International Conference on Advances in Human Machine Interaction (HMI - 2016),
March 03-05, 2016, R. L. Jalappa Institute of Technology, Doddaballapur, Bangalore, India

A Method for Carving Fragmented Document and Image Files

Akshara Ravi Raj Kumar T
Dept. of CSEDept of CSE
College of Engineering,Kallooppara,
Pathanamthitta,Kerala,IndiaPathanamthitta,Kerala,IndiaPathanamthitta,Kerala,India
akshararavi6@gmail.comrajcek@gmail.com

Angelo Renju Mathew
Dept of CSE
College of Engineering,KallooppараCollege of Engineering,Kallooppara
angelomanimala@gmail.com

***Abstract*—**Recovering deleted files play an important role in a digital forensic investigation. When a file is deleted, only pointers that link file's metadata to its content are deleted and metadata entry is marked as deleted. As long as data is not overwritten or wiped, deleted data will remain in unallocated space. One of the methods that can be used to recover these deleted files is file carving. File carving reconstructs files only based on their content unlike traditional data recovery methods that use metadata that points to the content. It is mainly done using headers and footers of file types. One of the primary challenges in file carving is to recover deleted files that were fragmented. When a file is fragmented, carving only based on header and footer will produce corrupted file. So this paper discusses a method for carving fragmented document and image files from a FAT32 formatted USB drive.

Keywords--fragmentation, carving, signature based approach, candidate weights, dictionary based approach, brute force approach.

I INTRODUCTION

Data carving is a technique that works by extracting files out of raw disk image, based on file format characteristics and internal structures [1]. It reconstructs files based on their content and does not use metadata that points to the clusters allocated to files. Data carving techniques are used during a digital investigation where the unallocated space of a disk is analyzed to extract deleted files. The files are carved from the unallocated space using file type-specific header and footer values.

Carving is widely used for forensics and data recovery. In the case of data recovery, data carving can recover files from a device that has been damaged and as a result important metadata entries are corrupted. For computer forensics, criminals may note down the clusters allocated to a confidential file and intentionally delete confidential files and reallocate its metadata entries to new files so that others cannot trace those files. In that case, block by block analysis of unallocated space is performed using the file type-specific header and footer values. In short, data carving can recover files that have been deleted and have had their directory entries reallocated to other files, but for which the sectors allocated to files on the disk have not yet been overwritten.

There are a number of tools that can perform file carving, based on different techniques. Header-footer carving is one of the simplest types of carving. It searches through the unallocated space of disk for the headers and footers of different file types and extract the data in between.Header-footer carving has a number of major problems [2]:

1) If the header and/or footers are short, then it can appear as raw data anywhere on disk even at points where there is no file start or end. PNG files have relatively long headers and footers, but JPEG headers and footers are short. Therefore this method produces many false positives in case of JPEG files.

2) This technique is not effective for fragmented and partial files.Suppose the header of a JPEG file is found in cluster 500 and footer in cluster 502.In cluster 501, there is the data of a PDF file.A signature based carver locates header and footer and consider clusters 500-502 as a JPEG file.But when we try to open image in an image viewer,an error is displayed that tells "This application does not support this file type". The problem is with incorrect cluster 501.

3) Some files have no fixed headers and footers. Header-Footer carving fails for these files. This is the case for plain-text file formats, like text documents and html files.

Basically, data carving only look for headers and footers of files and will not examine sectors inserted, deleted or modified in between. In that case if there is unrelated data in between, it is also carved considering it as a part of the file. Thus the file is incorrectly carved and manual opening using software fails. So this paper deals with the carving of fragmented document and image files in FAT32 file system. But it uses some of the techniques used by carvers that carve contiguous files like header-footer carving.

This paper is organized as follows: Section II describes fragmentation problem. Section III discusses the general steps to reconstruct fragmented files. Section IV describes the carving of fragmented documents.Section V describes the carving of fragmented images. Section VI gives the system design. Section VII gives a comparison to existing methods.Section VIII describes the conclusion and future work.

