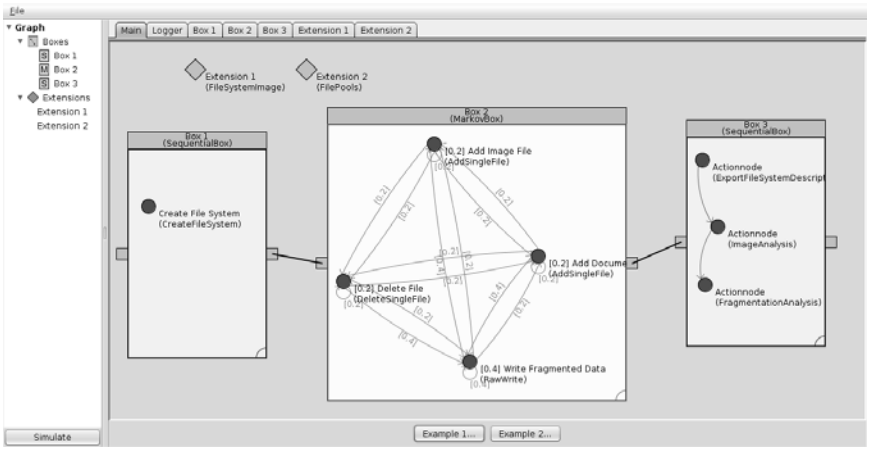


Figure 3. Screenshot of the model builder.

components, a graphical user interface (GUI) and modules that provide specific functionality. The GUI provides a model building interface that allows a model to be created quickly for a specific scenario using the actions available in the framework. Additionally, an image viewer is implemented to provide detailed views of the generated synthetic disk images.

New actions in the framework can be added by implementing a small number of interfaces that require minimal programming effort. Since the framework supports the specification and execution of an abstract synthetic data generation process, new actions can be implemented independently of a scenario for which a synthetic disk image is being created. For example, it is possible to work on a completely different scenario where financial data is to be created in an enterprise relationship management system. The corresponding actions that are relevant to creating the financial data can be implemented in a straightforward matter.

The screenshot in Figure 3 shows the model builder component of the framework. The Markov chain used for generating data corresponding to the sample scenario is shown in the center of the figure (green box).

7. Framework Evaluation

This section evaluates the performance of the framework. The sample model described above is executed to simulate a computer user who performs write and delete actions on a USB memory stick. The evaluation setup is as follows:

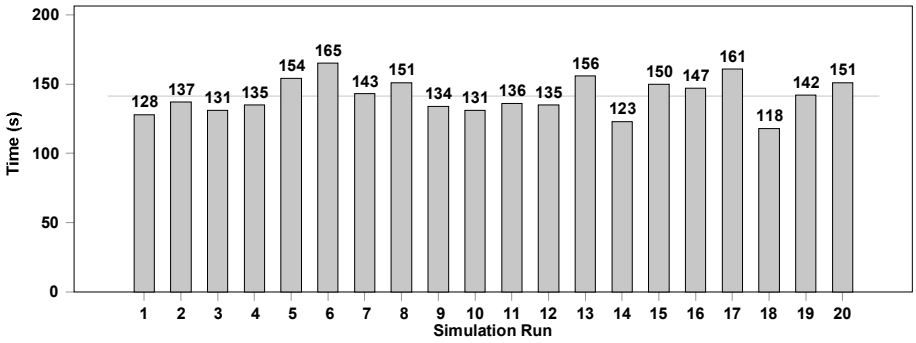
- **Model:** Described in Section 5.3.

- **Discrete Simulation Steps:** 4,000 actions.
- **Synthetic Disk Image Size:** 2,048 MiB (USB memory stick).
- **Filesystem:** FAT32 with 4,096-byte cluster size.
- **Add Document File Action:** A document (e.g., DOC, PDF or TXT) file is randomly copied from a local file source containing 139 document files.
- **Add Image File Action:** An image (e.g., PNG, JPEG or GIF) file is randomly copied from a local file source containing 752 image files.
- **Delete File Action:** A file is randomly chosen and deleted from the filesystem of the synthetic disk image without overwriting.
- **Write Fragmented Data Action:** An image file is randomly chosen from the local file source containing 752 image files. The file is written to the filesystem of the synthetic disk image using a random number of fragments between 2 and 20, a random fragment size corresponding to a multiple of the filesystem cluster size and randomly-selected cluster-aligned locations for fragment insertion.

