TZWorks® Windows Trash Inspection & Analysis (*tia*) Tool Users Guide

Abstract

tia is a standalone, command-line tool that can parse Windows recycle bin metadata files. ***tia*** can operate on a live volume, an image of a volume or a collection of artifact files obtained from another source. There are binary versions that run on Windows, Linux and Mac OS-X.

www.tzworks.net

Contact Info: info@tzworks.net

*Document applies to v0.27 of **tia***

Updated: Sept 9, 2020

Table of Contents

1 Introduction 2

1.1 Background 2

2 How to Use *tia* 3

2.1 Other Scan Types 4

2.2 Volume Shadow Copies 5

2.3 Pulling in a directory of trash artifacts 5

2.4 Reports 5

3 Available Options 6

4 Authentication and the License File 8

4.1 *Limited* versus *Demo* versus *Full* in the tool's Output Banner 9

5 References 9

TZWorks® Windows Trash Inspection & Analysis (*tia*) Users Guide

Webpage: http://www.tzworks.net/prototype_page.php?proto_id=38
Contact Information: info@tzworks.net

1 Introduction

tia is a command line version of a tool to parse Windows *recycle bin* artifacts. The tool was designed to work with the different versions of *recycle bin* formats from WinXP to Win10. ***tia*** analyzes the metadata stored for each file that is deleted, and does not try to restore the deleted file. The type of data that can be obtained through this analysis includes: (a) name of the file that was deleted, (b) time of deletion, (c) size of the deleted file and (d) the security identifier of the user account that deleted the file. The reports generated are various flavors of CSV output to allow maximum flexibility in exporting the results to a spreadsheet or another database.

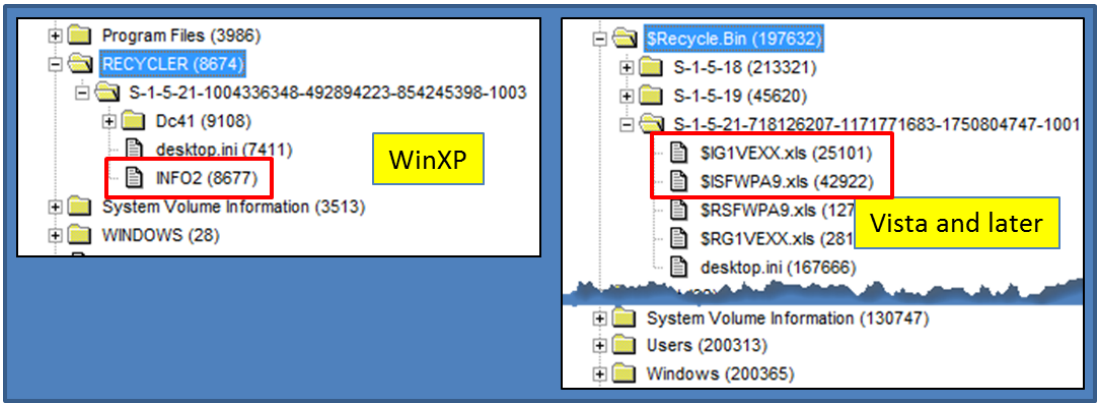
1.1 Background

In Windows, when a user deletes a file, the operating system renames the file and then puts it into a temporary directory. It stays in the temporary directory until the trash is emptied. During the deletion process, and under the covers, Windows creates another companion file that contains the metadata associated with the deleted file. This metadata contains the deletion time along with the path/name of the file or folder that was deleted.

The temporary folder that stores the deleted entry and its associated metadata is located in the *Recycle Bin* directory. For Windows XP, this root directory is the *Recycler* folder and the target file containing the metadata is the *info2* file. Each account on the machine has its own subfolder. This subfolder has its name defined by the SID (or Security Identifier) for the user account. For example when looking at the C: drive, it would be *C:\Recycler\{account SID}\info2*.