978-1-4673-8810-8/16/$31.00 ©2016 IEEE

II FRAGMENTATION

Fragmentation is said to occur when files are not stored in the contiguous clusters on disk and thus group of clusters belonging to the file are separated by unknown clusters in between [3].Modern operating systems allocate contiguous clusters to a file so that it can be written without fragmentation. In that case it is faster to write and read those files. But when files are inserted, modified, appended and deleted in between already existing files, files may become fragmented. There are three conditions under which a file may be fragmented:

1. There is no contiguous free disk space where file can be written without fragmentation.

2. When there are no free clusters at the end of a file and we have to append data to it, file may become fragmented. But some file systems relocate the original file to a large enough free space to hold appended data also so that file is again contiguous.

3. The file system may not write files of a certain size in a contiguous manner.

Allocation of files is under the control of operating system not the user. The operating system always tries to allocate contiguous sectors to files. So fragmentation of files was not a problem. But now, several tools like PassMarkFragger are available that allow the user to manually fragment files. Using this tool, user can choose the number of fragments, their size, location and also visualize them. These tools can be used by anyone to fragment existing files. Then they can delete these files containing evidence and reallocate their metadata entries. Thus both traditional data recovery method based on metadata and carving based on content fails to recover the file.

For carving contiguous files, normal signature based carving techniques that use headers and footers works well. If a file is fragmented and we extract the clusters from the header to the footer to consider it as a single file, it will give incorrect reconstruction of the file. For example, Fig 1 shows an image file F1 fragmented into two fragments. One fragment comprising the clusters 1-4 and the other comprising clusters 8-9.But clusters 5-7 belongs to a document file.

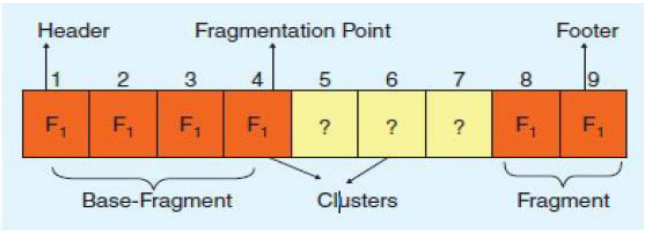

Fig 1 File F1 stored as two fragments with three unrelated clusters in between

Normal header-footer carvers will find the header in cluster 1 and footer in cluster 9 and extract the clusters 1-9 considering it as one. But the result will be a corrupted file. Fig 2 shows the corrupted image file when tried to open using Windows Photo Viewer.To recover fragmented files correctly a file carver have to determine the beginning of a file, the blocks that are required to reconstruct the file and then reassemble them in correct order.

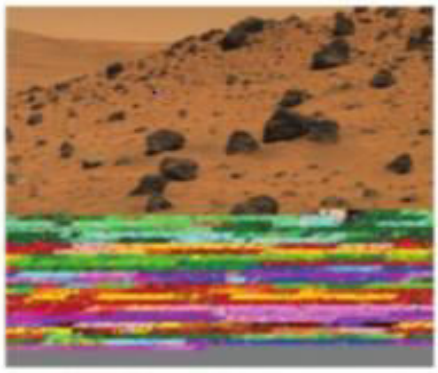

Fig 2 Error during Signature based carving

III GENERAL STEPS

The general steps for reassembling fragments before carving are as shown in the figure below [4].

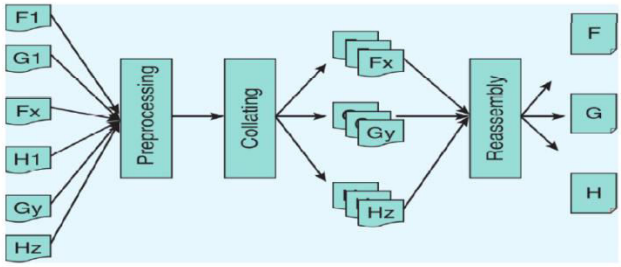

Fig 3 Steps for reassembling fragments

1)Preprocessing: Sometimes files found on a disk may be encrypted or compressed. So carving may not reconstruct these files effectively. So before carving, these files should be decrypted or decompressed. There are several algorithms for decompression. Techniques like brute force can be used to find out the key used for encryption. Also in case file system metadata is available, it can be used to remove allocated clusters of the disk from analysis. The goal of carving is to recover deleted files. So focus of carving is only on unallocated apace on disk. Thus preprocessing reduces data that is required for analysis.