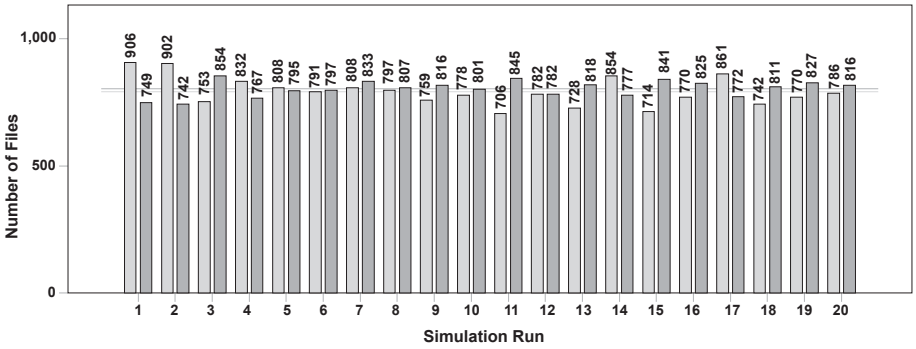
Twenty simulations of the model were executed using the setup. After each run, the time needed to completely generate the synthetic disk image was assessed, along with the amount of disk space used, number of files deleted, number of files still available in the filesystem and number of different file fragments written to the image.

Figure 4(a) shows the time required by framework to run each simulation. On the average, a simulation run was completed in 2 minutes and 21 seconds. Figure 4(b) presents an overview of the numbers of files that were allocated in and deleted from the synthetic disk images. Note that the allocated (created) files are shown in light gray while the deleted files are shown in dark gray; the average value is shown as a gray line. On the average, a disk image contained 792 allocated files and 803 deleted files, which are expected due to the probabilities chosen for the actions in the model.

Figure 5(a) shows the used disk space in the synthetic image corresponding to allocated files (light gray), deleted files (gray) and file fragments (dark gray). The used space differs considerably over the simulation runs because only the numbers of files to be written and deleted from the disk image were defined (individual file sizes were not specified). Since the files were chosen randomly during the simulation



(a) Time required for each simulation run.



(b) Numbers of allocated files and deleted files.

Figure 4. Evaluation results for 20 simulation runs.

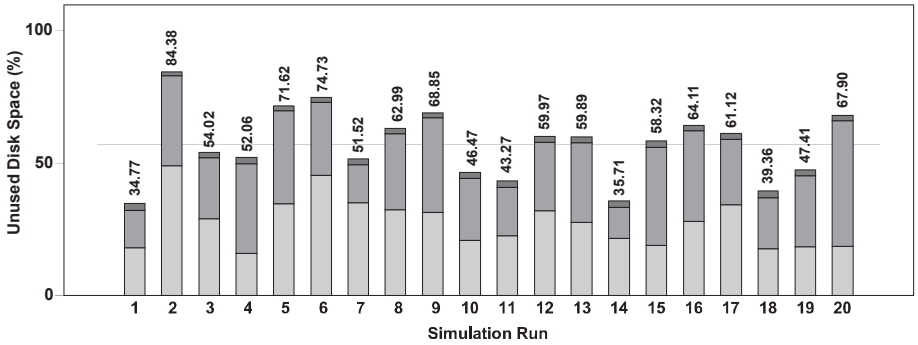
runs, the file sizes and, therefore, the disk space usage differ. On the average, 57% of the available disk space was used.

Figure 5(b) shows the average number of file fragments per file type over all 20 simulation runs. The writing of fragmented data used a dedicated file source containing only pictures; this explains the large numbers of JPEG and PNG fragments.

