

Guide to Computer Forensics and Investigations

Sixth Edition

Chapter 8

Recovering Graphics Files





Objectives

- Describe types of graphics file formats
- Explain types of data compression
- Explain how to locate and recover graphics files
- Describe how to identify unknown file formats
- Explain copyright issues with graphics



Recognizing a Graphics File

- Graphic files contain digital photographs, line art, three-dimensional images, text data converted to images, and scanned replicas of printed pictures
 - **Bitmap images:** collection of dots
 - **Vector graphics:** based on mathematical instructions
 - **Metafile graphics:** combination of bitmap and vector
- Types of programs
 - Graphics editors
 - Image viewers



Understanding Bitmap and Raster Images

- Bitmap images
 - Grids of individual **pixels**
- **Raster images** - also collections of pixels
 - Pixels are stored in rows
 - Better for printing
- Image quality
 - Screen **resolution** - determines amount of detail
 - Software contributes to image quality (drivers)
 - Number of color bits used per pixel



Understanding Vector Graphics

- Characteristics of vector graphics
 - Uses lines instead of dots
 - Store only the calculations for drawing lines and shapes
 - Smaller than bitmap files
 - Preserve quality when image is enlarged
- CorelDRAW, Adobe Illustrator



Understanding Metafile Graphics

- Metafile graphics combine raster and vector graphics
- Example
 - Scanned photo (bitmap) with text or arrows (vector)
- Share advantages and disadvantages of both types
 - When enlarged, bitmap part loses quality



Understanding Graphics File Formats (1 of 2)

- **Standard graphics file formats**

- Standard bitmap file formats

- Portable Network Graphic (.png)
 - Graphic Interchange Format (.gif)
 - Joint Photographic Experts Group (.jpeg, .jpg)
 - Tagged Image File Format (.tiff, .tif)
 - Window Bitmap (.bmp)

- Standard vector file formats

- Hewlett Packard Graphics Language (.hpgl)
 - Autocad (.dxr)



Understanding Graphics File Formats (2 of 2)

- Nonstandard graphics file formats

- Targa (.tga)
- Raster Transfer Language (.rtl)
- Adobe Photoshop (.psd) and Illustrator (.ai)
- Freehand (.fh11)
- Scalable Vector Graphics (.svg)
- Paintbrush (.pcx)



Understanding Digital Photograph File Formats (1 of 8)

- Witnesses or suspects can create their own digital photos
- Examining the raw file format
 - **Raw file format**
 - Referred to as a digital negative
 - Typically found on many higher-end digital cameras
 - Sensors in the digital camera simply record pixels on the camera's memory card
 - Raw format maintains the best picture quality



Understanding Digital Photograph File Formats (2 of 8)

- Examining the raw file format (cont'd)
 - The biggest disadvantage is that it's proprietary
 - And not all image viewers can display these formats
 - The process of converting raw picture data to another format is referred to as **demosaicing**
- Examining the Exchangeable Image File format
 - **Exchangeable Image File (Exif) format**
 - Commonly used to store digital pictures
 - Developed by JEITA as a standard for storing metadata in JPEG and TIF files

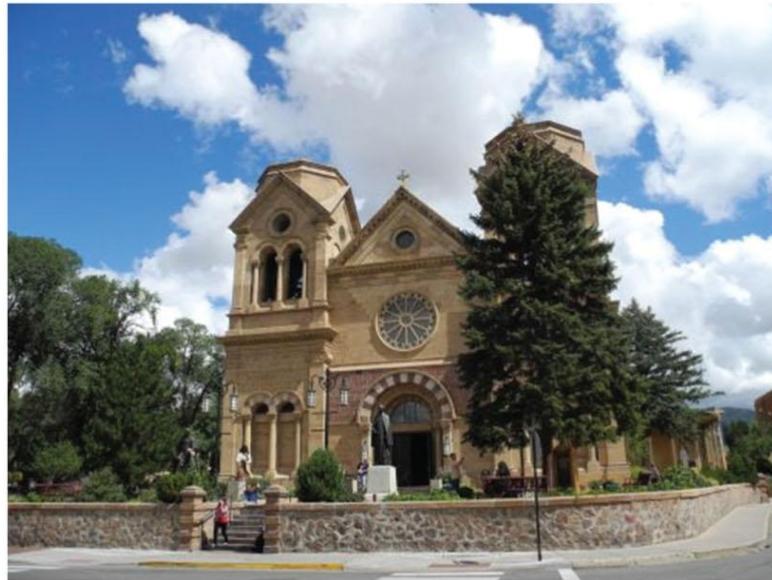


Understanding Digital Photograph File Formats (3 of 8)

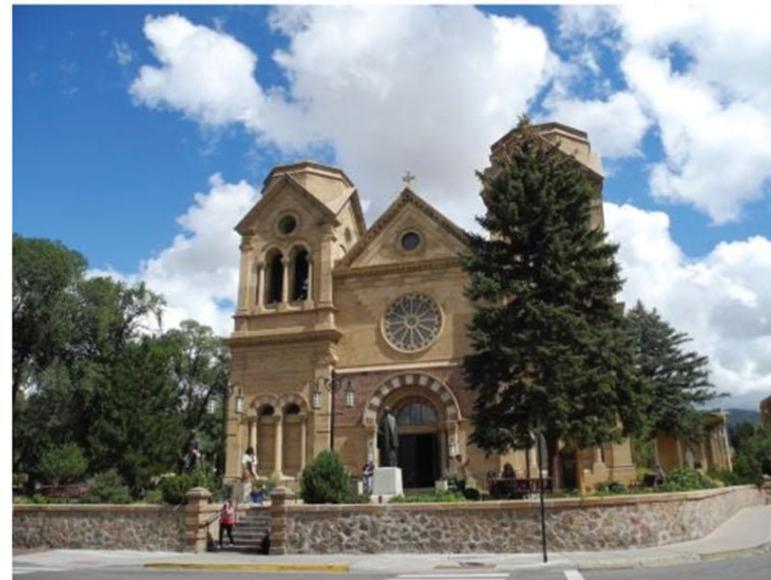
- Examining the Exchangeable Image File format (cont'd)
 - Exif format collects metadata
 - Investigators can learn more about the type of digital device and the environment in which photos were taken
 - Viewing an Exif JPEG file's metadata requires special programs
 - Exif Reader, IrfanView, or Magnet Forensics AXIOM
 - Exif file stores metadata at the beginning of the file



Understanding Digital Photograph File Formats (4 of 8)



Exif picture file



JPEG picture file

Figure 8-1 Similar Exif and JPEG photos



Understanding Digital Photograph File Formats (5 of 8)

Figure 8-2 Differences in Exif and JPEG file header information

Source: X-Ways AG, www.x-ways.net



Understanding Digital Photograph File Formats (6 of 8)

