Using open source forensic carving tools on split dd and EWF files.

PALMIERI, Gareth and ZARGARI, Shahrzad <http://orcid.org/0000-0001-6511-7646>

Using Open Source Forensic Carving tools on split dd and EWF files.

Gareth Palmieri
Department of Computing
Sheffield Hallam University
Sheffield
Gareth.G.Palmieri@student.shu.ac.uk

Shahrzad Zargari
Department of Computing
Sheffield Hallam University
Sheffield
S.Zargari@shu.ac.uk

***Abstract*—This study tests a number of open source forensic carving tools to determine their viability when run across split raw forensic images (dd) and Expert Witness Compression Format (EWF) images. This is done by carving files from a raw dd file to determine the baseline before running each tool over the different image types and analysing the results. A framework is then written in python to allow Scalpel to be run across any split dd image, whilst simultaneously concatenating the carved files and sorting by file type. This study tests the framework on a number of scenarios and concludes that this is an effective method of carving files using Scalpel over split dd images.**

***Keywords*—open-source, forensics, file carving, Scalpel, python framework**

I. Introduction

As storage capacity increases, raw forensic images such as dd are decreasing in popularity with split raw images or compressed image formats such as Expert Witness Compression Format (EWF) becoming the standard. Due to this, most forensic practitioners will be working with EWF or split dd files and this paper helps determine the usefulness of open-source forensic carving tools

Open Source forensic carving tools work by running across either a forensic image of the live media and, usually, uses the headers and footers found to extract (carve) the files. The benefit of this is that carving tools work on any Operating System(OS) even if the metadata for this has been destroyed.

The majority of open-source file carvers are designed to run across raw (dd) forensic images, therefore the purpose of the first section of this paper is to test their viability across split dd images and the compressed image type, EWF. The second section will discuss a novel and original approach to running Scalpel across a split dd image, by implementing a framework written in python. This approach will then be tested to ensure the capability and practicality of the framework.

A background of forensic images and carving tools are given in the following section (section II), before the methodology used for testing is explained in section III. Section III also contains the results of this testing and the baselines for the three carving tools used. Section IV discusses the framework created, and its technical details, before examining the results of testing this framework across a forensic image. Section V draws conclusions from the initial testing of the carving tools as well as from the framework and its subsequent testing. These conclusions, and the potential for further work, are explored in section VI with a number of suggestions for future improvements to the framework given.

II. Background

Raw dd images are a complete bit-by-bit forensic image of the media taken, meaning that the images are equal in terms of size as the size of the media. As 1TB+ HDD's and other media become more common with the drop in prices per GB [1], [2], the size of these raw dd images becomes a problem as working with 1TB+ dd images is not practical. To counter this, dd images can be split into segments allowing for smaller files to be created which holds the forensic image but is easier to store.

The EWF format is used by the two largest Digital Forensic Suits, EnCase and FTK [3] amongst others, and has come the standard across forensic companies, both commercial and public. EWF files are most commonly split into a number of sequentially numbered segment files, starting with the extension, E01 and incrementing with each segmented split.

The number of files in the image depends on the size of the segmented splits as defined by the user and the size of the media to be acquired. Like EWF, split dd images split the original media depending on the segment size chosen by the user, and these are also named sequentially, starting from .001 or .000 dependent on the forensic imaging software used. EWF images and split dd images use a number of splits to create a number of smaller files whilst preserving the integrity of the forensic image. Each spilt/segment contains a section of the disk and are numbered and read sequentially. The terms split and segment to describe a EWF or dd split are used interchangeably throughout this paper.

Although open source tools and their effectiveness within computer forensics have been comprehensively tested [4], [5] & [6], the aim of this paper is to determine the effectiveness of open-source carving tools on split dd and EWF images and not the comparison, or effectiveness, of the tools themselves. For this reason, both Scalpel and Foremost have been chosen as they are/were the leading authority on open source carving

tools, regardless of the fact that Scalpel is a re-write of Foremost [7], which is no longer supported.

Scalpel is a high performance, lightweight file carver that uses a database of headers and footers to search and carve files from both live and imaged media [8]. Scalpel is based loosely on the open source Foremost file carver and shares some of the same code, whilst implementing a much faster and more accurate method of carving, this results in far less false positives than foremost.

A comparison of Foremost and Scalpel has been conducted multiple times [9], [7], [10] with file and artefacts recovered compared in relation to a mobile phone image [11]. For the most part they return similar results but they can both carve files the other does not, and this is why they have both been chosen for this investigation.

