CAL POLY

California Cybersecurity Institute

Computer Forensics CCIC Training

Chapter 7: Recycle Bin

Lauren Pixley, Cassidy Elwell, and James Poirier

May 2019 (Version 1)

7

Recycle Bin

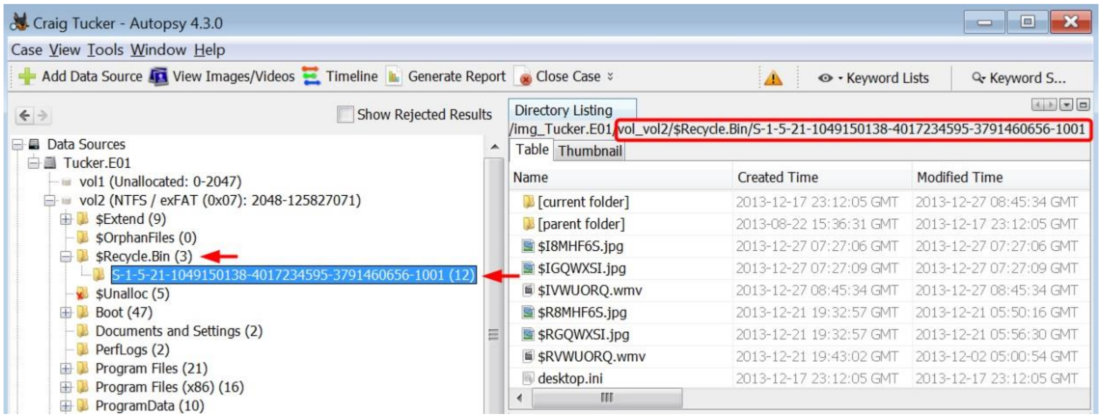
Introduction

Since the file “Underage_lolita_r@ygold_001.jpg is no longer in the Pictures folder and you don’t have Craig’s E: drive, you should see if it was deleted. Most computer users believe that when a file is deleted and the recycling bin is emptied, that the file cannot be accessed. However, deleting a file can still leave data behind for recovery. This is because when a file is deleted, the data is only marked as deleted by the computer and allows that area of the disk to be available for storing new data. It's not until the user overwrites this area of the disk that the data is actually deleted. Even part of the original file may still be recoverable if the user only overwrote a portion of the disk space. Therefore, this data can be recovered by investigators. This process is known as "file carving."

$R and $I Files

The first time a user deletes a file, the file is not actually deleted. A new folder is created in the Recycle Bin, and the deleted files are moved to it. The folder that is created is named after the security identifier (SID) and the relative ID (RID) of the associated user. You can use this information to determine which user account deleted the file. You are going to look at Craig’s Recycle Bin (see Figure 7-1), which is located in:

```
C:\$Recycle.Bin\S-1-5-21-1049150138-4017234595-3791460656-1001
```



Data Sources
Tucker.E01
vol1 (Unallocated: 0-2047)
vol2 (NTFS / exFAT (0x07): 2048-125827071)
$Extend (9)
$OrphanFiles (0)
$Recycle.Bin (3)
S-1-5-21-1049150138-4017234595-3791460656-1001 (12)
$Unalloc (5)
Boot (47)
Documents and Settings (2)
PerfLogs (2)
Program Files (21)
Program Files (x86) (16)
ProgramData (10)

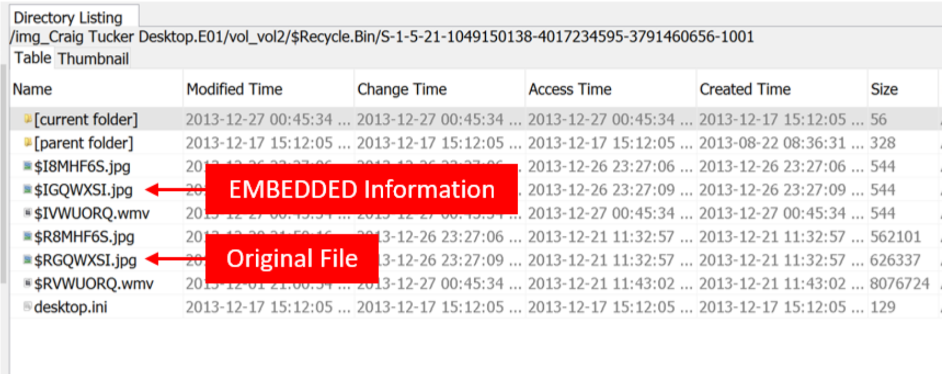
/img_Tucker.E01/vol_vol2/$Recycle.Bin/S-1-5-21-1049150138-4017234595-3791460656-1001