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Hex Dump	Description
00019110	00	F8	B5	E0	F9	5B	A7	F4	AC	AC	A0	0F	7F	E9	05	BE	00 19110 [Sô-- é %	
00019120	7F	FC	62	F8	7F	F9	0B	FD	2A	1B	AE	D0	A2	C0	C6	1D	übø ù ý* @DøÅE	
00019130	5A	FE	49	71	E1	5D	8A	A0	9F	9E	2B	2B	28	02	AD	77	ZþIqá] I ++(-w	
00019140	78	D7	32	BC	B7	0C	EE	EF	F7	99	8F	26	A1	EF	95	73	xx2%· ii+I &iiS	
00019150	B4	64	7B	81	59	59	4C	46	C2	75	C7	DD	15	82	E3	3D	'd{ VYLFÅuçÝ ä=	
00019160	16	B2	B2	80	08	D3	F5	19	6C	AF	62	B8	87	EF	C6	D9	²²€ Ø§ l~b, iÆÙr	
00019170	C7	A8	F3	1F	51	5D	4A	3B	D1	3C	09	22	16	D8	EA	18	Ç'ó Q]J;Ñ ^c " Øè	
00019180	67	D0	8A	CA	CA	00	A1	76	AA	C8	D8	DD	89	60	20	41	gÐIÉE iv³ÉØÝ` A	
00019190	36	48	5F	F0	9F	31	F2	A4	5B	DF	FC	46	B2	B2	80	21	6H_ë lð¤[BüF²² !	
000191A0	37	60	1C	17	23	E9	5B	AC	84	91	86	AC	AC	A0	0D	BC	7` #é[- ~` %	
000191B0	47	D7	F1	AD	79	27	CC	7D	6B	2B	28	03	C2	39	C6	6B	Gxñ-y'í}k+(Å9Æk	
000191C0	30	3D	7F	2A	CA	CA	00	F3	03	1E	75	E6	D1	E8	2B	2B	0= *ÉE ó uæÑè++	
000191D0	28	03	49	19	63	5C	95	15	E2	C9	BB	EE	2E	01	F5	35	(I c\ áÉ>i. 65	
000191E0	95	94	08	90	92	46	09	3F	4A	D7	38	E8	2B	2B	28	02	'F ?Jx8è++(
000191F0	32	E7	38	C5	6A	4F	AF	E9	59	59	40	11	93	CD	65	2ç8Åj0-éYY@ íee		
00019200	65	00	7F	FF	D9												e ýÙ	
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Hex Dump	Description
00001ED0	92	49	00	24	92	49	00	24	92	49	00	3A	9E	A3	C5	5A	' I \$' I \$' I :fÅZ	
00001EE0	A8	E1	1A	AA	6E	51	07	C5	C2	00	6F	1A	A3	9E	93	87	'á ñQ ÁÁ o f	
00001EF0	43	F2	5E	6F	86	EC	F4	D4	21	D3	E6	BD	1E	AD	72	87	Cð^o iðÔ!Óøk -r	
00001F00	56	A7	72	63	CD	2E	CE	86	57	D8	3F	01	C3	5B	45	B6	VSrcí.Í WØ? Á[E	
00001F10	12	ED	91	36	B6	18	D1	37	26	66	53	18	35	E8	A3	AB	i'6M N7&fs 5éé<	
00001F20	75	89	C8	D6	C7	82	49	89	BE	CA	60	22	35	54	19	5C	u ÉÖç I ¾É "5T \	
00001F30	03	28	9D	33	9B	45	19	2B	8F	25	8A	34	5A	6E	2C	77	(3 E + % 4Zn,w	
00001F40	5A	FE	CF	53	86	2C	7B	24	5D	6D	7B	37	7A	40	A7	FA	ZþÍS .{\$]m{7z@Sú	
00001F50	7E	C5	5A	B0	83	4C	09	E1	45	29	ED	5A	CC	C4	81	25	~Å° L áE)iZIÄ %	
00001F60	C6	94	94	A0	63	80	5D	01	36	57	42	00	EA	E8	0B	90	Æ c] 6WB éè	
00001F70	9C	14	80	A1	25	D4	94	01	8C	49	24	94	12	03	5C	49	I i%O IS \I	
00001F80	24	00	92	49	24	00	92	49	24	00	9C	A0	7A	E2	48	02	\$ 'IS 'IS zåH	
00001F90	2A	8A	0A	FA	24	92	5D	9D	0C	AF	B2	3A	7A	15	13	F4	* ús' -²:z ô	
00001FA0	29	24	B0	4B	B3	5B	2B	3B	64	47	87	EA	12	49	4C	BA)\$^K'[+;dGé ILº	
00001FB0	2B	E0	26	DD	96	C3	B3	7F	E9	04	92	5A	3D	3F	91	57	+à&Ý Ä³ é 'Z=?'W	
00001FC0	74	15	4F	6A	EA	4B	62	32	B1	C1	75	71	25	08	96	75	t OjéKb2±Auq% u	
00001FD0	3D	89	24	80	1C	92	49	20	OE	A4	92	4A	40	FF	D9	= S 'I m'J@yÙ		

Figure 8-3 EOI marker FFD9 for all JPEG files

Source: X-Ways AG, www.x-ways.net



Understanding Digital Photograph File Formats (7 of 8)

- Examining the Exchangeable Image File format (cont'd)
 - With tools such as Autopsy and Exif Reader
 - You can extract metadata as evidence for your case



Understanding Digital Photograph File Formats (8 of 8)

The screenshot shows the Autopsy interface displaying file metadata. At the top, there is a 'Directory Listing' table with columns: Source File, Date Created, Device Model, Device Make, Data Source, and Tags. Two entries are listed: 'IMG_1345.jpg' and 'Odessey11.txt'. Below the table, there are tabs for Hex, Strings, File Metadata, Results, Indexed Text, and Media. The 'File Metadata' tab is selected, showing detailed information for 'IMG_1345.jpg':

Name	/img_K:/Homework/IMG_1345.jpg
Type	File System
MIME Type	image/jpeg
Size	291877
File Name Allocation	Allocated
Metadata Allocation	Allocated
Modified	2017-07-10 17:50:56 PDT
Accessed	2017-07-10 00:00:00 PDT
Created	2017-07-10 17:55:54 PDT
Changed	0000-00-00 00:00:00
MD5	Not calculated
Hash Lookup Results	UNKNOWN
Internal ID	124

Below this, there is a section titled 'From The Sleuth Kit istat Tool:' containing raw directory entry and sector information:

```
Directory Entry: 47535
Allocated
File Attributes: File, Archive
Size: 291877
Name: IMG_1345.jpg

Directory Entry Times:
Written: 2017-07-10 17:50:56 (PDT)
Accessed: 2017-07-10 00:00:00 (PDT)
Created: 2017-07-10 17:55:54 (PDT)

Sectors:
10624 10625 10626 10627 10628 10629 10630 10631
10632 10633 10634 10635 10636 10637 10638 10639
10640 10641 10642 10643 10644 10645 10646 10647
```

Figure 8-4 Autopsy displaying metadata from an Exif JPEG file

Source: www.sleuthkit.org



Understanding Data Compression

- Most graphics file formats compress their data
 - GIF and JPEG
- Others, like BMP, do not compress their data
 - Use data compression tools for those formats
- **Data compression**
 - Coding data from a larger to a smaller form
 - Types
 - Lossless compression and lossy compression



Lossless and Lossy Compression

- **Lossless compression**

- Reduces file size without removing data
- Based on Huffman or Lempel-Ziv-Welch coding
 - For redundant bits of data
- Utilities: WinZip, PKZip, StuffIt, and FreeZip

- **Lossy compression**

- Permanently discards bits of information
- **Vector quantization (VQ)**
 - Determines what data to discard based on vectors in the graphics file
- Utility: Lzip



Locating and Recovering Graphics Files

- Operating system tools
 - Time consuming
 - Results are difficult to verify
- Digital forensics tools
 - Image headers
 - Compare them with good header samples
 - Use header information to create a baseline analysis
 - Reconstruct fragmented image files
 - Identify data patterns and modified headers



Identifying Graphics File Fragments

- **Carving or salvaging**
 - Recovering any type of file fragments
- Digital forensics tools
 - Can carve from file slack and free space
 - Help identify image files fragments and put them together



Repairing Damaged Headers (1 of 4)

- When examining recovered fragments from files in slack or free space
 - You might find data that appears to be a header
- If header data is partially overwritten, you must reconstruct the header to make it readable
 - By comparing the hexadecimal values of known graphics file formats with the pattern of the file header you found



Repairing Damaged Headers (2 of 4)

- Each graphics file has a unique header value
- Example:
 - A JPEG file has the hexadecimal header value FFD8, followed by the label JFIF for a standard JPEG or Exif file at offset 6
- Exercise:
 - Investigate a possible intellectual property theft by a new employee of Superior Bicycles, Inc.



Repairing Damaged Headers (3 of 4)

Chris Robinson

From: Bob Aspen <b_aspen@aol.com>
Sent: Monday, July 10, 2017 3:32 PM
To: cr-superior@outlook.com
Subject: FW: More info

Chris,
I got cc'd this odd message from Terry Sadler.
Do you have any projects that might need some capital investment?
Bob

-----Original Message-----

From: Terry Sadler [mailto:t_sadler@zoho.com]
Sent: Monday, July 10, 2017 3:28 PM
To: Jim Shu
Subject: Re: More info

Do you have a name for the project?