2)Collating: Fragments on the disk may belong to different file types like image, document, audio, video etc. So before reassembly, these fragments could be grouped into different groups i.e., document fragments together, image fragments together etc.

978-1-4673-8810-8/16/$31.00 ©2016 IEEE

3)Reassembly:Determine the correct order in which fragments in a group should be merged to reassemble the original file. Sometimes, several orderings may be possible. Then the correct order is determined manually.

IV CARVING FRAGMENTED DOCUMENTS

Suppose there is a document A fragmented into pieces $A_1,A_2 \ldots A_n$. The aim is to determine the order in which these fragments should be joined to reconstruct the original file A. For this adjacent pair of fragments should be found out [5].

A technique that can be used for determining adjacent fragments is dictionary based approach. Fragment A_i follows A_j if the word that is split across the boundary is a valid dictionary word. For example, suppose fragment A_i end with "hospi" and A_j begin with "zen" and A_k begin with "tal".Here "hospital" is a valid dictionary word, whereas "hospizen" is not. Thus it may be concluded that fragment A_k follow A_i. For this, a dictionary of valid words in a particular language should be created and used. A document file written in English located next to clusters written in Spanish may indicate that the two fragments belong to different files. The disadvantage of this approach is that it is language specific i.e., a dictionary used for English documents cannot be used for French documents. Moreover, it is difficult to create dictionaries for non-text files like executable. But two ambiguities may arise with this approach.

- Suppose first fragment end with 'fir' and 2 other fragments begin with 'st'. 'first' is a valid dictionary word. Then it is difficult to determine which the correct one is.
- Suppose first fragment end with 'tr',one fragment begin with 'ee' and other with 'ue', both 'tree' and 'true are valid dictionary words.. Here also, it is difficult to determine which the correct one is.

Consider the reconstruction of .txt file as an example.The txt files have no specific format and structure unlike PDF andWORD files.In fragmented txt filerecovery,fragment Aj follows Ai if word that straddle boundary is a dictionaryword.So a dictionary of valid words is created. Cluster by cluster analysis need to be performed which requires a buffer of size equal to cluster size. Then cluster by cluster analysis is done by reading clusters one by one into buffer. Then extract the first and last words of each cluster and store in the database. Then form all possible combinations of first and last words and search in the dictionary.If the word is a valid dictionary method, then there is the possibility that two fragments are adjacent. Then merge the two corresponding clusters and write it to a file. By this approach, original txt file can be recovered.

V CARVING FRAGMENTED IMAGES

Reconstruction of fragmented images is somewhat easier and faster than that of documents.This is because it is easy to identify starting and ending fragments. Only the difficulty is in finding out the in between fragments. But there are several methods for assigning candidate weights that give the probability with which one fragment follow other. In short, header-footer carving is the backbone of the new carver that reconstructs fragmented image files. It is supported by brute force approach that considers all possible permutations of fragments and techniques that filter results of brute force approach by finding the best match based on candidate weights.

Most image types will have specific headers and footers. These are hexadecimal values that denote the starting and ending points of files. Some file types will have both headers and footers like JPG, GIF etc., others may have only header like BMP. In that case, there will be a value in the header portion that gives the size of the image in bytes. So this size value can be used in carving. Also some file types will have several internal sections like PNG as shown in the figure below.

