Table of Contents

About The Project

The goal of this project is to create a Python tool that can extract different types of artifacts from EWF (Expert Witness Compression Format) evidence files. This tool can be useful for forensic investigators who need to analyze digital evidence and extract relevant information from EWF files.

The tool provides various commands to extract different types of artifacts, such as automatic extraction, browser artifacts, logs artifacts, registries artifacts, and hives artifacts. The tool also supports commands to list partitions, list folders, copy files, and print files.

This tool use a Merkle proof to validate the extracted artifacts.

(back to top)

Built With

Python

The tool is implemented using Python and uses libraries such as Pyewf, Pytsk3, and PyYAML to extract and parse data from EWF files. The tool can be run from the command line by executing the main.py script with the appropriate command and options.

Structure

```
cmds/
    cmd1.py
config/
    config1.yml
```

```
utils/
    util1.py
main.py
```

`cmds/` This folder is used for all the subcommands

`config/` This folder is used for all the YAML config profile.

`utils/` This folder is used for all the python utils.

(back to top)

YAML Configuration File

This repository includes a default YAML configuration file located at config/default.yml that can be used to customize certain aspects of the program.

Configuration Options The following configuration options are available:

DATA_PARTITION A binary-encoded string representing the data partition. This option is used to specify the location of the data partition.

CONFIG_DIR_PATH The path to the configuration directory. This option is used to specify the path to the system configuration directory.

LOGS_DIR_PATH The path to the logs directory. This option is used to specify the path to the logs directory.

MFT_PATH The path to the Master File Table (MFT). This option is used to specify the path to the MFT.

USERS_DIR_PATH The path to the users directory. This option is used to specify the path to the users directory.

LOGS_FILES A list of log files to extract. This option is used to specify which log files should be extracted.

REG_FILES A list of registry files to extract. This option is used to specify which registry files should be extracted.

USER_HIVES A list of user hives to extract. This option is used to specify which user hives should be extracted.

EDGE_PREFIX The prefix used for Microsoft Edge artifacts.

CHROME_PREFIX The prefix used for Chrome artifacts.

FIREFOX_PREFIX The prefix used for Firefox artifacts.

IE_PREFIXES A list of prefixes used for Internet Explorer artifacts.

BRAVE_PREFIXES A list of prefixes used for Brave artifacts.

(back to top)

Getting Started

This is an example of how you may give instructions on setting up your project locally. To get a local copy up and running follow these simple example steps.

Prerequisites

You will need to install the latest version of Python 3.10, which can be downloaded from the official Python website or installed using your system's package manager.

Installation

1. Clone the repo

   ```
   git clone https://github.com/MohammedBenhelli/EWFParser
   ```

2. Install python packages

   ```
   cd ./EWFParser
   pip install -r requirements.txt
   ```

(back to top)

Usage

Automatic extraction

This command extracts all available artifacts from the EWF file. It can be run using the following command:

```
python main.py extract --file <path-to-ewf-evidence> --dest <optional-path-to-destination> -
```

Extract browsers artifacts

This command extracts browser artifacts from the EWF file. It can be run using the following command:

```
python main.py browsers --file <path-to-ewf-evidence> --dest <optional-path-to-destination>
```

Extract logs artifacts

This command extracts logs artifacts from the EWF file. It can be run using the following command:

```
python main.py logs --file <path-to-ewf-evidence> --dest <optional-path-to-destination> --co
```

Extract registries artifacts

This command extracts registries artifacts from the EWF file. It can be run using the following command:

```
python main.py reg --file <path-to-ewf-evidence> --dest <optional-path-to-destination> --cor
```

Extract hives artifacts

This command extracts hives artifacts from the EWF file. It can be run using the following command:

```
python main.py hives --file <path-to-ewf-evidence> --dest <optional-path-to-destination> --c
```

Get a proof

This command verify the proof of a folder. It can be run using the following command:

```
python main.py get-proof <path-to-artifacts-folder>
```

Verify a proof

This command verify the proof of a folder. It can be run using the following command:

```
python main.py verify  --directory <path-to-artifacts-folder> --proof <proof-hash>
```

List partitions

This command lists the partitions in the EWF file. It can be run using the following command:

```
python main.py partition --file <path-to-ewf-evidence>
```

List folder

This command lists the contents of a folder in the EWF file. It can be run using the following command:

```
python main.py ls --file <path-to-ewf-evidence> <path-to-folder>
```

Copy file

This command copies a file from the EWF file to a destination folder. It can be run using the following command:

```
python main.py cp --file <path-to-ewf-evidence> <path-to-file> <path-to-destination>
```

Print file

This command prints the contents of a file in the EWF file. It can be run using the following command:

```
python main.py cat --file <path-to-ewf-evidence> <path-to-file>
```

(back to top)

Roadmap

- ☒ Extracting system registries and users hives
  - ☒ Convert to JSON
- ☒ Extracting logs
  - ☒ Convert to XML
- ☒ Add subcommand
  - ☒ `partition` command
  - ☒ `ls` command
  - ☒ `cp` command
  - ☒ `get-proof` command
  - ☒ `verify` command
  - ☒ `cat` command
  - ☒ `browsers` command
  - ☒ `logs` command
  - ☒ `hives` command
  - ☒ `reg` command
- ☒ Merkle proof for file signature check
- ☒ Extracting Browsers
  - ☒ Edge
  - ☒ Chrome
  - ☒ Internet Explorer
  - ☒ Firefox
  - ☒ Brave
  - ☒ Opera
  - ☒ Puffin
- ☒ Extracting MFT
  - ☐ Parsing MFT
- ☐ Refactor
  - ☐ Clean code
  - ☐ Add ruff linter

(back to top)

License

Distributed under the MIT License. See `LICENSE.txt` for more information.

(back to top)