On 7/10/2017 3:04 PM, Jim Shu wrote:

> Terry,
>
> Here a few more photos from Tom.
>
> How much you willing to pay for these?
>
> Jim
>

Figure 8-5 An e-mail from Terry Sadler



Repairing Damaged Headers (4 of 4)

Chris Robinson

From: Tom Johnson <1060waddisonst@gmx.us>
Sent: Monday, July 10, 2017 2:40 PM
To: Jim Shu
Subject: You might be interested

Jim,

I had a tour of the new kayak factory. I think we can run with this to the other party interested in competing. I smuggled these files out, they are JPEG files I edited with my hex editor so that the email monitor won't pick up on them. So to view them you have to re-edit each file to the proper JPEG header of offset 0x FF D8 FF E0 and offset 6 of 4A. Then you have to rename them to a .jpg extension to view them.

Tom

Figure 8-6 The e-mail with attachments IT found



Searching for and Carving Data from Unallocated Space (1 of 6)

- Steps
 - Planning your examination
 - Searching for and recovering digital photograph evidence
 - Use Autopsy for Windows to search for and extract (recover) possible evidence of JPEG files
 - False hits are referred to as **false positives**



Searching for and Carving Data from Unallocated Space (2 of 6)

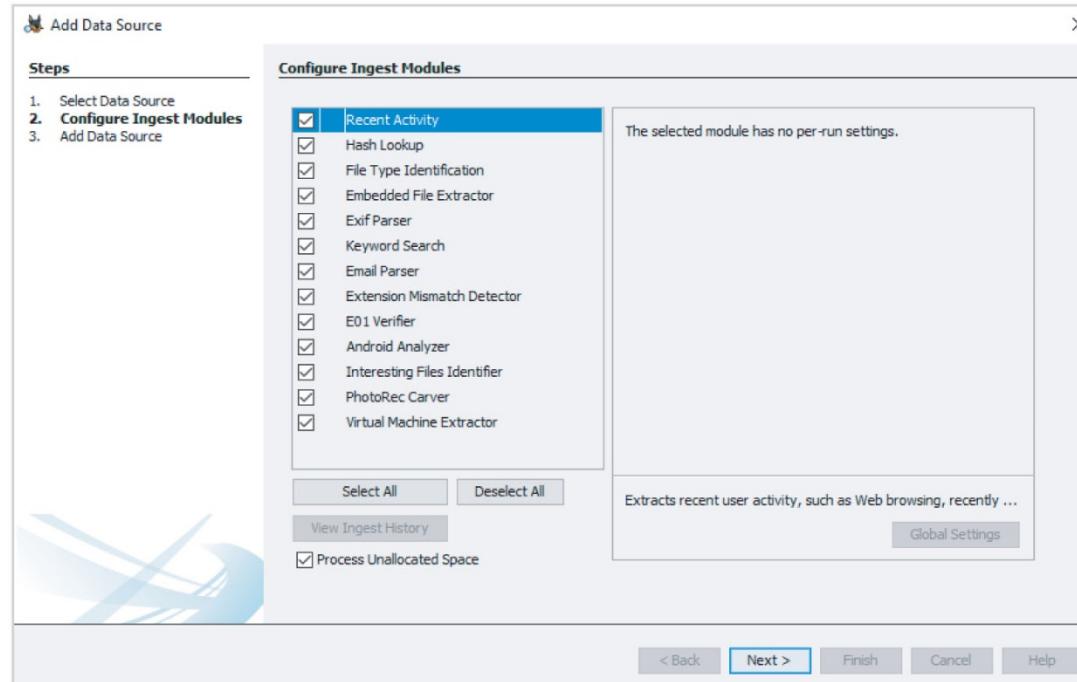


Figure 8-7 Processing options in the Configure Ingest Modules window

Source: www.sleuthkit.org



Searching for and Carving Data from Unallocated Space (3 of 6)

The screenshot shows the Autopsy 4.3.0 interface. The left sidebar displays a tree view of data sources, views, results (including extracted content, keyword hits, and interesting items), tags, and reports. The main pane shows a 'Directory Listing' of extracted content, specifically EXIF metadata files. A table lists five files: '_MG_1345.jpg', 'Odessey11.txt', '_RBATIM9.jpg', '_R2WHGRN.txt', and 'f0006352.jpg'. The columns include Source File, Date Created, Device Model, Device Make, Data Source, and Tags. Below the table is a hex editor window showing the raw file data for '_MG_1345.jpg', which includes the JFIF header and EXIF metadata.

Source File	Date Created	Device Model	Device Make	Data Source	Tags
_MG_1345.jpg	2009-04-05 18:39:04 PDT	Canon PowerShot SD870 IS	Canon	C08InChp.dd	
Odessey11.txt	2001-08-07 11:50:49 PDT	Dimage 2330 Zoom	Minolta Co., Ltd	C08InChp.dd	
_RBATIM9.jpg	2009-04-05 18:39:04 PDT	Canon PowerShot SD870 IS	Canon	C08InChp.dd	
_R2WHGRN.txt	2001-08-07 11:50:49 PDT	Dimage 2330 Zoom	Minolta Co., Ltd	C08InChp.dd	
f0006352.jpg	2009-04-05 18:39:04 PDT	Canon PowerShot SD870 IS	Canon	C08InChp.dd	

Hex Strings File Metadata Results Indexed Text Media
Page: 1 of 18 Page Go to Page: Jump to Offset: 0

```
0x00000000: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 60 .....JFIF.....
0x00000010: 00 00 00 FF E1 22 70 45 78 69 66 00 00 49 45 ....."pExif..II
0x00000020: 2A 00 08 00 00 00 02 00 0F 01 02 00 06 00 00 49 *.....
0x00000030: 86 00 00 00 10 01 02 00 19 00 00 00 8C 00 00 00 .....
0x00000040: 12 01 03 00 01 00 00 00 00 00 00 00 1A 01 05 00 .....
0x00000050: 01 00 00 A6 00 00 00 28 01 03 00 01 00 00 00 .....(....
0x00000060: 02 00 00 00 1B 01 05 00 01 00 00 00 AE 00 00 00 .....
0x00000070: 31 01 02 00 08 00 00 00 B6 00 00 00 32 01 02 00 1.....2...
0x00000080: 14 00 00 00 C2 00 00 00 13 02 03 00 01 00 00 00 .....
0x00000090: 01 00 00 00 69 87 43 04 01 00 00 00 D6 00 00 00 .....i....
0x000000A0: B0 0D 00 00 43 61 6E 6F 6E 00 43 61 6E 6F 6E 20 ....Canon.Canon
0x000000B0: 50 6F 77 65 72 53 68 6F 74 20 53 44 38 37 30 20 PowerShot SD870
0x000000C0: 49 53 00 00 B4 00 00 00 01 00 00 00 B4 00 00 00 IS...
0x000000D0: 01 00 00 00 50 69 63 61 73 61 20 33 2E 30 00 00 ...Picasa 3.0..
0x000000E0: 32 30 30 35 3A 30 34 3A 30 35 20 31 38 3A 33 35 2009-04-05 18:39
0x000000F0: 3A 30 34 00 21 00 5A 82 05 00 01 00 00 00 68 02 :04!....h.
0x00000100: 00 00 9D 82 05 00 01 00 00 70 02 00 00 27 88 .....p...
0x00000110: 03 00 01 00 00 00 7D 00 00 00 90 07 04 00 .....}.....
0x00000120: 00 00 30 32 32 30 03 90 02 00 14 00 00 00 78 02 ..0220...x.
0x00000130: ..00 00 04 00 00 00 14 00 00 00 05 07 00 00 01 61
```

Figure 8-8 Parsing Exif metadata in Autopsy

Source: www.sleuthkit.org



Searching for and Carving Data from Unallocated Space (4 of 6)

The screenshot shows the Autopsy 4.3.0 interface with the title bar "C08InChp - Autopsy 4.3.0". The main window displays a table of search results for the keyword "fif". The table has columns for Name, Location, Modified Time, Change Time, Access Time, Created Time, Size, Flags(Dir), Flags(Meta), Mode, and UserID. The results include various files such as "f0001008.txt", "RJNGOUQ.txt", "RANH8EF.txt", "gametour2.exe", "gametour3.exe", "f0001265.txt", "gametour4.exe", "f0001364.txt", ".5.xls", "f0001447.txt", "NTONY-1.TXT", and "f0001658.txt". Below the table, there is a hex dump of the file "f0001658.txt" starting at offset 0.