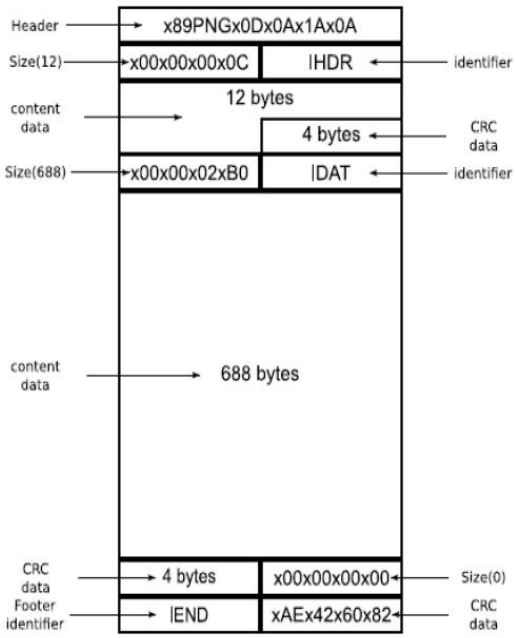

Fig 4 PNG File Structure

PNG have a long header and footer.In addition to it,it has three identifiers denoting several sections of image file.IHDR denote header portion.IDAT denote image data.IEND denote footer portion. Header of file types also gives resolution of image i.e., width and height of image in pixels. This information is used to calculate candidate weights for finding the best match.

The new carver can reconstruct two or three fragmented image files like JPG, PNG, and GIF. It does not work for BMP because of the absence of footer. So it is difficult to determine the point to which BMP file should be carved.A configuration file that list headers and footers of image file types is required to group different image types and reassemble them. Table 1 shows the headers and footers of some image types which may be helpful in carving fragmented image files.

Table 1 Headers and Footers of Image Types

Image Type	Header	Footer
JPEG	FFD8	FFD9
PNG	89504E470D0A1A0A	AE426082
GIF	474946383961or 474946383761	3b

The problem with carving is false positives. Headers and footers are hexadecimal values. So these values may appear anywhere on the disk as raw data. But it may be treated as image header or footer. So many false positives may result. In the case of GIF, the footer is very small (3b).So the number of false positives will be larger than for PNG which have large headers and footers.

Carver for reconstructing fragmented image files works mainly based on header-footer carving. In header-footer carving,disk image is searched on a cluster by cluster basis by using a buffer of the same size as clusters. First, search for headers and footers of different file types. From this information, the number of image types to be reconstructed can be determined. Clusters where header and footer is found can be stored in a database. In the case of two fragmented files, one fragment will have the header and other will have the footer. Then form all possible combinations of two fragments. Brute force approach is used because it will consider all possible permutationsof clusters.So correct output will be there.But it will result in many false positives. Finally,cluster having header and cluster having footer are merged andwritten to a file and saved with appropriate extension.A sequence of files is created. But this approach takes large processing time.

In the case of three fragmented files,one fragment will have the header, other will have thefooter and other will have no specific identifier which is difficult to identify.Sosearch for the header and footer in all clusters and store the corresponding cluster address in the database. So, all possible permutations of header fragment, any other fragment and footer fragment should be made. It will result in many false positives. To filter these results and determine the best match, assign candidate weights so that adjacent fragments can be found out.

For determining adjacent fragments, examine pixels that form the boundary when the fragments are joined together. Absolute sum of predictionerrors is calculated for the pixels along the boundary between the two fragments. Compute prediction errors for last row of pixels in the first fragmentand the first row of pixels in the second fragment as shown in the figure 5. In other words, determine the pixel values along the boundary formed between the two fragments because when the two fragments belong to an image, pixel value change will be smooth in most of the cases. So this fact can be used to determine the adjacent fragments of a file.