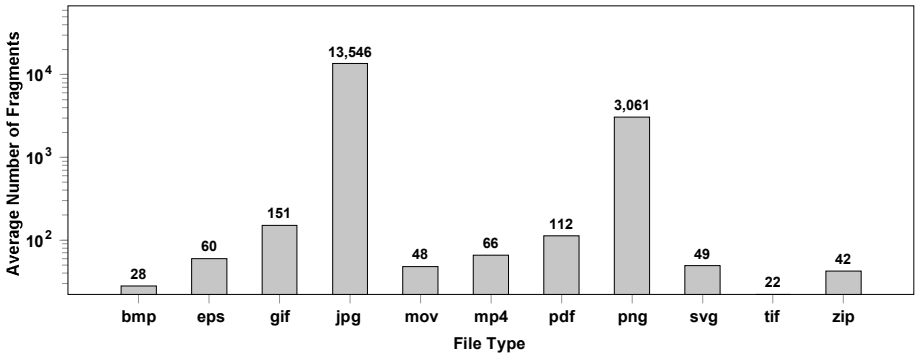
Figure 6 shows a screenshot of the image viewer provided by the framework. Information such as the data type, fragment size and filesystem status (allocated and deleted) is provided for each block.

8. Conclusions

The framework presented in this paper is well-suited to scenario-based model building and synthetic data generation. In particular, it provides a flexible and efficient approach for generating synthetic data corpora. The



(a) Used disk space corresponding to allocated files, deleted files and file fragments.



(b) Average number of fragments per file type.

Figure 5. Evaluation results for 20 simulation runs.

experimental evaluation of creating a synthetic disk image for testing the fragment recovery performance of file carvers demonstrates the utility for the framework.

Unlike real-world corpora, synthetic corpora provide ground truth data that is very important in digital forensics education and research. This enables students as well as developers and testers to acquire detailed understanding of the capabilities and performance of digital forensic tools. The ability of the framework to generate synthetic corpora based on realistic scenarios can satisfy the need for test data in applications for which suitable real-world data corpora are not available. Moreover, the framework is generic enough to produce synthetic corpora for a variety of domains, including forensic accounting and network forensics.

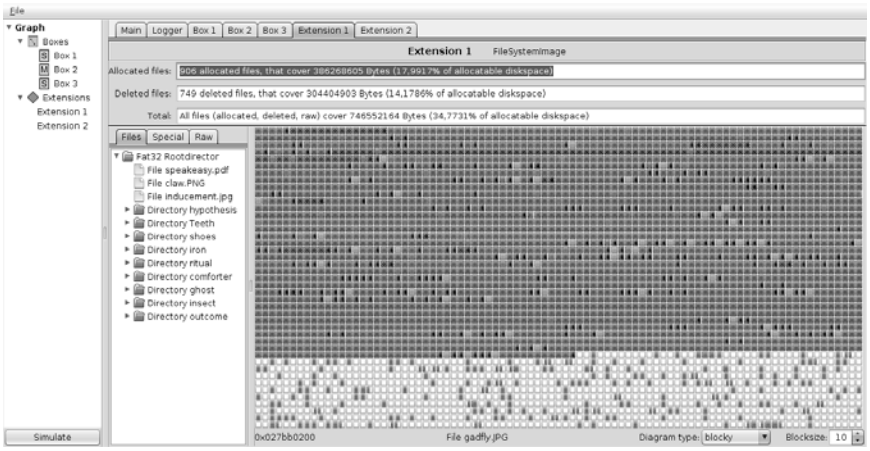


Figure 6. Screenshot of the image viewer.

Acknowledgement

This research was supported by the Center for Advanced Security Research Darmstadt (CASED).

References

[1] Air Force Office of Special Investigations, Foremost (`foremost.sourceforge.net`), 2001.

[2] B. Carrier, The Sleuth Kit (`www.sleuthkit.org/sleuthkit`), 2013.

[3] W. Cohen, Enron Email Dataset, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania (`www.cs.cmu.edu/~enron`), 2009.

[4] S. Garfinkel, Forensic corpora, a challenge for forensic research, unpublished manuscript, 2007.

[5] S. Garfinkel, Lessons learned writing digital forensics tools and managing a 30 TB digital evidence corpus, *Digital Investigation*, vol. 9(S), pp. S80–S89, 2012.

[6] S. Garfinkel, Digital Corpora (`digitalcorpora.org`), 2013.

