**Aim: Windows Recycle Bin Forensics**

**Theory:**

In Windows, the Recycle Bin is a folder or directory where deleted items are temporarily stored. Deleted files are not permanently removed from the hard drive but are sent instead to the Recycle Bin, unless they are too large. The files in the Recycle Bin can be restored to their original location. They cannot be used directly while they are in the Recycle Bin.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Operating System | File system structure | Location of deleted files |
| 1 | Windows 95/98/ME | FAT32 | C:\Recycled\INFO2 |
| 2 | Windows NT/2K/XP | NTFS | C:\Recycler\SID\INFO2 |
| 3 | Windows Vista | NTFS | C:\$Recycle.Bin\SID\$IXXXXXX  C:\$Recycle.Bin\SID\$RXXXXXX |

The recycle bin is in the root of the OS drive which is typically *C:\* and it is called $Recycle.Bin in Windows 10.

Each deleted file will result in two files placed within the path. Firstly, $I file which will contain meta-data specific to that file which is the original file name and path of the file prior to deletion, the size of the file and the time at which it was deleted. Secondly, the actual file contents themselves will be stored within a $R file. Both files will be renamed to some random six-character value and then $I and $R will be prepended to each one of the respectively creating an 8-character file name.

**$I Meta-data File (Windows Vista and Later)**

*C:\$Recycle.Bin\SID\$Ixxxxxx*

* File name and full path of the deleted file
* Size of the deleted file
* Date/time at which the file was deleted.

**$R File (Windows Vista and Later)***C:\$Recycle.Bin\SID\$Rxxxxxx*

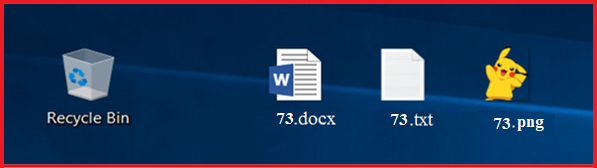
* $R File contains the contents of the deleted file.

*The*[*SID*](https://en.wikipedia.org/wiki/Security_Identifier)*sub-folder corresponds to the SID of the user that deleted the file. The sub-folder is created for a given user upon first deletion of a file that is sent to the Recycle Bin.*

So, $I is meta-data, $R is the actual recovery data representing the contents of the original file.

Let’s have an example:

There are docx, txt and png files on the desktop.



Let’s delete them and then look at the contents of the recycle bin

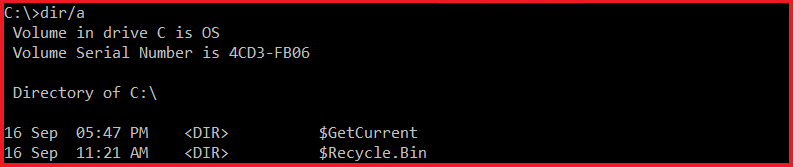
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Original Location | Date Deleted | Size | Item type | Date modified |
| 73.docx | C:\Users\HP\Desktop | 11 Mar 3:37 PM | 12 KB | Ms word | 11 Mar 3:34 PM |
| 73.txt | C:\Users\HP\Desktop | 11 Mar 3:37 PM | 1 KB | Text doc | 11 Mar 3:34 PM |
| 73.png | C:\Users\HP\Desktop | 11 Mar 3:37 PM | 43 KB | png file | 11 Mar 3:34 PM |

We can see that their original locations, deletion dates and times and sizes. This information is stored within the $I meta-data files.

Let’s take a look at this in the file system.

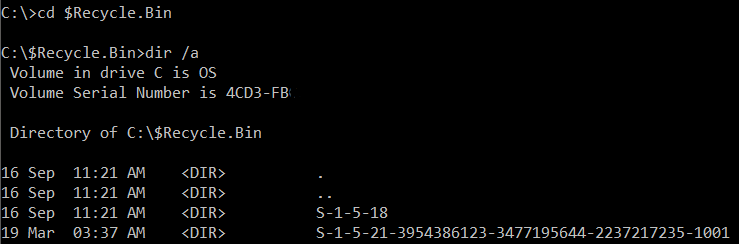
**1.**From command view hidden files by typing ***dir /a*** and we can see

$Recycle.Bin folder on the first line.