Name	Created Time	Modified Time
[current folder]	2013-12-17 23:12:05 GMT	2013-12-27 08:45:34 GMT
[parent folder]	2013-08-22 15:36:31 GMT	2013-12-17 23:12:05 GMT
$I8MHF6S.jpg	2013-12-27 07:27:06 GMT	2013-12-27 07:27:06 GMT
$IGQWXSI.jpg	2013-12-27 07:27:09 GMT	2013-12-27 07:27:09 GMT
$IVWUORQ.wmv	2013-12-27 08:45:34 GMT	2013-12-27 08:45:34 GMT
$R8MHF6S.jpg	2013-12-21 19:32:57 GMT	2013-12-21 05:50:16 GMT
$RGQWXSI.jpg	2013-12-21 19:32:57 GMT	2013-12-21 05:56:30 GMT
$RVWUORQ.wmv	2013-12-21 19:43:02 GMT	2013-12-02 05:00:54 GMT
desktop.ini	2013-12-17 23:12:05 GMT	2013-12-17 23:12:05 GMT

Figure 7-1 - Craig's Recycle Bin

Note: If a user deletes a file using the command prompt or does Shift+Del, the file bypasses the Recycle Bin and it is deleted right away.

Both file names contain 6 identical, random characters and its original extension, preceded by either $I or $R. The **$I** file contains information about the deleted file including the file's size, deleted time, path, etc. The **$R** file contains the actual deleted file itself.

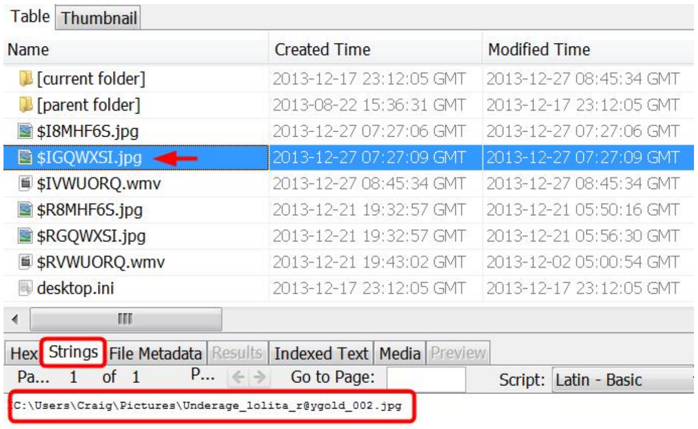


/img_Craig Tucker Desktop.E01/vol_vol2/$Recycle.Bin/S-1-5-21-1049150138-4017234595-3791460656-1001

Name	Modified Time	Change Time	Access Time	Created Time	Size
[current folder]	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	2013-12-17 15:12:05 ...	56
[parent folder]	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-08-22 08:36:31 ...	328
$I8MHF6S.jpg	[illegible]	[illegible]	2013-12-26 23:27:06 ...	2013-12-26 23:27:06 ...	544
$IGQWXSI.jpg	[illegible]	[illegible]	2013-12-26 23:27:09 ...	2013-12-26 23:27:09 ...	544
$IVWUORQ.wmv	[illegible]	[illegible]	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	544
$R8MHF6S.jpg	[illegible]	2013-12-26 23:27:06 ...	2013-12-21 11:32:57 ...	2013-12-21 11:32:57 ...	562101
$RGQWXSI.jpg	[illegible]	3-12-26 23:27:09 ...	2013-12-21 11:32:57 ...	2013-12-21 11:32:57 ...	626337
$RVWUORQ.wmv	[illegible]	2013-12-27 00:45:34 ...	2013-12-21 11:43:02 ...	2013-12-21 11:43:02 ...	8076724
desktop.ini	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	129

Figure 7-2 - $I and $R Files

In Autopsy, look at the $I file called $IGQWXSI.jpg and click the Strings view. You will see the path of where $IGQWXSI.jpg was originally stored on the computer before the user deleted the file.