Scalpel was chosen as it carves a variety of files and a large amount of research into Scalpels efficiency has been done, [12] [13] and it has been proven to be efficient and capable. Foremost was chosen as it offers a comparison to the newer Scalpel whilst still being used widely and still being very efficient. RecoverJpeg has been included in both the Backtrack and Kali Linux distributions and is also available to download as a standalone tool so it was chosen as an alternative to the other two tools whilst representing file carvers that are specifically for one/two files types.

As there is a number of open-source file carvers, which all differ slightly in the method used to carve, and the file types they are able to carve, a single comparison of all these tools and the files they recover has not been carried out as the results would be convoluted and unrepresentative.

III. Methodology

A number of open source forensic tools will be run to determine their capability and usability across split dd and EWF images. The tools tested will be; Foremost, Scalpel and RecoverJpeg. Each of these tools have their own positives and negatives and the types of files they are able to carve also differ making them ideal to allow a large range of testing options. A number of other tools were considered and there are many which would have been suitable for this research. Due to the vast number of open-source carving tools, the number had to be limited and therefore the choice of three tools was made to thoroughly test the hypothesis with a number of tools and consequently the types of files that could be carved. This research could be carried out using different open-source tools, but as explained above and in Section 2 (Background) Scalpel, Foremost and RecoverJpeg were chosen to offer diverse tools with varied file formats being carved for, although other tools were considered.

To test the effectiveness of these tools across a split dd and EWF file, first a baseline carve will be done using the software over a raw dd image, this will show how many files the tool should carve from the complete image. As the dd and EWF images will be split, the hypothesis is that the tools will only run over the initial dd (001/000) and initial EWF (E01) split, ignoring the subsequent files, and will therefore bring back a significantly lower number of carved files.

The media used will be that of a 4GB USB memory stick formatted using FAT32, and it will be imaged using FTK Imager Lite v 3.1.1.8 and guymager v 0.7.3-3. A USB device with FAT32 was chosen as the carving tools chosen are able to carve from this type of media, and the device itself was able to carry a large range, and amount, of file types further increasing the thoroughness of the testing.

Two imaging tools will be used to ensure the results are not influenced by the imager. FTK will be used to image the EWF files and guymager will be used for both the raw dd and split dd images, and a software write-blocker will be used to ensure the integrity of the image.

The results of the tools are shown below with an accompanying table to further illustrate the findings.

The framework will be coded using python and it will then be tested on a number of split dd images to determine its success rate. Python was chosen as the coding language as it has a number of libraries and functions that were needed, as well as being adaptable and flexible and also being cross platform allowing the framework to be adapted for use on other operating systems.

A. *Foremost*

Foremost is run from the command line using a number of parameters and user inputs. Running Foremost over the original raw dd image returned 638 files. Running Foremost over the split dd image returned only 282 carved files, this demonstrates that this tool does not run over all splits, only the first split.

Running Foremost over the EWF file returns 0 files, this is due to the format being both compressed and split. This makes Foremost unsuitable for running over and EWF files.

Table 1 Foremost Results

	Raw dd image	Split dd image	EWF split image
Foremost	638	252	0

B. *Scalpel*

Scalpel is run from the command line using a number of parameters and a configuration file. This configuration file allows the user to specify the types of files they wish to carve for. For this test the majority of image, video and data files will be carved. Running Scalpel over the original dd image returned 1,259 files, and as can be seen, this is a vast improvement on the number returned by Foremost.

Running Scalpel over the split dd image returned 658 files. This shows, that like Foremost, only the first (001) image was carved and Scalpel sees each split as its own forensic image rather than a split file.

Running Scalpel over the split EWF image resulted in 23 files being carved, with all files being false positives. It can therefore be seen that Scalpel does not run over EWF images.

Table 2 Scalpel Results

	Raw dd image	Split dd image	EWF split image
Scalpel	1,259	658	23

C. *RecoverJpeg*

RecoverJpeg, as the name suggests, carves .jpg files and .mov files and is therefore more limited than the other tools used, however it is very efficient and lightweight and provides an alternative to the multi-file carvers. RecoverJpeg carved 151 jpeg images from the original raw dd file, but only 145 from the split dd file. Again this demonstrates that the tool runs only over the first split file in the image.

RecoverJpeg did not carve any images from the EWF file.

Table 3 RecoverJpeg Results

	Raw dd image	Split dd image	EWF split image
RecoverJpeg	151	145	0

In conclusion, none of the open-source carving tools tested were able to run directly over EWF files successfully, while they only ran over the first segment in split dd images. This makes none of the above tools suitable for carving across the majority of forensic images which are split.