Name	Location	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Mode	User ID
f0001008.txt	/img_C08InChp.dd CarvedFiles/f0001008.txt	2000-01-10 00:00:00	2000-01-10 00:00:00	2000-01-10 00:00:00	2000-01-10 00:00:00	20316	Unallocated	Unallocated	r-----	0
RJNGOUQ.txt	/img_C08InChp.dd R/RECYCLE.BIN/_RJNGOUQ.txt	2007-02-01 13:09:18 PST	2000-01-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:54:13 PDT	23715	Unallocated	Unallocated	rwxrwxrwx	0
RANH8EF.txt	/img_C08InChp.dd R/RECYCLE.BIN/_RANH8EF.txt	2007-02-01 13:13:54 PST	2000-01-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:54:13 PDT	2034	Unallocated	Unallocated	rwxrwxrwx	0
gametour2.exe	/img_C08InChp.dd/Vacation Pictures/gametour2.exe	2001-08-05 07:50:24 PDT	2000-01-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:39 PDT	206281	Allocated	Allocated	rwxrwxrwx	0
gametour3.exe	/img_C08InChp.dd/Vacation Pictures/gametour3.exe	2001-08-07 04:51:44 PDT	2000-01-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:39 PDT	251101	Allocated	Allocated	rwxrwxrwx	0
f0001265.txt	/img_C08InChp.dd CarvedFiles/f0001265.txt	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	49715	Unallocated	Unallocated	r-----	0
gametour4.exe	/img_C08InChp.dd/Vacation Pictures/gametour4.exe	2001-08-08 08:23:54 PDT	2000-01-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:39 PDT	18115	Allocated	Allocated	rwxrwxrwx	0
f0001364.txt	/img_C08InChp.dd CarvedFiles/f0001364.txt	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	24294	Unallocated	Unallocated	r-----	0
.5.xls	/img_C08InChp.dd/Accounts/GovernmentDATA/.5.xls	2007-02-01 13:33:28 PST	2000-01-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:53:53 PDT	40448	Allocated	Allocated	rwxrwxrwx	0
f0001447.txt	/img_C08InChp.dd CarvedFiles/f0001447.txt	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	17808	Unallocated	Unallocated	r-----	0
NTONY-1.TXT	/img_C08InChp.dd CarvedFiles/NTONY-1.TXT	2007-02-01 19:51:50 PST	2000-01-00 00:00:00	2007-02-01 19:51:50 PST	2007-02-06 15:20:57 PST	17932	Unallocated	Unallocated	rwxrwxrwx	0
f0001658.txt	/img_C08InChp.dd CarvedFiles/f0001658.txt	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	2000-01-00 00:00:00	23671	Unallocated	Unallocated	r-----	0

Hex Strings File Metadata Results Indexed Text Media

Page: 1 of 18 Page Go to Page: Jump to Offset: 0

```
0x00000000: FF D0 FF E0 00 10 4A 46 49 46 00 01 01 00 E0 ...JFIF...`.  
0x00000010: 00 E0 00 00 FF E1 22 70 45 78 65 66 00 49 49 ...dExif..II  
0x00000020: 2A 00 08 00 00 00 0A 00 07 01 02 00 06 00 00 00 ...  
0x00000030: 8E 00 00 00 10 01 02 00 19 00 00 00 8C 00 00 00 ...  
0x00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...  
0x00000050: 01 00 00 A6 00 00 00 00 01 03 00 01 00 00 00 00 ...  
0x00000060: 02 00 00 00 1B 01 05 00 01 00 00 00 A8 00 00 00 ...  
0x00000070: 3E 01 02 00 08 00 00 00 B6 00 00 00 32 01 02 00 1.....2...  
0x00000080: 1E 01 00 00 C2 00 00 00 13 02 03 00 01 00 00 00 ...  
0x00000090: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...  
0x000000A0: B0 0D 00 00 43 E1 E8 00 43 E1 E8 E7 E2 20 ...Canon.Canon  
0x000000B0: 50 E7 77 E5 72 53 E9 E7 74 20 53 44 38 37 30 20 PowerShot SD870  
0x000000C0: 4E 53 00 00 B4 00 00 00 01 00 00 00 B6 00 00 00 IS...Picasa 3.0..  
0x000000D0: 01 00 00 00 50 E9 E1 73 E1 20 33 2E 30 00 00 ...
```

Figure 8-9 The results of searching for "fif"

Source: www.sleuthkit.org



Searching for and Carving Data from Unallocated Space (5 of 6)

File header overwritten with zzzz

The screenshot shows the Autopsy 4.3.0 interface with the title bar "C08InChp - Autopsy 4.3.0". The menu bar includes Case, View, Tools, Window, Help, Add Data Source, View Images/Videos, Timeline, Generate Report, and Close Case. The left sidebar shows "Data Sources", "Views", "Results", "Extracted Content" (with "EXIF Metadata (5)" selected), "Keyword Hits" (with "Single Literal Keyword Search (0)", "Single Regular Expression Search (166)", and "Email Addresses (7)" listed), "Hashset Hits", "E-Mail Messages", "Interesting Items", "Accounts", "Tags", and "Reports". The main area displays a "Directory Listing" table with columns: Name, Location, Modified Time, and Change Time. A specific file, "gametour2.exe", is highlighted in blue. Below the table is a "Hex" view showing the file's content. The hex dump starts with the byte sequence 0x00000000: 7A 7A 7A 7A 00 10 7A 46 followed by several lines of binary data and their ASCII representations. A vertical line with an arrow points from the "File header overwritten with zzzz" text at the top down to the start of the hex dump.

Name	Location	Modified Time	Change Time
f0001008.txt	/img_C08InChp.dd/\$CarvedFiles/f0001008.txt	0000-00-00 00:00:00	0000-00-00 00:00:00
_RJNGOUQ.txt	/img_C08InChp.dd/\$RECYCLE.BIN/_RJNGOUQ.txt	2007-02-01 13:09:18 PST	0000-00-00 00:00:00
Odessey20.txt	/img_C08InChp.dd/Homework/Odessey20.txt	2007-02-01 13:13:54 PST	0000-00-00 00:00:00
_RANH86F.txt	/img_C08InChp.dd/\$RECYCLE.BIN/_RANH86F.txt	2007-02-01 13:11:50 PST	0000-00-00 00:00:00
gametour2.exe	/img_C08InChp.dd/Vacation Pictures/gametour2.exe	2001-08-05 07:50:24 PDT	0000-00-00 00:00:00
gametour3.exe	/img_C08InChp.dd/Vacation Pictures/gametour3.exe	2001-08-07 04:51:44 PDT	0000-00-00 00:00:00
f0001265.txt	/img_C08InChp.dd/\$CarvedFiles/f0001265.txt	0000-00-00 00:00:00	0000-00-00 00:00:00
gametour4.exe	/img_C08InChp.dd/Vacation Pictures/gametour4.exe	2001-08-08 08:23:54 PDT	0000-00-00 00:00:00
f0001364.txt	/img_C08InChp.dd/\$CarvedFiles/f0001364.txt	0000-00-00 00:00:00	0000-00-00 00:00:00
_S.XLS	/img_C08InChp.dd/Accounts/GovernmentDATA/_S.XLS	2007-02-01 13:33:28 PST	0000-00-00 00:00:00
f0001447.txt	/img_C08InChp.dd/\$CarvedFiles/f0001447.txt	0000-00-00 00:00:00	0000-00-00 00:00:00
_NTONY+1.TXT	/img_C08InChp.dd/\$OrphanFiles/_NTONY+1.TXT	2007-02-01 19:51:50 PST	0000-00-00 00:00:00
f0001658.txt	/img_C08InChp.dd/\$CarvedFiles/f0001658.txt	0000-00-00 00:00:00	0000-00-00 00:00:00