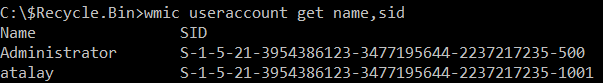
**2.**Enter into the $Recycle.Binfolder,

**3.**do ***dir /a*** again and then we see the SID folders.



We can see the users and their associated SIDS by entering

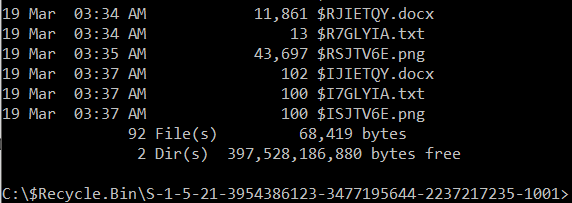
**4.*wmic useraccount get name,sid***



I’m currently logged in as atalay, so that is the folder which my recycle bin should be stored under.

Let’s go into it by

**5.*cd  S-1-5-21-3954386123-3477195644-2237217235-1001***

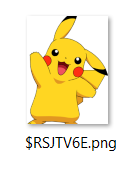


So, we see the files as expected. There are 3 $R files containing actual recovery data for those files and 3 $I files containing the meta-data for 3 files I just deleted.

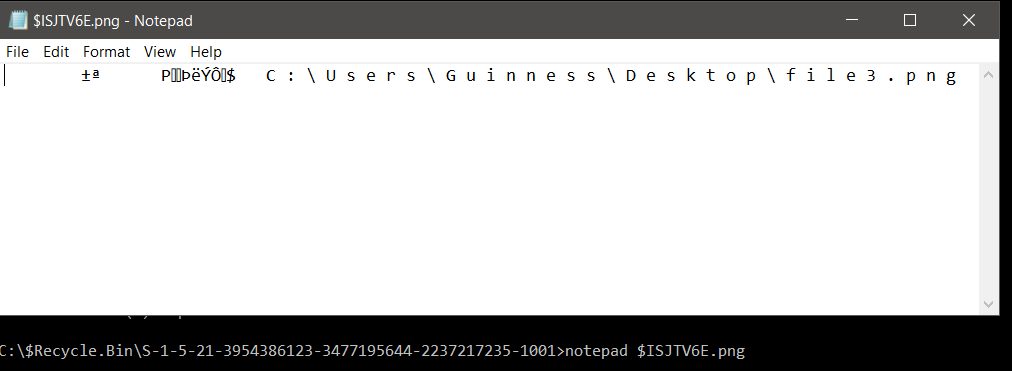
For example, if I take $RSJTV6E.png file and copy it on to RecFor folder we will see that it is indeed the original file that was deleted.



It has been renamed with a random 6 character in front of that the $R has been prepended.

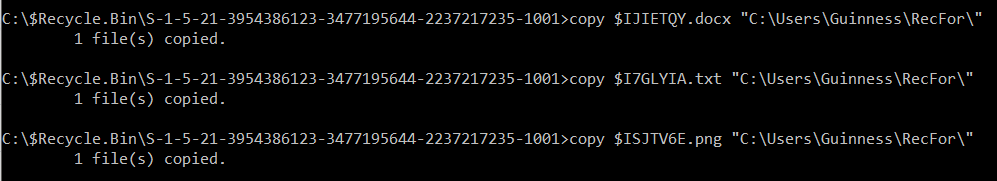


Now let’s look at one of the two $I files by opening it up with notepad

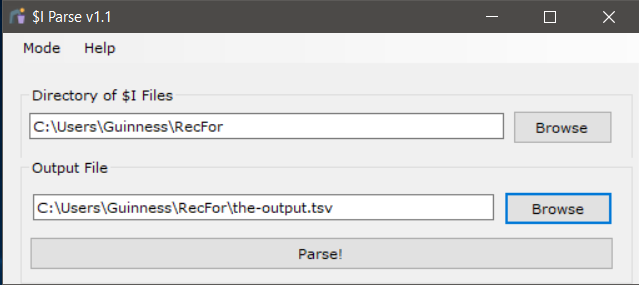


We can see that it’s not easy to parse, but we can see the original path and the file name that was deleted as shown above. However, there is a tool that makes this much easier to parse. The tool is called $I Parse written by Jason Hale. You can download the tool.

Let’s copy the $I meta-data files on to RecFor folder.



Open the $I Parse tool, specify the directory of $I files and call output file as the-output.



That is it. Click parse.

Finally, if we open the tsv file that has been generated, will see the deletion date, original file name as full path, and their sizes, and the version.

|  |  |  |  |
| --- | --- | --- | --- |
| Deleted Date | File Name | File Size (bytes) | Version |
| 03 11 201900:03:37 UTC | C:\Users\HP\Desktop\73.docx | 12 K | Windows 10 |
| 03 11 201900:03:37 UTC | C:\Users\HP\Desktop\73.txt | 1 K | Windows 10 |
| 03 11 201900:03:37 UTC | C:\Users\HP\Desktop\73.png | 43 K | Windows 10 |