

GitHub

This repository ▾

Search or type a command 🔍

Explore

Features

Enterprise

Blog

Sign up

Sign in

.IC

williballenthin / INDXParse

★ Star

22

🍴 Fork

4

Tool suite for inspecting NTFS artifacts. <http://www.williballenthin.com/forensics/mft/indxparse/>

📌 141 commits

🌿 7 branches





















📦 18 releases

👤 3 contributors

🔗

branch: master ▾

INDXParse / +

add CONTRIBUTORS.txt		
Willi Ballenthin authored 2 months ago latest commit 81c037d4e1		
 .gitignore	update gitignore	5 months ago
 BinaryParser.py	add MFT utils, like list_mft and get_file_info	6 months ago
 CHANGELOG	start CHANGELOG	5 months ago
 CONTRIBUTORS.txt	add CONTRIBUTORS.txt	2 months ago
 FileMap.py	add fuse-mft.py and supporting code	5 months ago
 INDXParse.py	make INDXParse use the function, not the statement	4 months ago
 LICENSE	add license file	3 years ago
 MFT.py	remove bad logging statement	3 months ago
 MFTINDX.py	add MFT utils, like list_mft and get_file_info	6 months ago
 MFTView.py	remove debugging statements from MFTView.py	10 months ago
 Progress.py	fix import issue with progressbar and the version in pip	4 months ago
 README	fix bug reported by Jerome Leseinne in parsing INDX_ALLOCATION attrib...	5 months ago
 SDS.py	implement most of the INDEX structs using Block notation.	a year ago
 SDS_get_index.py	add basic tool for printing out the SDS index entries	7 months ago
 SortedCollection.py	add fuse-mft.py and supporting code	5 months ago
 extract_mft_record_slack.py	refactor record data extraction. add active/slack strings to get-file...	4 months ago
 fuse-mft.py	fix import issue with ProgressBar	4 months ago
 get_file_info.py	fix timeline entries in get_file_info	3 months ago
 list_mft.py	fix bug in list_mft when a path prefix is provided. users would recei...	2 months ago
 tree_mft.py	add fuse-mft.py and supporting code	5 months ago

<> Code

🔔 Issues

7

🔗 Pull Requests

0

📶 Pulse

📊 Graphs

🌐 Network

HTTPS clone URL

https://github.com

📄

You can clone with HTTPS or Subversion. ⓘ

🖥 Clone in Desktop

📄 Download ZIP

📖 README

INDXParse

=====

Introduction

INDX files are features of the Windows NTFS file system. They can be thought of as nodes in a B+ tree, where each directory has an INDX file. The INDX files contain records for each file within a directory. Records contain at least the following information:

- Filename

- Physical size of file

- Logical size of file
- Modified timestamp
- Accessed timestamp
- Changed timestamp
- Created timestamp

INDX files are interesting to forensic investigators for a number of reasons. First, an investigator may use INDX files as a source of timestamps to develop a timeline of activity. Secondly, these files have significant slack spaces. With careful parsing, an investigator may recover old or deleted records from within these data chunks. In other words, the investigator may be able to show a file existed even if it has been deleted.

INDX files are not usually accessible from within the Windows operating system. Forensic utilities such as the FTK Imager may allow a user to extract the file by accessing the raw hard disk. FTK names the INDX file "\$I30". Tools like the Sleuthkit can extract the directory entries from a forensic image. INDXParse will not work against a live system.

Previous work & tools

I'd like to first mention John McCash, who mentioned he was unaware of any non-EnCase tools that parse INDX files in a SANS blog post. That got my mental gears turning.

I started out with a document called NTFS Forensics: A Programmers View of Raw Filesystem Data Extraction by Jason Medeiros. Unfortunately, while this document describes parsing INDX files in detail, a number of steps in the explanation were wrong.

The second resource I used, and used extensively, was Forensic computing by A. J. Sammes, Tony Sammes, and Brian Jenkinson. I found the relevant section was available for free via Google books. This was an excellent document, and I now plan on buying the full book.

42 LLC provides the INDX Extractor Enpack as a compiled EnScript for EnCase. This was not useful to me, because I was unable to get to the logic of the script.

The Sleuthkit has INDX structures defined in the `tsk_ntfs.h` header files. I didn't do much digging in the code to see if TSK does any parsing of the INDX files (I suspect it does), but I did use it to verify the file structure.

Usage

INDXParse.py accepts a number of command line parameters and switches that determine what data is parsed and output format. INDXParse.py currently supports both CSV (default) and Bodyfile (v3) output formats. The CSV schema is as follows:

- Filename
- Physical size of file
- Logical size of file
- Modified timestamp
- Accessed timestamp
- Changed timestamp
- Created timestamp

INDXParse.py will parse INDX structure slack space if provided

the '-d' flag. Entries identified in the slack space will be tagged with a string of the form "(slack at ###)" where ### is the hex offset to the slack entry. Note that slack entries will have separate timestamps from the live entries, and could be used to show the state of the system at a point in time.

If the program encounters an error while parsing the filename, the filename field will contain a best guess, and the comment "(error decoding filename)". If the program encounters an error while parsing timestamps, a timestamp corresponding to the UNIX epoch will be printed instead.

The full command line help is included here:

```
INDX $ python INDXParse.py -h
usage: INDXParse.py [-h] [-c | -b] [-d] filename
```

Parse NTFS INDX files.

positional arguments:

filename Input INDX file path

optional arguments:

-h, --help show this help message and exit
-c Output CSV
-b Output Bodyfile
-d Find entries in slack space

INDXTemplate.bt is a template file for the useful 010 Editor.
Use it as you would any other template by applying it to INDX files.

TODO

- Brainstorm more features ;-)

License

INDXParse is released under the Apache 2.0 license.

Contributors

- Jerome Leseinne for identifying a bug in the is_valid constraint and null blocks