Hex Strings File Metadata Results Indexed Text Media
Page: 1 of 13 Page Go to Page: Jump to Offset: 0

```
0x00000000: 7A 7A 7A 7A 00 10 7A 46 49 46 00 01 01 01 00 78 zzzz..zFIF.....x
0x00000010: 00 78 00 00 FF E1 03 1C 45 78 69 66 00 00 49 49 .x...Exif..II
0x00000020: 2A 00 08 00 00 00 0B 00 0E 01 02 00 0A 00 00 00 *
0x00000030: 92 00 00 00 0F 01 02 00 12 00 00 00 9C 00 00 00 .....
0x00000040: 10 01 02 00 12 00 00 00 AE 00 00 00 12 01 03 00 ...
0x00000050: 01 00 00 00 01 00 08 00 1A 01 05 00 01 00 00 00 ...
0x00000060: C0 00 00 00 1B 01 05 00 01 00 00 00 C8 00 00 00 ...
0x00000070: 28 01 03 00 01 00 00 00 02 00 97 02 31 01 02 00 (...).1...
0x00000080: 0A 00 00 00 D0 00 00 00 32 01 02 00 14 00 00 00 ...
0x00000090: DA 00 00 00 13 00 03 00 01 00 00 00 02 00 00 02 ...
0x000000a0: 52 00 00 00 01 00 00 00 02 00 00 00 00 00 00 00 ...
0x000000b0: 20 20 20 20 20 20 20 20 20 00 4D 69 EE EF EC 74 ...Miscel...
0x000000c0: 61 20 43 6F 2C 2C 4C 74 64 20 44 45 ED 61 a Co., Ltd. Disc...
0x000000d0: 67 65 20 32 33 33 30 20 5A 67 6F DF 20 00 49 00 ge 2330 Zone H.
0x000000e0: 00 00 01 00 00 00 48 00 00 00 01 00 00 00 20 20 .....H.....
0x000000f0: 20 20 20 20 20 20 20 20 32 30 30 31 3A 30 38 3A ..2001:08:...
0x00000100: 30 35 20 31 34 3A 35 30 3A 30 37 00 10 00 27 88 08 14:50:07...
0x00000110: 03 00 04 00 00 00 B4 01 00 00 00 90 07 00 04 00 ...
0x00000120: 00 00 30 32 31 30 03 90 02 00 14 00 00 00 BC 01 ..0210.....
```

Figure 8-10 The altered file header

Source: www.sleuthkit.org



Searching for and Carving Data from Unallocated Space (6 of 6)

The screenshot shows the C0BnChp - Autopsy 4.3.0 interface. The top navigation bar includes 'Case', 'View', 'Tools', 'Window', 'Help', 'Add Data Source', 'View Images/Videos', 'Timeline', 'Generate Report', and 'Close Case'. A search bar at the top right is set to 'Keyword Lists' with the query '2 -ff'. The main pane displays a list of files from a directory listing, including their names, locations, modified times, change times, access times, creation times, sizes, and flags. A search bar at the top left is set to 'Keyword search' with the query '2 -ff'. The bottom pane shows detailed information for the selected file 'semateur.exe'.

Name	Location	Modified Time	Change Time	Access Time	Created Time	Size	Flags(DR)	Flags(Meta)	Mode	User ID
file0008.txt	/img_C0BnChp_dd\%CarvedFiles\0001008.txt	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	20316	Unallocated	Unallocated	r-----	0
JUNGOV.bzt	/img_C0BnChp_dd\%RECYLE.BIN\JUNGOV.bzt	2007-02-01 13:09:18 PST	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:54:13 PDT	27315	Unallocated	Unallocated	mrwörwörx	0
Odyssey20.bzt	/img_C0BnChp_dd\%RECYCLE.BIN\Odyssey20.bzt	2007-02-01 13:13:54 PST	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:54:13 PDT	20316	Unallocated	Unallocated	mrwörwörx	0
JAN-B9F.bzt	/img_C0BnChp_dd\%RECYCLE.BIN\JAN-B9F.bzt	2007-02-01 13:11:50 PST	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:54:13 PDT	23671	Unallocated	Unallocated	mrwörwörx	0
Semateur2.exe	/img_C0BnChp_dd\%Vocation Pictures\Semateur2.exe	2001-09-05 07:50:24 PDT	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:39 PDT	206281	Allocated	Allocated	mrwörwörx	0
Semateur3.exe	/img_C0BnChp_dd\%Vocation Pictures\Semateur3.exe	2001-09-07 09:51:44 PDT	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:39 PDT	251101	Allocated	Allocated	mrwörwörx	0
file001265.txt	/img_C0BnChp_dd\%CarvedFiles\0001265.txt	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	49715	Unallocated	Unallocated	r-----	0
semateur.exe	/img_C0BnChp_dd\%Vocation Pictures\Semateur.exe	2001-08-06 00:23:54 PDT	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:39 PDT	181115	Allocated	Allocated	mrwörwörx	0
file001364.txt	/img_C0BnChp_dd\%CarvedFiles\0001364.txt	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	24294	Unallocated	Unallocated	r-----	0
.XLS	/img_C0BnChp_dd\%Accounts\GovernmentDATA_5.XLS	2007-02-01 13:33:28 PST	2000-00-00 00:00:00	2017-07-10 00:00:00 PDT	2017-07-10 17:55:33 PDT	40488	Allocated	Allocated	mrwörwörx	0
file001447.txt	/img_C0BnChp_dd\%CarvedFiles\0001447.txt	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	17888	Unallocated	Unallocated	r-----	0
NTONY-1.TXT	/img_C0BnChp_dd\%CarvedFiles\NTONY-1.TXT	2007-02-01 19:51:50 PST	2000-00-00 00:00:00	2007-02-06 00:00:00 PDT	2007-02-06 15:20:57 PST	179032	Unallocated	Unallocated	mrwörwörx	0
file001658.txt	/img_C0BnChp_dd\%CarvedFiles\0001658.txt	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	2000-00-00 00:00:00	23671	Unallocated	Unallocated	r-----	0

Figure 8-11 Viewing all sectors used by the gametour2.exe file

Source: www.sleuthkit.org



Rebuilding File Headers (1 of 6)

- Before attempting to edit a recovered graphics file
 - Try to open the file with an image viewer first
 - If the image isn't displayed, you have to inspect and correct the header values manually
 - Steps
 - Recover more pieces of file if needed
 - Examine file header
 - Compare with a good header sample
 - Manually insert correct hexadecimal values
 - Test corrected file



Rebuilding File Headers (2 of 6)

