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

Graphic File Types

Types of Graphic Files

- 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

Bitmap vs Raster

- Bitmap images
  - Grids of individual **pixels**
- **Raster images** - also collections of pixels
  - Pixels are stored in rows
  - Better for printing
- Neither of these types can scale without reducing quality as you're effectively just increasing the size of the pixels

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
- Several software products to design, almost any web browser can view

Metafile

- 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

Example Formats

- Portable Network Graphic (png)
- Graphic Interchange Format (gif)
- Joint Photographic Experts Group (jpeg, jpg)
- Tagged Image File Format (tiff, tif)
- Bitmap (bmp)
- Raw - https://en.wikipedia.org/wiki/Raw_image_format
- Targa (tga)
- Raster Transfer Language (rtl)
- Photoshop (psd)
- Illustrator (ai)
- Scalable Vector Graphics (svg)

Exchangeable Image File Format (EXIF)

- Commonly used to store digital pictures
- Developed by Japan Electronics & Information Technology Industries Association (JEITA) as a standard for storing metadata in JPEG and TIF files
- Could contain
  - Date & time
  - Camera information
  - Location
  - Original resolution info

Recovering Graphics Files

Compression in Images

- Many types do compress their contents
  - GIF, JPEG
- Others do not
  - BMP
- 2 types of compression
  - Lossless & lossy

Lossless vs Lossy

- Lossless
  - Reduces file size without removing data
  - Example: PNG, GIF
- Lossy
  - Permanently discards information
  - Example: JPEG

Finding 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
- Most commonly done by
  - Reviewing the file system for space marked unallocated
  - Examination of file signatures and headers

File Signatures / Magic Numbers

- Originally 2-byte IDs at the beginning of files, but now realistically any number of first bytes - https://en.wikipedia.org/wiki/File_format#Magic_number

- Examples:
  - https://en.wikipedia.org/wiki/List_of_file_signatures
  - https://www.garykessler.net/library/file_sigs.html

Repairing Damaged Headers

- 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
  - Hex editors that support templates are particularly helpful for this purpose

Reconstructing 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

Exploring the JPEG Header

Exploring EXIF

Recovery Example

Pages 352-359

Part 1

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

Part 2

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

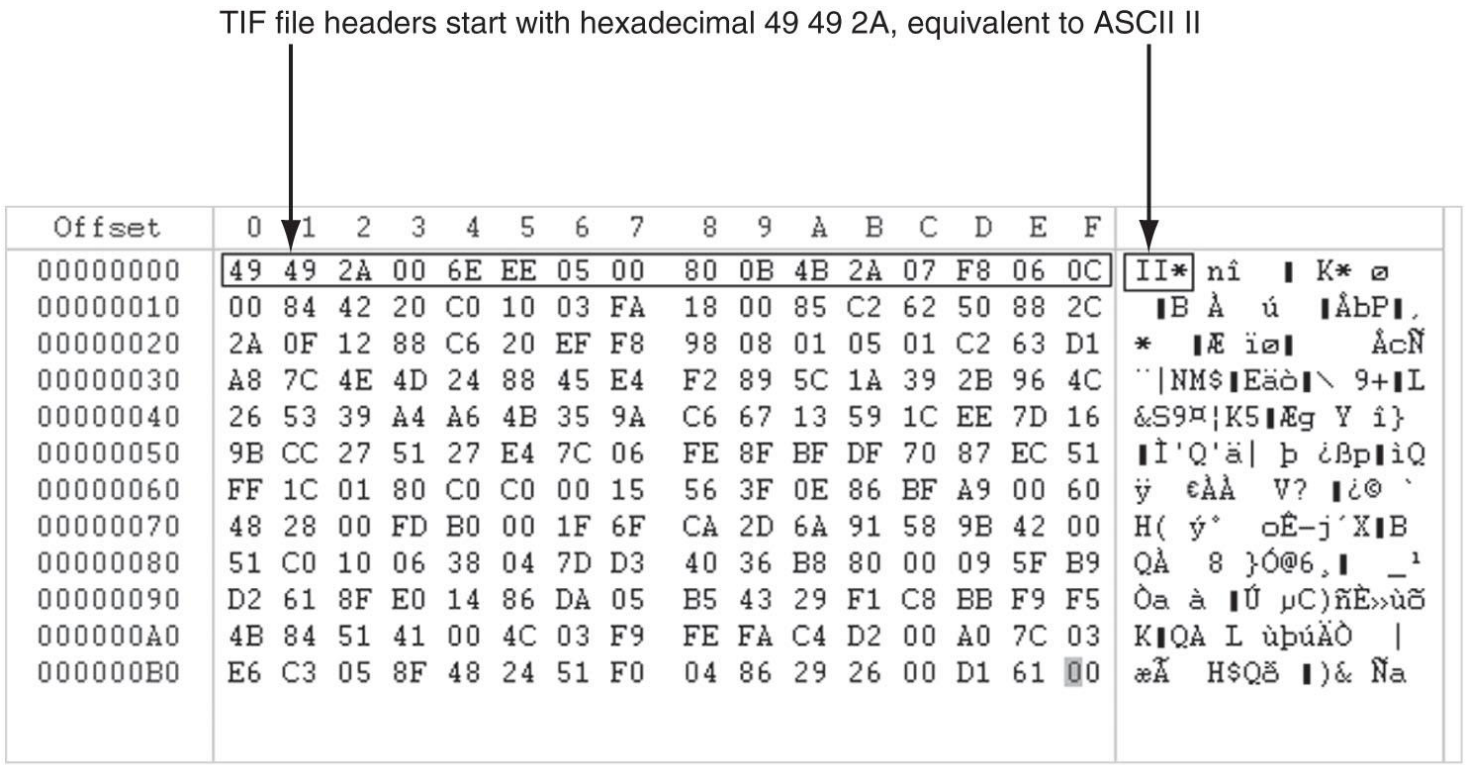
Figure 8-6 The e-mail with attachments IT found