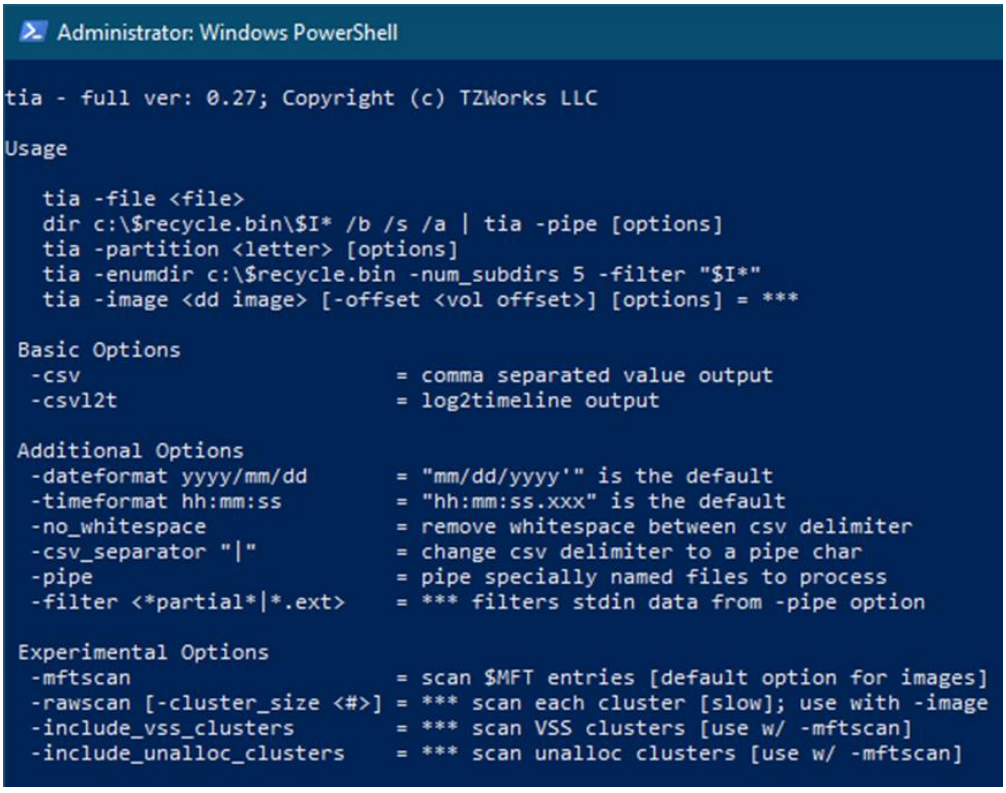
For Windows Vista and later, the root directory is the *$Recycle.Bin* and the metadata files are the ones prefixed with the characters *$I* followed by some random characters (which are also located in each respective account).

From a forensics standpoint, parsing the metadata in these files allows one to view which and when files and/or folders were deleted, as well as, which user account was responsible for the deletion. Below is a screenshot of the two types of trash directory structures for Windows. The first is for Windows XP and the second applies to Vista on up to Win10. The highlighted areas in red are the file types ***tia*** processes.



2 How to Use *tia*

The ***tia*** tool is flexible in that it allows one to process trash artifacts from a number of sources. For example, one can: (a) parse an individual *recycle bin* metadata file, (b) target a *recycle bin* directory on a specified volume, (c) scan/carve the entire volume for *recycle bin* metadata information, or (d) process *recycle bin* metadata in an offline manner by feeding in artifacts into STDIN (standard input). Below is a screen shot of the command menu that shows which options are available.



The first two options shown above (***-file*** and ***-pipe***) are used if desiring to parse individual files. If one has a collection of *recycle bin* files that are desired to be processed offline, the ***-pipe*** option is the best option, since it will rip through all the subdirectories parsing out each file in turn.

If one cannot use the ***-pipe*** option, one can use the experimental ***-enumdir*** option, which has similar functionality with more control. The ***-enumdir*** option takes as its parameter the folder to start with. It also allows one to specify the number of subdirectories to evaluate using the ***-num_subdirs <#>*** sub-option.

If one wants to operate on a live system, one can just use the ***-partition*** option and specify the drive letter. In this mode, ***tia*** will automatically scan the proper *recycle bin* artifacts and parse those on the specified volume.

The last main category is the ***-image*** option, which is used to target 'dd' images of volumes or any chunk of data. For those cases where a 'dd' image of an entire drive is present, one can use the ***-offset*** parameter to specify the offset of the volume to target on the drive. On the other hand, if one desires to scan a blob of data, then the only constrains are: (a) the data is not encrypted, (b) the data is not compressed, and (c) the data is aligned on cluster boundaries. In this mode, ***tia*** will scan each cluster and just look for trash artifact signatures and parse any it finds.