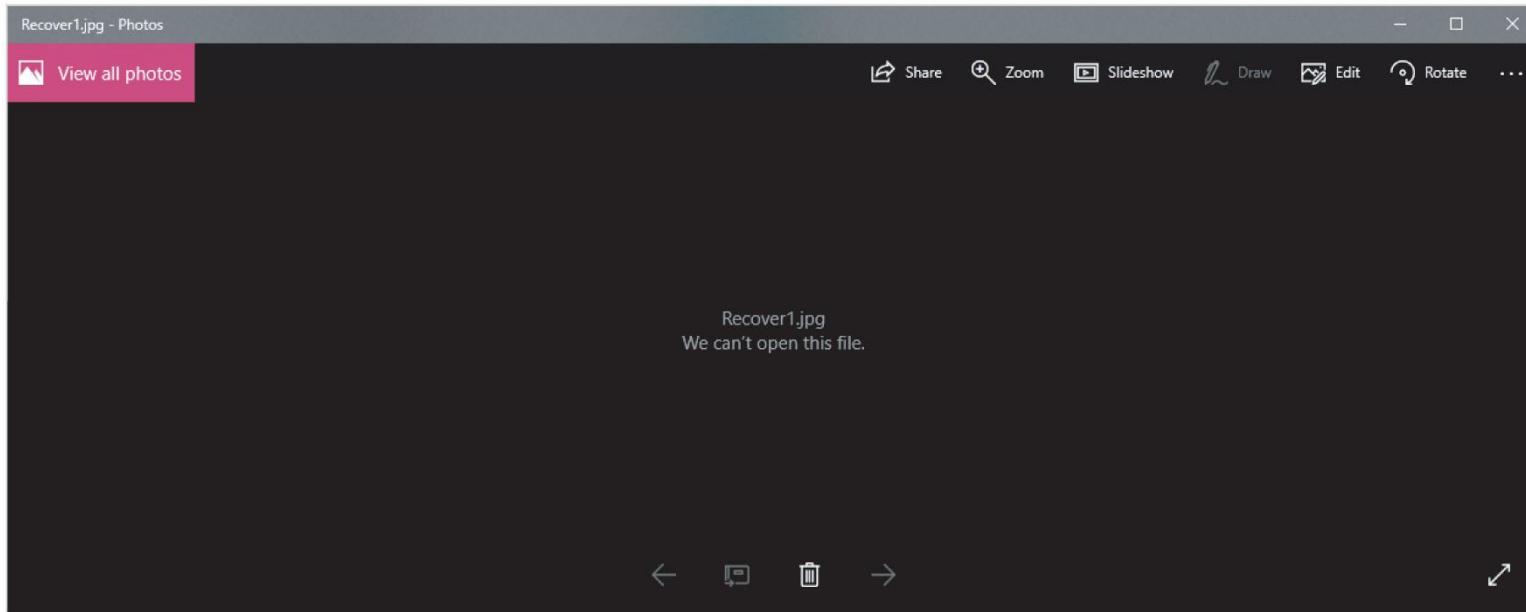


Figure 8-12 Error message indicating a damaged or an altered graphics file



Rebuilding File Headers (3 of 6)

The screenshot shows the WinHex application interface with the file 'Recover1.jpg' open. The left pane displays the hex editor with columns for Offset, Hex, ASCII, and Hex+ASCII. The right pane shows the file's metadata and preview. Two arrows point to specific offsets: 'Offset position 0' points to the first byte (0x7A) in the hex view, and 'Offset position 6' points to the byte at offset 6 (0x49).

WinHex - [Recover1.jpg]

File Edit Search Position View Tools Specialist Options Window Help

Case Data

Recover1.jpg

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	7A	7A	7A	7A	00	10	7A	46	49	46	00	01	01	01	00	78
00000010	00	78	00	00	FF	E1	03	1C	45	78	69	66	00	00	49	49
00000020	2A	00	08	00	00	00	0B	00	0E	01	02	00	0A	00	00	00
00000030	92	00	00	00	0F	01	02	00	12	00	00	00	9C	00	00	00
00000040	10	01	02	00	12	00	00	00	AE	00	00	00	12	01	03	00
00000050	01	00	00	00	01	00	FF	FF	1A	01	05	00	01	00	00	00
00000060	C0	00	00	00	1B	01	05	00	01	00	00	00	C8	00	00	00
00000070	28	01	03	00	01	00	00	00	02	00	FF	FF	31	01	02	00
00000080	0A	00	00	00	D0	00	00	00	32	01	02	00	14	00	00	00
00000090	D4	00	00	00	13	02	03	00	01	00	00	00	02	00	FF	FF
000000A0	69	87	04	00	01	00	00	00	EE	00	00	00	00	00	00	00
000000B0	20	20	20	20	20	20	20	20	00	4D	69	6E	6C	74		
000000C0	61	20	43	6F	2E	2C	20	4C	74	64	20	00	44	69	6D	61
000000D0	67	65	20	32	33	33	30	20	5A	6F	6F	6D	20	00	48	00
000000E0	00	00	01	00	00	00	48	00	00	00	01	00	00	00	20	20
000000F0	20	20	20	20	20	20	20	20	32	30	30	31	3A	30	38	3A
00000100	30	38	20	31	35	3A	32	33	3A	35	34	00	10	00	27	88
00000110	03	00	04	00	00	00	B4	01	00	00	00	90	07	00	04	00
00000120	00	00	30	32	31	30	03	90	02	00	14	00	00	BC	01	
00000130	00	00	04	90	02	00	14	00	00	00	D0	01	00	01	91	
00000140	07	00	04	00	00	00	01	02	03	00	02	91	05	00	01	00
00000150	00	00	E4	01	00	00	01	92	0A	00	01	00	00	EC	01	
00000160	00	00	02	92	05	00	01	00	00	00	F4	01	00	00	04	92
00000170	0A	00	01	00	00	00	FC	01	00	00	09	92	03	00	01	00
00000180	00	00	00	00	FF	FF	0A	92	05	00	01	00	00	04	02	
00000190	00	00	7C	92	07	00	08	01	00	00	0C	02	00	00	A0	
000001A0	07	00	04	00	00	00	30	31	30	30	01	A0	03	00	01	00
000001B0	00	00	01	00	A1	96	02	A0	04	00	01	00	00	00	80	03
000001C0	00	00	03	A0	04	00	01	00	00	00	58	02	00	00	00	00
000001D0	00	00	64	00	64	00	64	00	64	00	32	30	30	31	3A	30
000001E0	38	3A	30	38	20	31	35	3A	32	33	3A	35	34	00	32	30

Recover1.jpg
C:\Work\Chapter08\C08InChp\Expo

In-place mode!

File size: 177 KB
181,115 bytes

Undo level: 0
Undo reverses: n/a

Creation time: 07/18/2017 00:08:03

Last write time: 07/18/2017 00:08:03

Attributes: A
Icons: 0

Mode: Text
Character set: ANSI ASCII
Offsets: hexadecimal
Bytes per page: 37x16=592

Window #: 1
No. of windows: 1

Clipboard: available
TEMP folder: 429 GB free C:\Work

Figure 8-13 Recover1.jpg open in WinHex

Source: X-Ways AG, www.xways.net



Rebuilding File Headers (4 of 6)

Inserting FF D8 FF E0 starting at offset 0

After changing z to an uppercase J

WinHex - [Recover1.jpg]

File Edit Search Position View Tools Specialist Options Window Help

Case Data Recover1.jpg

Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F

00000000 FF D8 FF E0 00 10 4A 6E 49 46 00 01 01 01 01 00 78
00000010 00 78 00 0F E1 03 1C 45 78 69 66 00 00 49 49
00000020 2A 00 08 00 00 00 0B 00 0E 01 02 00 0A 00 00 00 *
00000030 92 00 00 00 0F 01 02 00 12 00 00 00 9C 00 00 00
00000040 10 01 02 00 12 00 00 00 AE 00 00 00 12 01 03 00
00000050 01 00 00 00 01 00 FF FF 1A 01 05 00 01 00 00 00
00000060 C0 00 00 00 1B 01 05 00 01 00 00 00 C8 00 00 00
00000070 28 01 03 00 01 00 00 00 02 00 FF FF 31 01 02 00
00000080 0A 00 00 00 D0 00 00 00 32 01 02 00 14 00 00 00
00000090 D4 00 00 00 13 02 03 00 01 00 00 00 02 00 FF FF
000000A0 69 87 04 00 01 00 00 00 EE 00 00 00 00 00 00 00
000000B0 20 20 20 20 20 20 20 20 00 4D 69 6E 6F 6C 74
000000C0 61 20 43 6F 2E 2C 20 4C 74 64 20 00 44 69 6D 61
000000D0 67 65 20 32 33 33 30 20 5A 6F 6F 6D 20 00 48 00
000000E0 00 00 01 00 00 00 48 00 00 00 01 00 00 00 20 20
000000F0 20 20 20 20 20 20 00 32 30 30 31 3A 30 38 3A
00000100 30 38 20 31 35 3A 32 33 3A 35 34 00 10 00 27 88
00000110 03 00 04 00 00 00 B4 01 00 00 00 90 07 00 04 00
00000120 00 00 30 32 31 30 03 90 02 00 14 00 00 00 BC 01
00000130 00 00 04 90 02 00 14 00 00 00 D0 01 00 00 01 91
00000140 07 00 04 00 00 00 01 02 03 00 02 91 05 00 01 00
00000150 00 00 E4 01 00 00 01 92 0A 00 01 00 00 EC 01
00000160 00 00 02 92 05 00 01 00 00 00 F4 01 00 00 04 92
00000170 0A 00 01 00 00 00 FC 01 00 00 09 92 03 00 01 00
00000180 00 00 00 00 FF FF 0A 92 05 00 01 00 00 00 04 02
00000190 00 00 7C 92 07 00 08 01 00 00 0C 02 00 00 00 A0
000001A0 07 00 04 00 00 00 30 31 30 30 01 A0 03 00 01 00
000001B0 00 00 01 00 A1 96 02 A0 04 00 01 00 00 00 80 03
000001C0 00 00 03 A0 04 00 01 00 00 00 58 02 00 00 00 00 00
000001D0 00 00 C4 00 C4 00 C4 00 C4 00 C4 00 22 20 20 21 21 20

Recovered1.jpg
C:\Work\Chapter08\C08InChp\Expo

File size: 177 KB
181,115 bytes

In-place mode!