IV. SCALPEL FRAMEWORK

Running Scalpel over a split dd image would require the examiner to run Scalpel over each split file individually, and then examine the results of each split file individually which would be held in multiple locations. This method would take a large amount of time to complete and would increase the chance of errors due to repetition and the chance of missing a segment.

This framework aims to increase the speed at which a split dd file can be carved by running Scalpel sequentially over the dd segments of a forensic image and combine the findings into a single folder which in turn is sorted by file type. The framework will be written in python and run on Linux based machines. It will take two user inputs, the location of the split dd file (the absolute directory path) and the name of the dd file without extension. These inputs will allow the framework to calculate how many dd splits there are and the working directory Scalpel should run in. This section will not explain every line of code individually but will discuss the framework on a higher level, due to the length of code it can be found, complete with comments, at https://github.com/g-palmieri/Scalpel_framework. Scalpel version 1.60, which is based on Foremost 0.69 will be used, but this framework should be compatible with both older and any future versions of Scalpel providing the input parameters are the same.

Once the user inputs have been accepted, the framework calculates the number of dd splits in that directory that relate to the file name given, and then loops over the main script that number of times. The script starts with the .000 or .001 (this is dependent on the tool used to create the image) file, the first dd split, and increments each loop until .nnn is equal to the number of split files. The script uses Scalpels inbuilt ‘output folder’ feature to specify a temporary directory where the files carved from each dd split are carved too, before being renamed and moved to the ‘Complete folder’, whilst the temporary directory is deleted.

As files carvers do not have access to the FAT/MFT or inode data, they cannot replicate the original files name or any of its associated metadata. Scalpel outputs the carved files using a sequential numbering system starting at ‘00000000’. The framework will run multiple instances of Scalpel, and consequently these file names will be repeated and therefore the possibility of overwriting images becomes an issue. To counter this, once Scalpel has finished carving files from each dd split, it will rename all files carved using the name of the split as the pre-fix, e.g. the twelfth file carved from the second dd split would have the name, dd002-(00000012), and the nineteenth file from the fourth split would have the name dd004-(00000019). This naming method not only avoids any files being overwritten, it also allows the examiner to link the carved files with the specific dd split, allowing for further investigation into that segment if it is required.

The last section of code runs through the carved files and sorts into specific folders based on the file extension, for example, all jpg, tif, gif files, etc. are sorted into the “images” folder, whilst all, ZIP files are sorted into the “ZIP” folder. This file sorting can be adapted to include more file extensions, or to place certain files in certain folders, by adding to, or commenting out lines of code. This makes reviewing the carved data much easier and faster, whilst also maintaining the look of a standard Scalpel output.

A. *Testing*

Using the same forensic image as previously, the Scalpel framework was run over the split dd file and returned a total of 1,257 carved files. These files were carved from the entire dd image, irrespective of it being a split dd image, using one command. It can be seen that the raw dd file (non-split) returned 1,259 carved files. This shows a drop of only 2 files and this can be accounted for if a file spans two splits, i.e., the header on one split and the footer on another.

A different USB device, this time a 16GB, was also forensically imaged using both raw dd and split dd image types and Scalpel was run over these images. Scalpel carved 403 files from the raw dd image whilst 403 files were also carved form the split dd image using the framework. This shows the consistency of the framework and its ability to run over split dd images successfully.

A third USB device of 4GB was forensically imaged as above. Scalpel was run over the raw dd image and carved 1,938 files, the framework was then used to run Scalpel over the split dd image and this also carved 1,938 files.

It can be seen that this framework creates a quick and effective way of running Scalpel over a split dd image with the results being comparable to having a raw dd image, whilst

maintaining the ease of use of the Scalpel tool and its convenient, sorted output.

V. Conclusion

None of the open source carving tools tested were capable of running over split dd or EWF files automatically and viewed each split as an individual forensic image as opposed to a joined EWF or dd file. The consequence of this was that only the initial split was carved and therefore the majority of the image was not carved meaning that a large amount of evidence would have been missed. This limits the usability of these tools as the majority of forensic imagers use the EWF format or the split dd format as standard, and carving files using these forensic tools would be time consuming. It can be seen that the original hypothesis is therefore correct, with none of the tools running over more than the first image segment and thus the number of carved files being significantly lower.

The framework circumnavigates this problem and allows Scalpel to be run over any split dd file, regardless of its original size, segment size or number of splits, using only one command and two user inputs. It concatenates the files by type and outputs the files in a sorted, readable folder structure. In the tests completed, the framework was equal in results to carving the raw dd image showing the effectiveness of running Scalpel in this manner. This creates a productive, fast and useful tool for forensic investigators.