2.1 Other Scan Types

The basic scan types are to target the *recycle bin* directory and parse any artifact data. In addition to the artifacts in this directory, there are other areas that can be scanned as well. For example, one can sometimes find *recycle bin* artifacts that have been flushed from the *recycle bin* directory (eg. permanently deleted), but their clusters have not been overwritten yet. There are 3 main sources for these deleted artifacts: (a) The MFT entry listing which may have an 'unrecycled' MFT entry that contains a deleted *recycle bin* artifact, (b) The Volume shadow store has clusters that may contain prior snapshots with trash data (embedded in the $MFT file record), and (c) any unallocated clusters.

To handle these other areas, one can use the following options:

Option	Targets
-mftscan	$MFT entries on target volume [default option for images]
-rawscan	All clusters on target volume; brute force scan [use for images]
-include_vss_clusters	Volume Shadow store clusters on target [images or partitions]
-include_unalloc_clusters	Unallocated clusters on target volume [images or partitions]

The above scans will look at data and determine the artifact type (as far as OS version) and parse it appropriately. For the cases where multiple versions of Windows trash artifacts are found, then the output will display the results annotated with what ***tia*** thought the artifact signature version was.

When using any of the above options, ***tia*** will report the offset the trash metadata was found at. For the ***-mftscan***, ***tia*** will also try to build the path of where it located the *info2* or *$I* file from, which can be useful. This allows the user account to be determined, by analyzing the subfolder containing the SID.

The ***-rawscan*** option is a cluster based scan with a twist. It will traverse each cluster and see if there is a signature at the start of the cluster with a *recycle bin* entry. Secondarily, it will also see if any MFT file records are within the cluster (at the MFT file record boundary). This second scan is to check if there are any *recycle bin* data within the resident data of the MFT entry. In the default mode, the tool looks at the internals of the volume to try to determine the cluster size to use. If the tool cannot find the cluster size this way (and one is not explicitly specified), then it will use 4096 byte chunk as the size. To force a different cluster size, one can use the sub-option ***-cluster_size <#>***. One should keep in mind, since this scan option looks at each cluster, it is the slowest of the scanning options.

2.2 Volume Shadow Copies

When using the ***-include_vss_clusters*** option, ***tia*** will locate each of the Volume Shadow Stores and their respective cluster runs. From this list, the tool will then run in *semi-rawscan* mode by analyzing each of the clusters for signature types associated with *recycle bin* artifacts.

As a point of clarification, ***tia*** will not try to reconstruct the VSS volume and the various Snapshots over time, but will do a brute force scan on the raw clusters associated with the shadow store. The results will be annotated so the analyst can see which entry came from which shadow store along with the image offset.

2.3 Pulling in a directory of trash artifacts

Since ***tia*** is agnostic to the version of recycle bin format, one can mix and match different recycle bin artifacts within a session run. ***tia*** has a separate field what it thought the format version of Windows the trash artifact came from, shown below as 'ver'.

cmdline: <filelist> ... | tia64 -pipe -csv -dateformat yyyy/mm/dd -base10 -out results1.csv