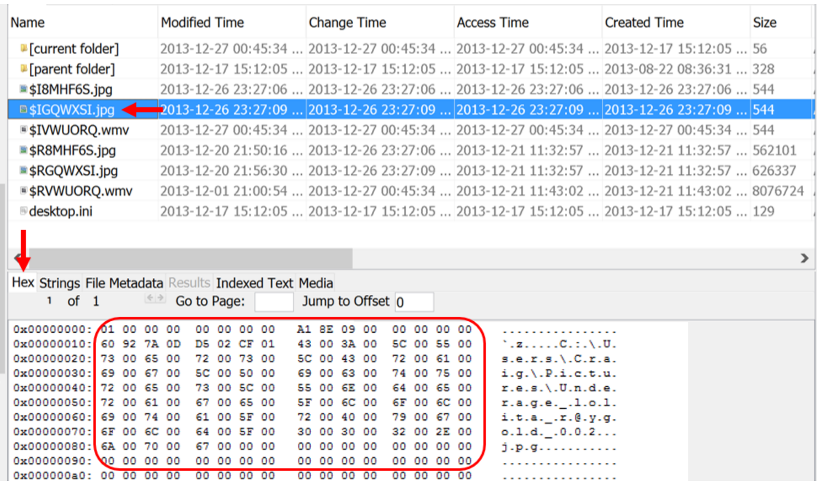


Name	Created Time	Modified Time
[current folder]	2013-12-17 23:12:05 GMT	2013-12-27 08:45:34 GMT
[parent folder]	2013-08-22 15:36:31 GMT	2013-12-17 23:12:05 GMT
$I8MHF6S.jpg	2013-12-27 07:27:06 GMT	2013-12-27 07:27:06 GMT
$IGQWXSI.jpg	2013-12-27 07:27:09 GMT	2013-12-27 07:27:09 GMT
$IVWUORQ.wmv	2013-12-27 08:45:34 GMT	2013-12-27 08:45:34 GMT
$R8MHF6S.jpg	2013-12-21 19:32:57 GMT	2013-12-21 05:50:16 GMT
$RGQWXSI.jpg	2013-12-21 19:32:57 GMT	2013-12-21 05:56:30 GMT
$RVWUORQ.wmv	2013-12-21 19:43:02 GMT	2013-12-02 05:00:54 GMT
desktop.ini	2013-12-17 23:12:05 GMT	2013-12-17 23:12:05 GMT

```
C:\Users\Craig\Pictures\Underage_lolita_r@ygold_002.jpg
```

Figure 7-3 - $IGQWXSI File Shows Location and Name of File Before Deletion

Next, if you view $IGQWXSI.jpg in Hex view, you can see there is EMBEDDED data regarding the file before it was deleted. There are three pieces of critical information: the Windows version, file size, and time and date.



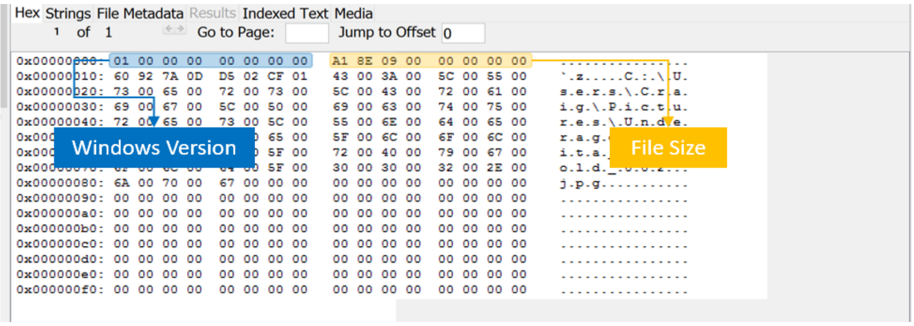
Name	Modified Time	Change Time	Access Time	Created Time	Size
[current folder]	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	2013-12-17 15:12:05 ...	56
[parent folder]	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-08-22 08:36:31 ...	328
$I8MHF6S.jpg	2013-12-26 23:27:06 ...	2013-12-26 23:27:06 ...	2013-12-26 23:27:06 ...	2013-12-26 23:27:06 ...	544
$IGQWXSI.jpg	2013-12-26 23:27:09 ...	2013-12-26 23:27:09 ...	2013-12-26 23:27:09 ...	2013-12-26 23:27:09 ...	544
$IVWUORQ.wmv	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	2013-12-27 00:45:34 ...	544
$R8MHF6S.jpg	2013-12-20 21:50:16 ...	2013-12-26 23:27:06 ...	2013-12-21 11:32:57 ...	2013-12-21 11:32:57 ...	562101
$RGQWXSI.jpg	2013-12-20 21:56:30 ...	2013-12-26 23:27:09 ...	2013-12-21 11:32:57 ...	2013-12-21 11:32:57 ...	626337
$RVWUORQ.wmv	2013-12-01 21:00:54 ...	2013-12-27 00:45:34 ...	2013-12-21 11:43:02 ...	2013-12-21 11:43:02 ...	8076724
desktop.ini	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	2013-12-17 15:12:05 ...	129