None of the open-source tools were able to carve any files from the EWF image, this is due to the construction of the EWF format, along with the compression with the carving tools also only recognized the first segment, as explained above. Scalpel carved 23 false positive files from the EWF file, causing it to look as if it has successfully carved files. The python framework could easily be adapted to run across split EWF files, but as Scalpel is unable to carve files from this format it would simply return more false positives and it was therefore disregarded. This makes the use of these tools redundant in most situations as, with the increasing size of digital media, split and/or compressed forensic images are becoming standard.

VI. Future Work

Using more open-source forensic tools would allow a deeper understanding into this subject, with the ability to compare and quantify the results across a number of dd, split dd and EWF images. Using this data and analysing the number of false positives would give an indication to the effectiveness of these tools, both in general, and in relation to the type of file the tool was run over, however this depth of research is outside the scope of this current paper.

Future work would be to make changes to the framework that would allow Scalpel to be used in this manner on both Linux and Windows machines. This would require a re-write of portions of the script as the naming conventions are different on the two OS's, and would then require thorough re-testing on all platforms and versions of python.

Due to the nature of the framework, it could be adapted to run using different forensic carving tools, separately or as a package offering a complete carving solution for split dd files using open source tools. This new framework would require a hashing and sorting mechanism to reduce the number of duplicates found by the various tools. The output would be similar to that of the current framework, with the files being sorted by file type and renamed to reflect the split they originated from.

Opening the framework up to the forensic community would allow for the framework to be adapted and the individual needs of the users met. As it is already hosted on the code sharing site, GitHub, making it available would be practical and allow any persons to fork and adapt the code, whilst allowing any issues to be highlighted. An interesting continuation of this paper would be to review this framework and any additions/modifications made after a period of time of it being in the community and come to a conclusion about its practicality and effectiveness at the initial time it was published, and the merit of any adaptations made.

References

[1] The Wayback Machine, "Average Cost of Hard Drive Storage," statisticbrain, 11 November 2014. [Online]. Available: http://www.statisticbrain.com/average-cost-of-hard-drive-storage/. [Accessed 1 December 2015].

[2] Statistic Brain, "Average Cost of Hard Drive Storage," 2016. [Online]. Available: http://www.statisticbrain.com/average-cost-of-hard-drive-storage/. [Accessed 21 March 2017]

[3] J. Metz, "EWF specification," [Online]. Available: https://53efc0a7187d0baa489ee347026b8278fe4020f6.googledrive.com/host/0B3fBvzttpiiSMTdoaVExWWNsRjg/Expert%20Witness%20Compression%20Format%202%20(EWF2).pdf. [Accessed 04 December 2015].

[4] C. Altheide and H. Carvey, Digital Forensics with Open Source Tools, Burlington : Elsevier Science, 2011.

[5] E. E. Northrop and H. R. Lipford, "Exploring the usability of open source network forensic tools," in ACM Conference on Computer and Communications Security, New York, 2014.

[6] S. Shiaeles, A. Chryssanthou and V. Katos, "On-scene triage open source forensic tool chests: Are they effective?," Digital Investigation, vol. 10, no. 2, pp. 99-115, 2013.

[7] C. Beek, "Intro to file carving," McAfee, 2011. [Online]. Available: http://www.mcafee.com/uk/resources/white-papers/foundstone/wp-intro-to-file-carving.pdf. [Accessed 25 November 2015].

[8] G. G. R. III, "Scalpel: A Frugal, High Performance File Carver," New Orleans, 2005.

[9] A. Hoog, "Difference between foremost and scalpel?," NowSecure, 13 January 2009. [Online]. Available: https://www.nowsecure.com/blog/2009/01/13/difference-between-foremost-and-scalpel/. [Accessed 28 November 2015].

[10] A. Schuster, "Scalpel 1.54 vs. Foremost 1.1," 18 February 2016. [Online] Available: http://computer.forensikblog.de/en/2006/02/scalpel-154-vs-foremost-11.html [Accessed 21 March 2016].

[11] G. Grispos, T. Storer and W. B. Glisson, "A comparison of forensic evidence recovery techniques for a windows mobile smart phone," Digital Investigation, vol. 8, no. 1, pp. 23-36, 2011.

[12] X. Zha and S. Sahni, "Fast In-Place File Carving For Digital Forensics," [Online]. Available: http://www.cise.ufl.edu/~sahni/papers/scalpel.pdf. [Accessed 15 December 2015].

[13] B. Kloet, "Advanced File Carving - How much evidence are you ignoring?," September 2010. [Online]. Available: https://digital-forensics.sans.org/summit-archives/2010/eu-digital-forensics-incident-response-summit-bas-kloet-advanced-file-carving.pdf. [Accessed 15 December 2015].