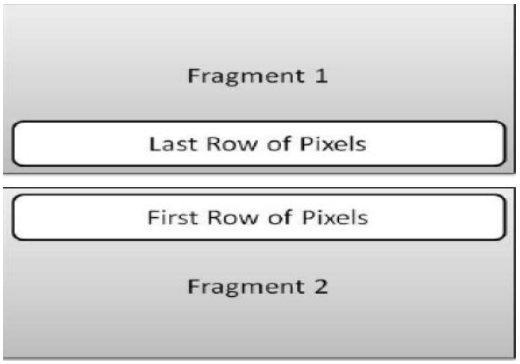

Fig 5 Comparing pixels' values in a fragment's last row with another fragment's first row [6]

For calculating candidate weights,corresponding pixels in the last row of the first fragment and the first row of second fragment are compared as shown in figure 6 and a value is found out.Three methods can be used for computing this particular value.An overview of these techniques is shown below.

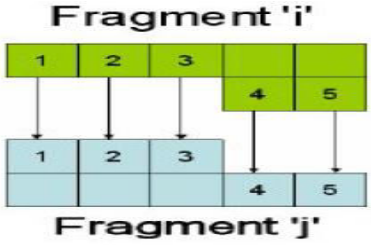

Fig 6 Pixel values being compared when calculating candidate weightsbetween two fragments

The techniques are [7]:

1)Pixel Matching (PM): This is a simple technique for determining adjacent fragments. Here the total numberof pixels matching along the edges of size w for the two fragments issummed. Suppose the width (w) is 5, PM compare 1 numberedpixel in fragment i with the same numbered pixel in fragmentj and the process is repeated for all boundary pixels. If the pixels matched, PM weight is incremented by one. For PM, if the weight is higher, the match will be better.

2)Sum of Differences (SoD): Calculate the RGB pixel values of two fragments. Extract the RGB pixel values in the last row of first fragment and first row of second fragment. Then calculate the sum of differences of these

values.SoDwould sum the absolute value of the difference between each numberedpixel in fragment i with the same numbered pixel in fragment j.ForSoD, if the weight is lower, the two fragments compared have more chances of being adjacent.

3)Median Edge Detection (MED): Pixel value is predicted from the value of thepixel above, to the left and left diagonal to it. Take the difference between the predicted value in fragment jand the actual value and repeat the process for all boundary pixels. Then compute the sum of these differences. For MED, the lower the weight the better the match.
SoD and MED techniques are more useful than PM.

VI SYSTEM DESIGN

The general flow of steps in developing the new carver that carves afterreassembling fragments is shown below.

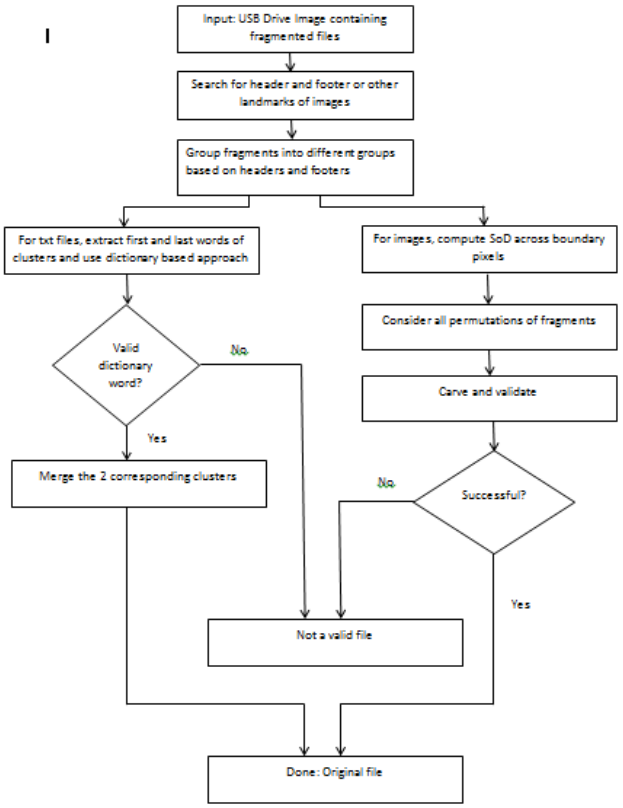

Fig 7 System design of carver that carve fragmented files

Steps can be summarized as follows. Carver is based on FAT32 file system found on USB drives. Disk image containing fragmented files is searched for headers and footers. Then fragments are grouped into different groups. In the case of txt files, first and last words of all clusters are stored in a database, form all possible combinations of last and first words and search in the dictionary. If it is a valid dictionary word, merge the two corresponding clusters. In the case of image files, find adjacent fragments from the permutations of all fragments by computing SoD across boundary pixels.