Identifying Unknown Formats

Analysis of Headers

- Necessary when you find files your tools do not recognize
- Use a hexadecimal editor
  - 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

TIF vs XIF



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
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

Offset	ASCII
00000000	II* nî ▌ K* ø
00000010	▌B À ú ▌ÅbP▌,
00000020	* ▌Æ ïø▌ ÅcÑ
00000030	¨|NM$▌Eäò▌\ 9+▌L
00000040	&S9¤|K5▌Æg Y î}
00000050	▌Ì'Q'ä| þ ¿ßp▌ìQ
00000060	ÿ €ÀÀ V? ▌¿© `
00000070	H(ý° oÊ-j´X▌B
00000080	QÀ 8 }Ó@6¸▌ _¹
00000090	Òa à ▌Ú µC)ñÈ»ùõ
000000A0	K▌QA L ùþúÄÒ |
000000B0	æÃ H$Qð ▌)& Ña

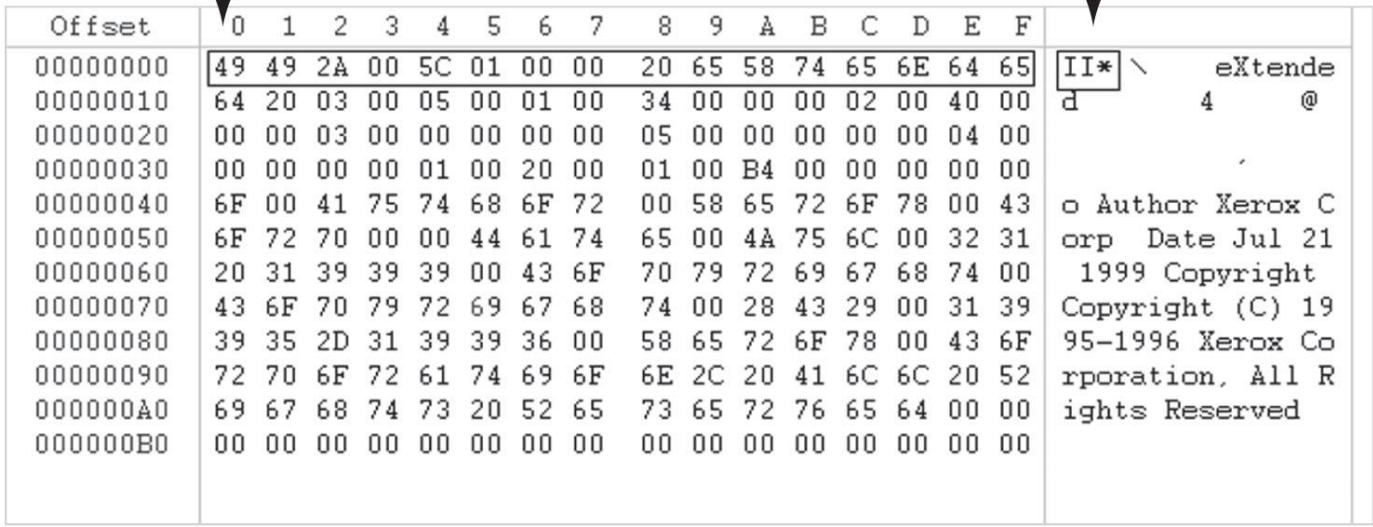
Figure 8-17 A TIF file open in WinHex

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

TIF vs XIF

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	
00000000	49	49	2A	00	5C	01	00	00	20	65	58	74	65	6E	64	65	II* \ eXtende
00000010	64	20	03	00	05	00	01	00	34	00	00	00	02	00	40	00	d 4 @
00000020	00	00	03	00	00	00	00	00	05	00	00	00	00	00	04	00	
00000030	00	00	00	00	01	00	20	00	01	00	B4	00	00	00	00	00	
00000040	6F	00	41	75	74	68	6F	72	00	58	65	72	6F	78	00	43	o Author Xerox C
00000050	6F	72	70	00	00	44	61	74	65	00	4A	75	6C	00	32	31	orp Date Jul 21
00000060	20	31	39	39	39	00	43	6F	70	79	72	69	67	68	74	00	1999 Copyright
00000070	43	6F	70	79	72	69	67	68	74	00	28	43	29	00	31	39	Copyright (C) 19
00000080	39	35	2D	31	39	39	36	00	58	65	72	6F	78	00	43	6F	95-1996 Xerox Co
00000090	72	70	6F	72	61	74	69	6F	6E	2C	20	41	6C	6C	20	52	rporation, All R
000000A0	69	67	68	74	73	20	52	65	73	65	72	76	65	64	00	00	ights Reserved
000000B0	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*