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

```
0x00000000: 01 00 00 00  00 00 00 00  A1 8E 09 00  00 00 00 00   ................
0x00000010: 60 92 7A 0D  D5 02 CF 01  43 00 3A 00  5C 00 55 00   `.z.....C.:.\.U.
0x00000020: 73 00 65 00  72 00 73 00  5C 00 43 00  72 00 61 00   s.e.r.s.\.C.r.a.
0x00000030: 69 00 67 00  5C 00 50 00  69 00 63 00  74 00 75 00   i.g.\.P.i.c.t.u.
0x00000040: 72 00 65 00  73 00 5C 00  55 00 6E 00  64 00 65 00   r.e.s.\.U.n.d.e.
0x00000050: 72 00 61 00  67 00 65 00  5F 00 6C 00  6F 00 6C 00   r.a.g.e._.l.o.l.
0x00000060: 69 00 74 00  61 00 5F 00  72 00 40 00  79 00 67 00   i.t.a._.r.@.y.g.
0x00000070: 6F 00 6C 00  64 00 5F 00  30 00 30 00  32 00 2E 00   o.l.d._.0.0.2...
0x00000080: 6A 00 70 00  67 00 00 00  00 00 00 00  00 00 00 00   j.p.g...........
0x00000090: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000a0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Figure 7-4 – Embedded Data in $IGQWXSI File

The first 8 bytes of the $I file tell you the Windows version the file was created on. If the first byte is "01" then the user's computer was running Windows 8 and "02" if the user was running Windows 10. The second set of 8 bytes (starting at offset 8) represent the size of the original file.

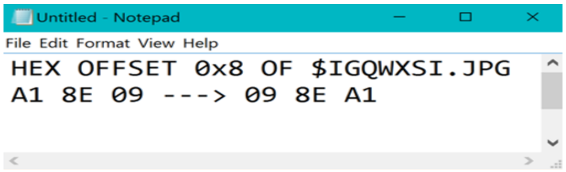


Hex Strings File Metadata Results Indexed Text Media
1 of 1 Go to Page: Jump to Offset 0
Windows Version
File Size

```
0x00000000: 01 00 00 00  00 00 00 00  A1 8E 09 00  00 00 00 00   ................
0x00000010: 60 92 7A 0D  D5 02 CF 01  43 00 3A 00  5C 00 55 00   `.z.....C.:.\.U.
0x00000020: 73 00 65 00  72 00 73 00  5C 00 43 00  72 00 61 00   s.e.r.s.\.C.r.a.
0x00000030: 69 00 67 00  5C 00 50 00  69 00 63 00  74 00 75 00   i.g.\.P.i.c.t.u.
0x00000040: 72 00 65 00  73 00 5C 00  55 00 6E 00  64 00 65 00   r.e.s.\.U.n.d.e.
0x000                         65 00  5F 00 6C 00  6F 00 6C 00   r.a.g.
0x000                         5F 00  72 00 40 00  79 00 67 00   i.t.a.
0x000                         5F 00  30 00 30 00  32 00 2E 00   o.l.d._
0x00000080: 6A 00 70 00  67 00 00 00  00 00 00 00  00 00 00 00   j.p.g...........
0x00000090: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000a0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000b0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000c0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000d0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000e0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
0x000000f0: 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Figure 7-5 – Embedded Information on Windows Version and Size of File