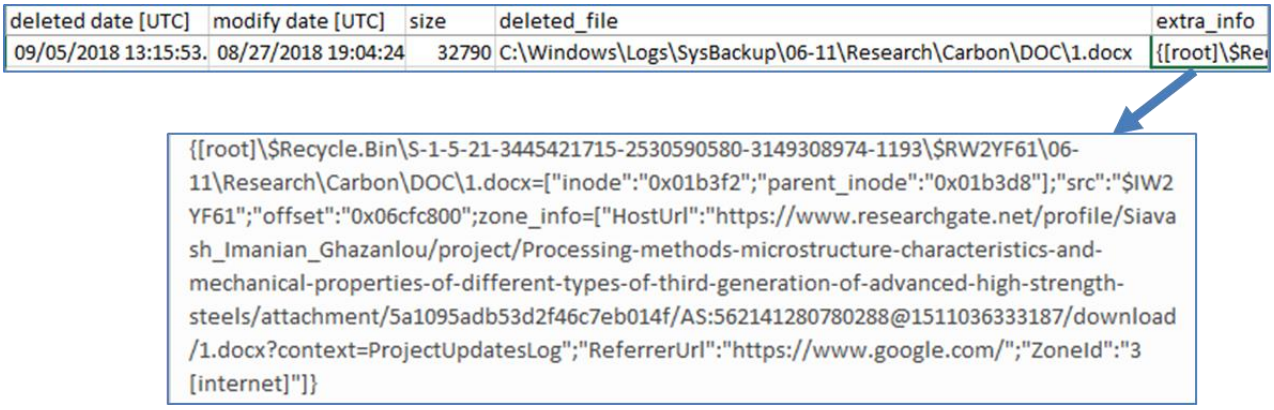
deleted date [UTC]	size	ver	deleted_file	user_sid
2009/01/16 23:27:24.503	34816	WinXP	C:\Documents and Settings\Donald Blake\My	S-1-5-21-1004336348-492894223
2016/10/03 15:10:25.744	0	Win7 or 8	C:\dump\webpage.hidden	S-1-5-21-2590462788-5707957-2
2016/10/03 15:10:25.744	856902	Win7 or 8	C:\dump\webpage.hidden\EBBLS8A7.mht	S-1-5-21-2590462788-5707957-2
2016/10/03 15:10:25.744	259186	Win7 or 8	C:\dump\webpage.hidden\Waves of Bank Fa	S-1-5-21-2590462788-5707957-2
2016/10/03 15:10:25.744	0	Win7 or 8	C:\dump\webpage.hidden	S-1-5-21-2590462788-5707957-2
2016/10/03 15:10:25.744	856902	Win7 or 8	C:\dump\webpage.hidden\EBBLS8A7.mht	S-1-5-21-2590462788-5707957-2
2016/10/03 15:10:25.744	259186	Win7 or 8	C:\dump\webpage.hidden\Waves of Bank Fail	S-1-5-21-2590462788-5707957-2
2016/10/03 15:10:25.744	0	Win7 or 8	C:\dump\webpage.hidden	S-1-5-21-2590462788-5707957-2

2.4 Reports

The -csv option outputs the data in the form of a single line per entry, where each field is separated by a delimiter of your choice. There is also a -csvl2t option to output the data in a timeline fashion. For dates, the tool pulls the MACB data from the $Rxxx file and the delete timestamp from the $I data. For

those $Rxxx files that do not have a companion $Ixxx file, the MFT change timestamp is used for the deletion date.

For the scan options that go into the internals the NTFS metadata, such as the ***-mftscan***, ***-include_vss_clusters***, etc, then additional miscellaneous metadata is displayed; this includes; the disk/volume offset, MFT record numbers, and sometimes the Zone Identifier information if available. Since this additional data can be voluminous, the data in this field is formatted in a quasi-JSON format where key/value pairs are used. Below is an example of the type of data extracted and the way it is displayed in this field



In this case, the metadata includes: (a) the source of the $Rxxx file is shown, along with the inode for the entry; (b) the companion $I file, (c) where artifact was found in the volume, and (d) the Zone Information that was attached to the $Rxxx file. The Zone Information gives the analyst more insight, in that the file that was deleted was downloaded from the Internet, along with the URLs associated with the downloaded file.

For other output options, such as HTML, JSON, or SQLite, refer to the ***csvdx*** tool from TZWorks, which can take the output from ***tia*** (or any other TZWorks tool) and reformat the output to one of those listed.

3 Available Options

The options labeled as 'Extra' require a separate license for them to be unlocked.

Option	Extra	Description
-file		Specifies the file to parse. Syntax is ***-file <filename>***
-pipe		Used to pipe files into the tool via STDIN (standard input). Each file passed in is parsed in sequence.
-enumdir	*	Experimental. Used to process files within a folder and/or subfolders. Each file is parsed in sequence. The syntax is ***-enumdir***