Undo level: 0
Undo reverses: keyboard input

Creation time: 07/18/2017 00:08:03

Last write time: 07/18/2017 00:08:03

Attributes: A
Icons: 0

Mode: Text
Character set: ANSI ASCII
Offsets: hexadecimal
Bytes per page: 3x16=592

Window #: 1
No. of windows: 1

Clipboard: available
TEMP folder: 429 GB free
C:\Work

Figure 8-14 Inserting correct hexadecimal values for a JPEG file

Source: X-Ways AG, www.xways.net



Rebuilding File Headers (5 of 6)

ASCII hexadecimal conversion table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
2	SP	!	"	#	\$	%	&	'	{)	*	+	,	-
3	0	1	2	3	4	5	6	7	8	9	:	:	<	=
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M
5	P	Q	R	S	T	U	V	W	X	Y	Z	I	\]
6	.	a	b	c	d	e	f	g	h	i	j	k	l	m
7	p	q	r	s	t	u	v	w	x	y	z	{	}	

Uppercase "A" = 41
Lowercase "a" = 61

Figure 8-15 ASCII equivalents of hexadecimal values



Rebuilding File Headers (6 of 6)



Figure 8-16 Fixed1.jpg open in an image viewer



Reconstructing File Fragments

- Locate the noncontiguous clusters that make up a deleted file
- Steps
 - Locate and export all clusters of the fragmented file
 - Determine the starting and ending cluster numbers for each fragmented group of sectors
 - Copy each fragmented group of sectors in their correct sequence to a recovery file
 - Rebuild the file's header to make it readable in a graphics viewer
 - Add a .txt extension on all the copied sectors



Identifying Unknown File Formats

- Knowing the purpose of each format and how it stores data is part of the investigation process
- The Internet is the best source
 - Search engines
 - Find explanations and viewers
- Popular Web sites
 - [FileFormat.info](#)
 - [Extension Informer](#)
 - [The Graphics File Formats Page](#)



Analyzing Graphics File Headers (1 of 3)

- Necessary when you find files your tools do not recognize
- Use a hexadecimal editor such as WinHex
 - Record hexadecimal values in the header and use them to define a file type
- Example:
 - XIF file format is old, little information is available
 - The first 3 bytes of an XIF file are the same as a TIF file
 - Build your own header search string



Analyzing Graphics File Headers (2 of 3)

TIF file headers start with hexadecimal 49 49 2A, equivalent to ASCII II

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	49	49	2A	00	6E	EE	05	00	80	0B	4B	2A	07	F8	06	0C	II* ni K* ø B Á ú ÅbP . * Æ iø ÅcÑ NMS Eäò \ 9+IL &S9¤ K5 Eg Y i} 'Q'ä p Bp iQ ÿ eÀÀ V? ç@ ` H(ý° oÈ-j'X B QÀ 8 }Ó@6, _ Öa à Ú µC)ñÈ»ùÑ K QA L ùþúÅ æÃ H\$QÑ)& Ña
00000010	00	84	42	20	C0	10	03	FA	18	00	85	C2	62	50	88	2C	
00000020	2A	0F	12	88	C6	20	EF	F8	98	08	01	05	01	C2	63	D1	
00000030	A8	7C	4E	4D	24	88	45	E4	F2	89	5C	1A	39	2B	96	4C	
00000040	26	53	39	A4	A6	4B	35	9A	C6	67	13	59	1C	EE	7D	16	
00000050	9B	CC	27	51	27	E4	7C	06	FE	8F	BF	DF	70	87	EC	51	
00000060	FF	1C	01	80	C0	C0	00	15	56	3F	0E	86	BF	A9	00	60	
00000070	48	28	00	FD	B0	00	1F	6F	CA	2D	6A	91	58	9B	42	00	
00000080	51	C0	10	06	38	04	7D	D3	40	36	B8	80	00	09	5F	B9	
00000090	D2	61	8F	E0	14	86	DA	05	B5	43	29	F1	C8	BB	F9	F5	
000000A0	4B	84	51	41	00	4C	03	F9	FE	FA	C4	D2	00	A0	7C	03	
000000B0	E6	C3	05	8F	48	24	51	F0	04	86	29	26	00	D1	61	00	

Figure 8-17 A TIF file open in WinHex

Source: X-Ways AG, www.x-ways.net



Analyzing Graphics File Headers (3 of 3)

XIF file header ASCII equivalent shows the same beginning values as a TIF extension

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
000000000	49	49	2A	00	5C	01	00	00	20	65	58	74	65	6E	64	65	II*	\ eXtende
000000010	64	20	03	00	05	00	01	00	34	00	00	00	02	00	40	00	d	4 @
000000020	00	00	03	00	00	00	00	00	05	00	00	00	00	00	04	00		
000000030	00	00	00	00	01	00	20	00	01	00	B4	00	00	00	00	00		
000000040	6F	00	41	75	74	68	6F	72	00	58	65	72	6F	78	00	43	o Author Xerox C	
000000050	6F	72	70	00	00	44	61	74	65	00	4A	75	6C	00	32	31	orp Date Jul 21	
000000060	20	31	39	39	39	00	43	6F	70	79	72	69	67	68	74	00	1999 Copyright	
000000070	43	6F	70	79	72	69	67	68	74	00	28	43	29	00	31	39	Copyright (C) 19	
000000080	39	35	2D	31	39	39	36	00	58	65	72	6F	78	00	43	6F	95-1996 Xerox Co	
000000090	72	70	6F	72	61	74	69	6F	6E	2C	20	41	6C	6C	20	52	rporation, All R	
0000000A0	69	67	68	74	73	20	52	65	73	65	72	76	65	64	00	00	ights Reserved	
0000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		

Figure 8-18 An XIF file open in WinHex

Source: X-Ways AG, www.x-ways.net



Tools for Viewing Images

- After recovering a graphics file
 - Use an image viewer to open and view it
- No one viewer program can read every file format
 - Having many different viewer programs is best
- Most GUI forensics tools include image viewers that display common image formats
- Be sure to analyze, identify, and inspect every unknown file on a drive



Understanding Steganography in Graphics Files (1 of 7)

- Steganography hides information inside image files
 - An ancient technique
- Two major forms: insertion and substitution
 - Hidden data is not displayed when viewing host file in its associated program
 - You need to analyze the data structure carefully
 - Example: Web page
-



Understanding Steganography in Graphics Files (2 of 7)