The hex value of the file size is represented in "Little Endian" format which means the little end is read first. To convert this into human-readable information you must read the hex value with the larger order first (read the sets of two digits from right to left). Therefore, you would convert from "Little Endian" as follows:



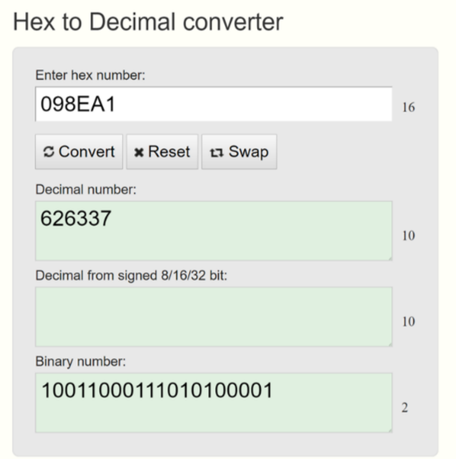
```
HEX OFFSET 0x8 OF $IGQWXSI.JPG
A1 8E 09 ---> 09 8E A1
```

Figure 7-6 – Convert Little Endian Format by Reading Data Right to Left

Then convert this hex value to a decimal to get the file size in bytes by using a conversion calculator. To do so, you are going to use the following website:

```
http://www.rapidtables.com/convert/number/hex-to-decimal.htm
```

Once you have this calculator open, type your converted human-readable hex value into the “Enter hex number” box and then click “Convert”.



Hex to Decimal converter

Enter hex number:

098EA1 16

Decimal number:

626337 10

Decimal from signed 8/16/32 bit:

10

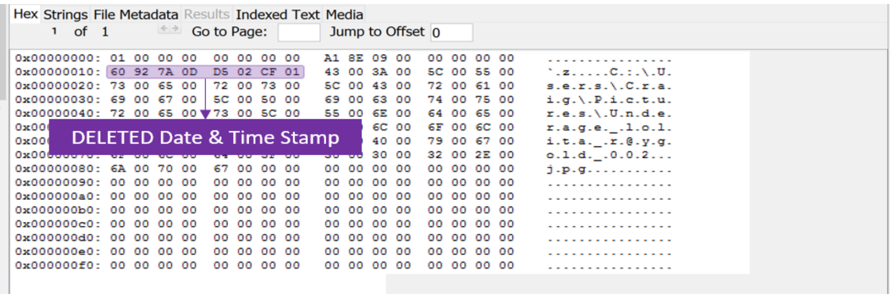
Binary number:

10011000111010100001 2

Figure 7-7 – Type in Converted File Size Hex Value from $$IGQWXSI.jpg and Click Convert

The value displayed in the “Decimal number” box in the conversion calculator is the size of the file before it was deleted by the user in bytes.

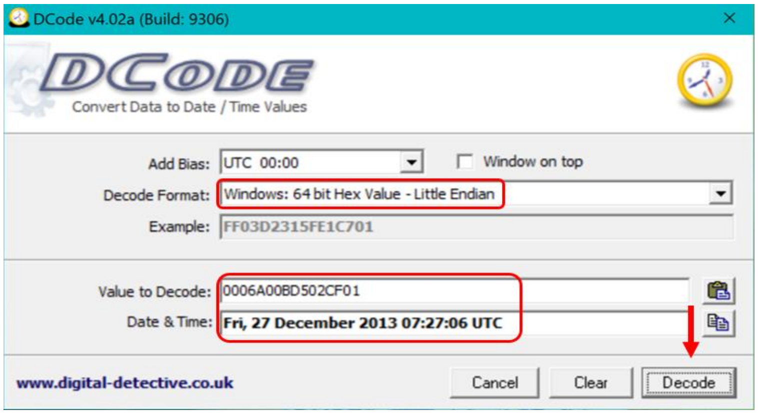
Next, encoded in the hex value of $IGQWXSI.jpg is the date and time stamp of when the file was deleted.



```
0x00000000: 01 00 00 00  00 00 00 00   A1 8E 09 00  00 00 00 00    ................
0x00000010: 60 92 7A 0D  D5 02 CF 01   43 00 3A 00  5C 00 55 00    `.z.....C.:.\.U.
0x00000020: 73 00 65 00  72 00 73 00   5C 00 43 00  72 00 61 00    s.e.r.s.\.C.r.a.
0x00000030: 69 00 67 00  5C 00 50 00   69 00 63 00  74 00 75 00    i.g.\.P.i.c.t.u.
0x00000040: 72 00 65 00  73 00 5C 00   55 00 6E 00  64 00 65 00    r.e.s.\.U.n.d.e.
0x00                                      6C 00  6F 00 6C 00    r.a.g.e._.l.o.l.
0x00                                      40 00  79 00 67 00    i.t.a._.r.@.y.g.
0x00                                      30 00  32 00 2E 00    o.l.d._.0.0.2...
0x00000080: 6A 00 70 00  67 00 00 00   00 00 00 00  00 00 00 00    j.p.g...........
0x00000090: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
0x000000a0: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
0x000000b0: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
0x000000c0: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
0x000000d0: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
0x000000e0: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
0x000000f0: 00 00 00 00  00 00 00 00   00 00 00 00  00 00 00 00    ................
```