[7] S. Garfinkel, P. Farrell, V. Roussev and G. Dinolt, Bringing science to digital forensics with standardized forensic corpora, *Digital Investigation*, vol. 6(S), pp. S2–S11, 2009.

[8] M. Grgic and K. Delac, Face Recognition Homepage, Zagreb, Croatia (`www.face-rec.org/databases`), 2013.

[9] B. Klimt and Y. Yang, Introducing the Enron Corpus, presented at the *First Conference on Email and Anti-Spam*, 2004.

[10] B. Klimt and Y. Yang, The Enron Corpus: A new dataset for email classification research, *Proceedings of the Fifteenth European Conference on Machine Learning*, pp. 217–226, 2004.

[11] Lincoln Laboratory, Massachusetts Institute of Technology, DARPA Intrusion Detection Data Sets, Lexington, Massachusetts (`www.ll.mit.edu/mission/communications/cyber/CSTcorpora/idev al/data`), 2013.

[12] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham and M. Zissman, Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation, *Proceedings of the DARPA Information Survivability Conference and Exposition*, vol. 2, pp. 12–26, 2000.

[13] R. Lippmann, J. Haines, D. Fried, J. Korba and K. Das, The 1999 DARPA off-line intrusion detection evaluation, *Computer Networks*, vol. 34(4), pp. 579–595, 2000.

[14] E. Lundin, H. Kvarnstrom and E. Jonsson, A synthetic fraud data generation methodology, *Proceedings of the Fourth International Conference on Information and Communications Security*, pp. 265–277, 2002.

[15] E. Lundin Barse, H. Kvarnstrom and E. Jonsson, Synthesizing test data for fraud detection systems, *Proceedings of the Nineteenth Annual Computer Security Applications Conference*, pp. 384–394, 2003.

[16] J. McHugh, Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory, *ACM Transactions on Information and System Security*, vol. 3(4), pp. 262–294, 2000.

[17] C. Moch and F. Freiling, The Forensic Image Generator Generator (Forensig2), *Proceedings of the Fifth International Conference on IT Security Incident Management and IT Forensics*, pp. 78–93, 2009.

[18] C. Moch and F. Freiling, Evaluating the Forensic Image Generator Generator, *Proceedings of the Third International Conference on Digital Forensics and Cyber Crime*, pp. 238–252, 2011.

[19] National Institute of Standards and Technology, The CFReDS Project, Gaithersburg, Maryland (`www.cfreds.nist.gov`), 2013.

[20] K. Ricanek and T. Tesafaye, Morph: A longitudinal image database of normal adult age-progression, *Proceedings of the Seventh International Conference on Automatic Face and Gesture Recognition*, pp. 341–345, 2006.

[21] M. Steinebach, H. Liu and Y. Yannikos, FaceHash: Face detection and robust hashing, presented at the *Fifth International Conference on Digital Forensics and Cyber Crime*, 2013.

[22] T. Vidas, MemCorp: An open data corpus for memory analysis, *Proceedings of the Forty-Fourth Hawaii International Conference on System Sciences*, 2011.

[23] Volatilty, The Volatility Framework (`code.google.com/p/volatility`), 2014.

[24] WikiLeaks, The Global Intelligence Files (`wikileaks.org/the-gifiles.html`), 2013.

[25] K. Woods, C. Lee, S. Garfinkel, D. Dittrich, A. Russell and K. Kearton, Creating realistic corpora for security and forensic education, *Proceedings of the ADFSL Conference on Digital Forensics, Security and Law*, 2011.

[26] Y. Yannikos, F. Franke, C. Winter and M. Schneider, 3LSPG: Forensic tool evaluation by three layer stochastic process-based generation of data, *Proceedings of the Fourth International Conference on Computational Forensics*, pp. 200–211, 2010.

[27] Y. Yannikos and C. Winter, Model-based generation of synthetic disk images for digital forensic tool testing, *Proceedings of the Eighth International Conference on Availability, Reliability and Security*, pp. 498–505, 2013.

[28] Y. Yannikos, C. Winter and M. Schneider, Synthetic data creation for forensic tool testing: Improving performance of the 3LSPG Framework, *Proceedings of the Seventh International Conference on Availability, Reliability and Security*, pp. 613–619, 2012.