VII. COMPARISON TO EXISTING DATA RECOVERY METHODS

There are basically two types of data recovery methods-based on metadata and content.But metadata information of all files are concentrated in a particular area of disk.So if that area gets corrupted,then we will not be able to recover any files.In that case content based method is used.Existingcarving tools use header-footer carving.It does not look at blocks of data in between header and footer blocks.So existing carving tools does not carve fragmented files

This new method of carving fragmented files worked well in the case of fragmented image and text files.But since it examines all sectors from header to footer,time consumption is high compared to header-footer carvers.But with the advent of high speed processors,fragmented files can be recovered efficiently using this method.Also existing carvers cannot recover text files because of the absence of a specific header and footer.

VIII CONCLUSION AND FUTURE WORK

Traditional data recovery methods use metadata to recover deleted files.When a file is deleted,file system will simply mark file as deleted.File's metadata structures and clusters will reside in unallocated space until it is reallocated to other files.So even after deletion,with the use of this metadata files can be recovered.But in situations where metadata is corrupt or missing,we need an alternative technique.Moreover,it is possible that clusters of deleted file are reallocated before their metadata entries.In that case,we retrieve metadata of deleted file and determine allocated clusters of that file.But now content in that location belong to a new file. In short, metadata and content became out of sync.So in these situations, traditional methods give false results. Data carving can overcome these problems. It extracts deleted data from unallocated space based on content without using file system metadata.

Most of the current carving tools are based on header-footer carving. They are not useful in carving fragmented files. DFRWS challenge 2007 addressed the carving of fragmented files. So this paper discussed a method for developing a new carver that can carve fragmented document and image files.A dictionary based approach can be used in the case of fragmented .txt files. Header-Footer carving and brute force approach can be used for fragmented image files. To determine the best match, Sum of Differences (SoD) of boundary pixels of two fragments is determined. But the carver takes more processing time because of cluster by cluster analysis. Also in data carving,names of the carved files cannot be determined as

the metadata are not used .So in future; techniques that make use of both metadata and content to recover deleted files need to be devised. Data carving is based on guess work, so results may include many false hits especially in case of fragmented files. So use of many internal characteristics in addition to headers, footers, identifiers and size information must be entertained. Carver that reconstructs files with large number of fragments should be developed.

REFERENCES

[1]NadeemAlherbawi,ZarinaShukur,"Systematic literature review on data carving in digital forensic",The 4th International Conference on Electrical Engineering

[2] Christian Beek,"Introduction to File Carving",White paper byMcAfee Foundstone Professional Services

[3] Anandabrata Pal and NasirMemon, "The evolution of file carving," IEEE SignalProcessing Magazine, vol. 26, no. 2, pp. 59–71, 2009.

[4] Anandabrata Pal, KuleshShanmugasundaram and NasirMemon."Automated Reassemblyof Fragmented Images," Presented at ICASSP, 2003.

[5]KuleshShanmugasundaram and NasirMemon "Automatic reassembly of documentfragments via context based statistical models". In Proceedingsof the 19th Annual Computer Security Applications Conference, page152, 2003.

[6]AzzatAl-Sadi, Manaf Bin Yahya, Ahmad Almulhem,"Identification of image fragments for file carving", Internet Security (WorldCIS) World Congress,London, 2013

[7] Anandabrata Pal, NasirMemon, "Automated reassembly of file fragmentedimages using greedy algorithms", In: IEEE transactions onimage processing, February 2006

978-1-4673-8810-8/16/$31.00 ©2016 IEEE