Figure 7-8 – Embedded Deleted Time Stamp in $IGQWXSI.jpg

To decode the timestamp, you are going to once again use the DCode (Version 4.02a) tool. Once you have DCode open, you need to copy the time stamp from the Hex tab of the selected $I file which begins at offset 16 and is 8 bytes in length. Then, set the Decode Format to Windows: 64 bit Hex Value - Little Endian. Click the Decode button to see the decoded Date and Time. You should have Fri, 27 December 2013 07:27:06 UTC for when the file Underage_lolita_r@ygold_002.jpg was deleted from the user’s Pictures folder (see Figure 7-9).


DCode v4.02a (Build: 9306)
DCODE
Convert Data to Date / Time Values
Cancel
Clear
Decode

Add Bias: UTC 00:00

□ Window on top

Decode Format: Windows: 64 bit Hex Value - Little Endian

Example: FF03D2315FE1C701

Value to Decode: 0006A00BD502CF01

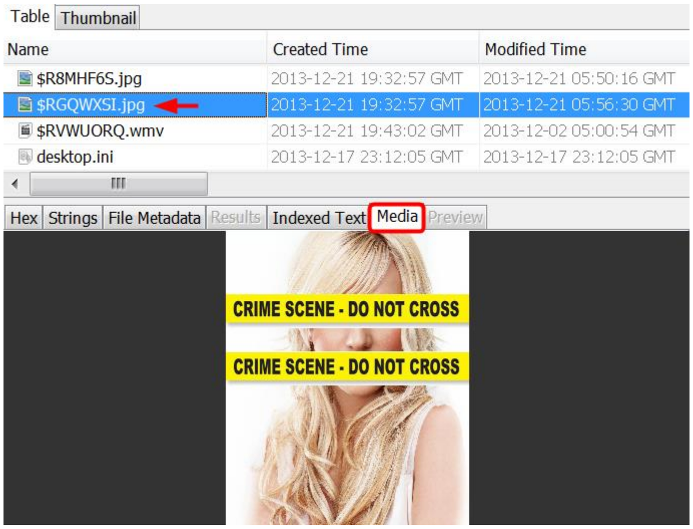
Date & Time: Fri, 27 December 2013 07:27:06 UTC

www.digital-detective.co.uk

Figure 7-9 – Copy Deleted Time Stamp, Set Format to Little Endian, and Click Decode

Note: This encoded date is important to decode since Autopsy does not automatically provide the Deleted time in its platform. This often helps with creating timelines in analysis.

Now, look at the file called $RGQWXSI.jpg and click on Media view. You will see a preview of the file that was deleted by the user. In this case, you should see a .jpg picture.



Name	Created Time	Modified Time
$R8MHF6S.jpg	2013-12-21 19:32:57 GMT	2013-12-21 05:50:16 GMT
$RGQWXSI.jpg	2013-12-21 19:32:57 GMT	2013-12-21 05:56:30 GMT
$RVWUORQ.wmv	2013-12-21 19:43:02 GMT	2013-12-02 05:00:54 GMT
desktop.ini	2013-12-17 23:12:05 GMT	2013-12-17 23:12:05 GMT

Figure 7-10 – Preview $RGQWXSI.jpg with Media View

Note: Throughout these practice and test images, you may see pictures with crime scene tape. These are the simulated child pornography images, and you should tag any of these images since they are important to your investigation.