		<folder> -num_subdirs <#>.
-partition		Windows only option. Extracts artifacts from a mounted Windows volume. The syntax is ***-partition <drive letter>***.
-image	*	Extracts artifacts from a Windows volume specified by an image and volume offset. The syntax is ***-image <filename> -offset <volume offset>***
-filter	*	Filters data passed in via STDIN via the ***-pipe*** or ***-enumdir*** options. The syntax is ***-filter <"*.ext | *partialname* | ..."***>. The wildcard character '*' is restricted to either before the name or after the name.
-csv		Outputs the data fields delimited by commas. Since filenames can have commas, to ensure the fields are uniquely separated, any commas in the filenames get converted to spaces.
-csvl2t		Outputs the data fields in accordance with the log2timeline format.
-no_whitespace		Used in conjunction with ***-csv*** option to remove any whitespace between the field value and the CSV separator.
-csv_separator		Used in conjunction with the ***-csv*** option to change the CSV separator from the default comma to something else. Syntax is ***-csv_separator "|"*** to change the CSV separator to the pipe character. To use the tab as a separator, one can use the ***-csv_separator "tab"*** OR ***-csv_separator "\t"*** options.
-dateformat		Output the date using the specified format. Default behavior is ***-dateformat "mm/dd/yyyy"***. This allows more flexibility for a desired format. For example, one can use this to show year first, via ***"yyyy/mm/dd"***, or day first, via ***"dd/mm/yyyy"***, or only show 2 digit years, via the ***"mm/dd/yy"***. The restriction with this option is the forward slash (/) symbol needs to separate month, day and year, and the month is in digit (1-12) form versus abbreviated name form.
-timeformat		Output the time using the specified format. Default behavior is ***-timeformat "hh:mm:ss.xxx"*** One can adjust the format to microseconds, via ***"hh:mm:ss.xxxxxx"*** or nanoseconds, via ***"hh:mm:ss.xxxxxxxxx"***, or no fractional seconds, via ***"hh:mm:ss"***.

		The restrictions with this option is a colon (:) symbol needs to separate hours, minutes and seconds, a period (.) symbol needs to separate the seconds and fractional seconds, and the repeating symbol 'x' is used to represent number of fractional seconds.
-quiet		This option suppresses any intermediate progress during a session run
-mftscan		This switch tells the tool to target $MFT entries during the scan
-rawscan	*	Scan each cluster for *recycle bin* metadata signatures. If found they are carved and parsed. Since this is a cluster based scan, there is a sub-option ***-cluster_size*** to force the search to use the specified boundary. The value given needs to be consistent with the normal cluster sizes available for Windows. Without specifying a cluster size, the tool will try to determine the proper boundary by looking at the internals of the volume. For the scan to be effective, the data cannot be encrypted or compressed.
-include_vss_clusters	*	This switch tells the tool to target volume shadow store clusters
-include_unalloc_clusters	*	This switch tells the tool to target unallocated clusters

4 Authentication and the License File

This tool has authentication built into the binary. There are two authentication mechanisms: (a) the digital certificate embedded into the binary and (b) the runtime authentication. For the first method, only the Windows and Mac OS-X (if available) versions have been signed by an X-509 digital code signing certificate, which is validated by Windows (or OS-X) during operation. If the binary has been tampered with, the digital certificate will be invalidated.

For the second (runtime authentication) method, the authentication does two things: (a) validates that the tool has a valid license and (b) validates the tool's binary has not been corrupted. The license needs to be in the same directory of the tool for it to authenticate. Furthermore, any modification to the license, either to its name or contents, will invalidate the license. The runtime binary validation hashes the executable that is running and fails the authentication if it detects any modifications.

4.1 *Limited* versus *Demo* versus *Full* in the tool's Output Banner

The tools from *TZWorks* will output header information about the tool's version and whether it is running in *limited, demo* or *full* mode. This is directly related to what version of a license the tool authenticates with. The *limited* and *demo* keywords indicates some functionality of the tool is not available, and the *full* keyword indicates all the functionality is available. The lacking functionality in the *limited* or *demo* versions may mean one or all of the following: (a) certain options may not be available, (b) certain data may not be outputted in the parsed results, and (c) the license has a finite lifetime before expiring.

5 References

1. "Cyber Dumpster-Diving: $Recycle.Bin Forensics for Windows 7 and Windows Vista", Paper by Timothy R. Leschke, US DoD Cyber Crime Institute.
2. "Once Upon a Time in Recycle Bin", http://4n6xplorer.com/forensics/once-upon-a-time-in-recycle-bin
3. The Forensic Analysis of the Microsoft Windows Vista Recycle Bin, by Mitchell Machor, 1/22/2008.