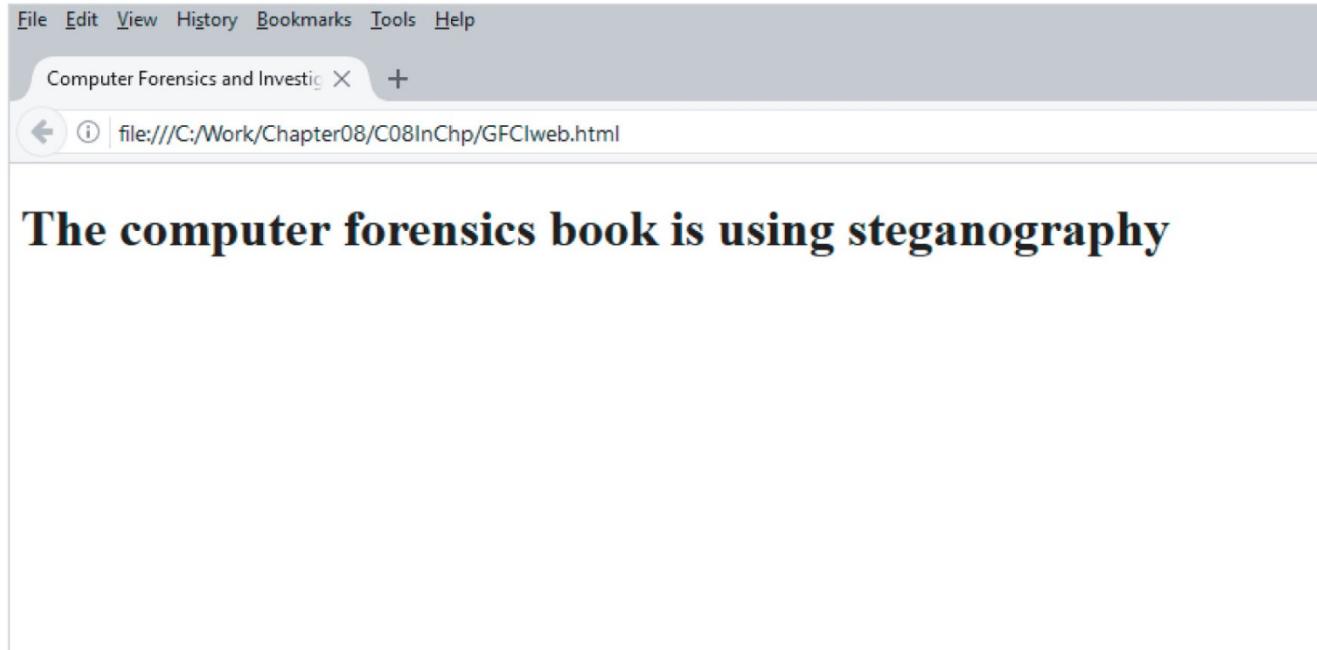
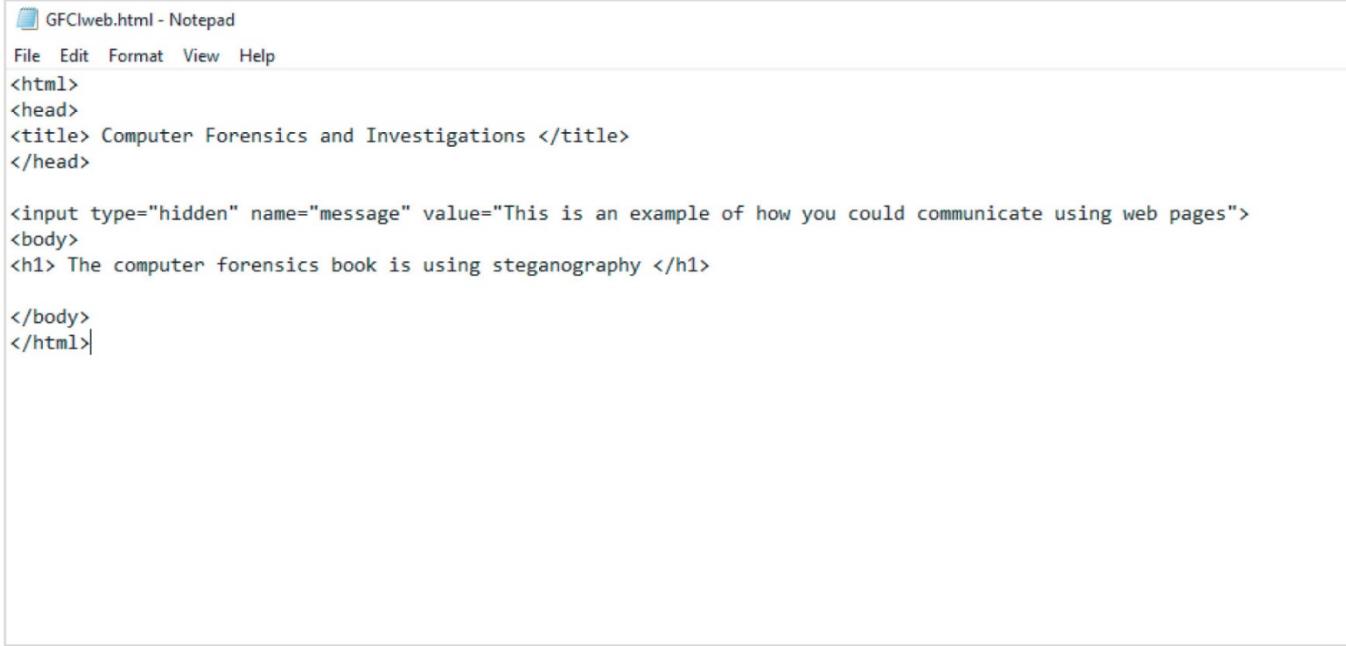


Figure 8-19 A simple Web page displayed in a Web browser

Source: The Mozilla Foundation, www.mozilla.org



Understanding Steganography in Graphics Files (3 of 7)



```
GFCIweb.html - Notepad
File Edit Format View Help
<html>
<head>
<title> Computer Forensics and Investigations </title>
</head>

<input type="hidden" name="message" value="This is an example of how you could communicate using web pages">
<body>
<h1> The computer forensics book is using steganography </h1>

</body>
</html>
```

Figure 8-20 The HTML code reveals hidden text

Source: The Mozilla Foundation, www.mozilla.org



Understanding Steganography in Graphics Files (4 of 7)

- Substitution
 - Replaces bits of the host file with other bits of data
 - Usually change the last two LSBs (**least significant bit**)
 - Detected with steganalysis tools (a.k.a - steg tools)
- You should inspect all files for evidence of steganography
- Clues to look for:
 - Duplicate files with different hash values
 - Steganography programs installed on suspect's drive



Understanding Steganography in Graphics Files (5 of 7)

Table 8-1	Bit breakdown of a secret message
Original Pixel	Altered Pixel
1010 1010	1010 1001
1001 1101	1001 1110
1111 0000	1111 0011
0011 1111	0011 1100



Understanding Steganography in Graphics Files (6 of 7)



Figure 8-21 Original and altered images



Understanding Steganography in Graphics Files (7 of 7)

My secret bank accounts:				
Country	Bank	Account No.	Passcode	Currency Amt.
Swiss	Swiss National SA	26845622	Y1115AQ	1.2 million CHF
Caymen Is.	Caribbean Intn. Bank Ltd.	5589999	SAMMM242	5.82 million KYD
Malta	Valletta Nat. Bank Limited	57896165	558TF558	2.3 million EUR
Hong Kong	Chan Wag Bank	A5AA59	665308888	8.9 million HKD
South Africa	Rand Bank of Cape Town	6982543	AAF8	0.53 million ZAL

Figure 8-22 A hidden message in the altered image



Using Steganalysis Tools

- Use steg tools to detect, decode, and record hidden data
- Detect variations of the graphic image
 - When done correctly you cannot detect hidden data in most cases
- Check to see whether the file size, image quality, or file extensions have changed



Understanding Copyright Issues with Graphics

- Steganography has been used to protect copyrighted material
 - By inserting digital watermarks into a file
- Digital investigators need to aware of copyright laws
- Copyright laws for Internet are not clear
 - There is no international copyright law
- Check the [U.S. Copyright Office](#)
 - U.S. Copyright Office identifies what can and can't be covered under copyright law in U.S.
- Fair use
 - Another guideline to consider



Summary (1 of 3)

- Three types of graphics files
 - Bitmap
 - Vector
 - Metafile
- Image quality depends on various factors
- Standard file formats: .gif, .jpeg, .bmp, and .tif
- Nonstandard file formats: .tga, .rtl, .psd, and .svg
- Some image formats compress their data
 - Lossless compression
 - Lossy compression



Summary (2 of 3)

- Digital camera photos are typically in raw and EXIF JPEG formats
- Recovering image files
 - Carving file fragments
 - Rebuilding image headers
- The Internet is best for learning more about file formats and their extensions
- Software
 - Image editors
 - Image viewers



Summary (3 of 3)

- Steganography
 - Hides information inside image files
 - Forms
 - Insertion
 - Substitution
- Steganalysis
 - Finds whether image files hide information
 - Fair use allows using copyrighted material for noncommercial or educational purposes without having to compensate the material's originator or owner