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
  - We talked about the importance of having viewers for all files you should encounter previously; you may need to adapt over time and by case
- Most GUI forensics tools include image viewers that display common image formats

Steganography

What is Steganography?

- Steganography hides information inside image files
  - An ancient technique, still used today to watermark files
- Insertion
  - Hidden data is not displayed when viewing host file in its associated program
  - You need to analyze the data structure carefully
- 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)

Clues

- Suspect drive has steganography tools installed
- Duplicate files with different hashes

Insertion Example

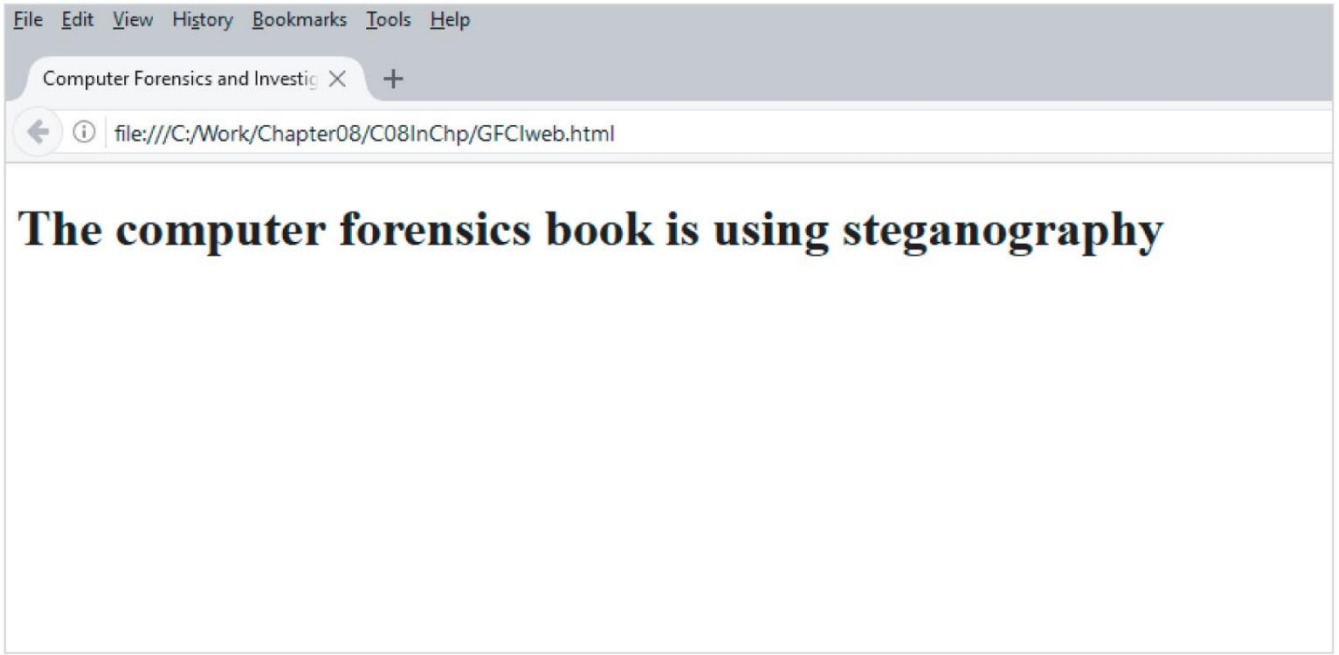


Figure 8-19 A simple Web page displayed in a Web browser

Source: The Mozilla Foundation, *www.mozilla.org*

Insertion Example (Cont.)

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*

Insertion w/ Image

- giraffe.png

Insertion w/ Space

- hi.c
- hii.c

Substitution w/ LSB in a BMP

- Pixels are stored in many images as 24 bits of color
  - 111111111111111111111111 = white
  - 000000000000000000000000 = black
- The decimal 255 in binary is 11111111
- We can shift the least significant bits of each color to store data within them

Significance	128	64	32	16	8	4	2	1
Value	1	0	1	1	0	1	0	0

Substitution w/ LSB in a BMP

- The letter “a” = 97 = 01100001
- If the red value for the first pixel is 10011100 and we change the low order bit to reflect our lowest bit for ‘a’ it becomes 157 or 10011101

^156 ^157

Substitution w/ LSB in a BMP

- cat.bmp

Lab Intro

What are these? Where was that? Who took the picture?

References

- *Guide to Computer Forensics and Investigations*
  - *ISBN: 9780357688595*