Multiplatform DFIR Live Triage and Collection Tool

Breakpoint Forensics

Release Notes & Guide

Release Notes

V1.3.7:

1/2/23 -Fix for crash if Keyword Filtering Enabled and keyword file empty.

-Added support for common archive formats to VICS JSON creation.

V1.3.6:

10/26/22 – Hotfix for incomplete Relative Path value when generating VICS JSON that would cause import error in Griffeye.

V1.3.5:

10/24/22 – Initial Public Release

Requirements and Tested Platforms:

- Windows (X64) Versions: 7, 10 or 11
- MacOS 11.7-12.6 (Intel X64 or M1-ARM))
- Ubuntu 20.04 -22.04(X64)

Contents

About

FileSifter is a digital forensics live-triage collection tool designed for deployment across multiple OS platforms including Windows, MacOS, and Linux.

The portable tool is meant to offer a means of efficiently triaging the contents of a running computer, its hard drive, or any other storage device and its associated file-system. Ideal scenarios where this might be employed is in situations where the computer or device might be encrypted and can't be shut down for a typical dead-box examination of the storage device.

In these situations, an examiner would historically be limited to conducting a live acquisition of every file on the computer or storage device. While this might be effective, it can significantly increase the time it takes to review the contents of a device, as the examiner has no real-time feedback as to the contents of the device. Additionally, critical time might be wasted as data is being collected that may not be relevant. These are some of the situations that File Sifter is designed for.

Setup

File sifter is packaged and distributed in 3 current versions covering Windows, MacOS, and Linux executables.

As the tool is designed to be executed as a Live Triage Tool, its executable requires no ‘installation’ and runs as a portable application.

Setup only requires copying the appropriate executable to an examiners collection drive, where it is then executed from.

All collected files will automatically be stored on this collection drive or wherever the FileSifter executable is located.

Primary Features

-Live File Collection to either ZIP or TAR packages.

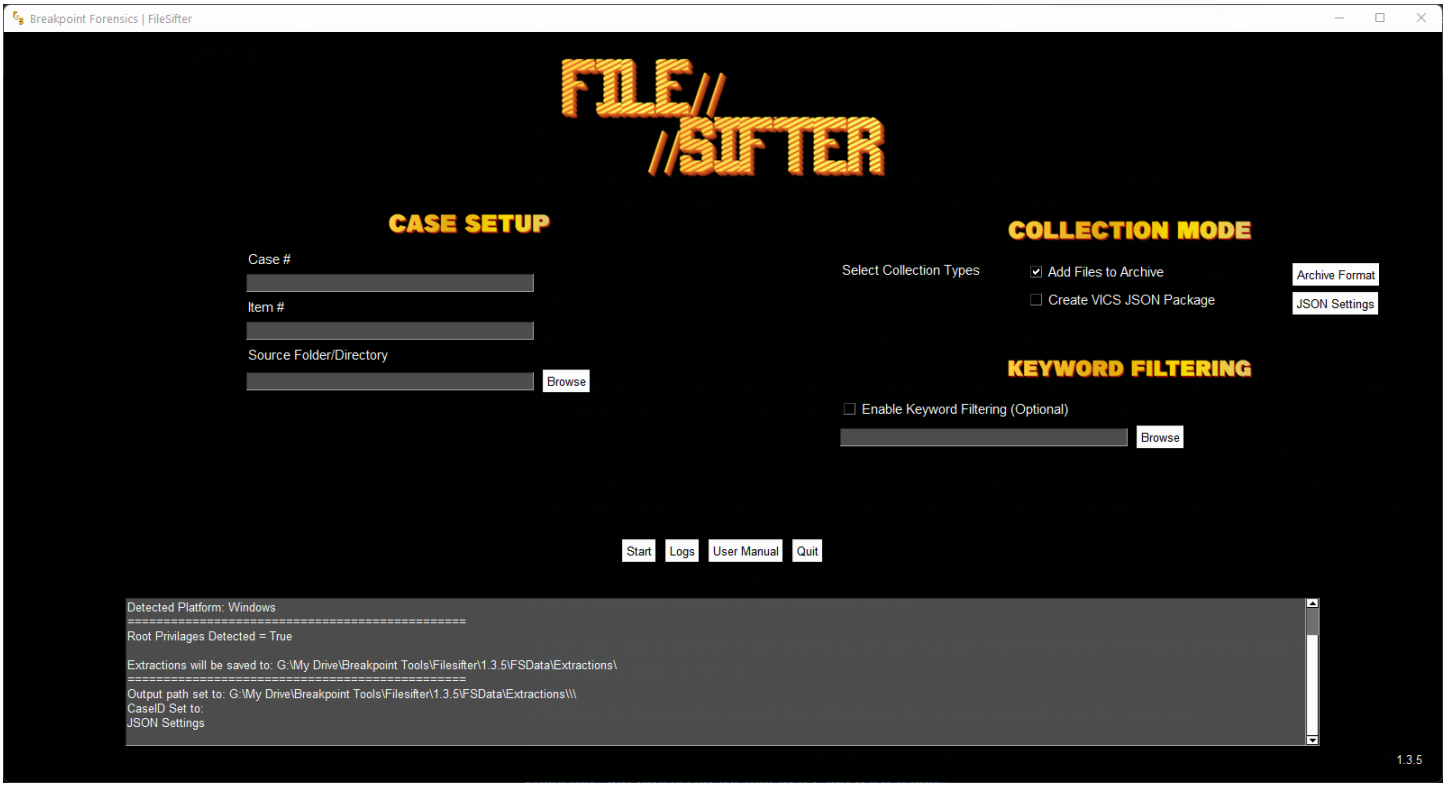
-Keyword Filtering function. Allows import of custom keyword dictionary file that when enabled will only collect files with match in keyword list.

-Easy targeting of files/folders to be collected using simple user interface and case setup.

-Support for targeted collection of Image, Video, Archives, and/or Documents and packaging into VICS JSON evidence package for easy import and review into tools such as Griffeye Analyze.

-Automatically generates CSV report for all files collected storing original metadata such as MAC times, paths, etc.

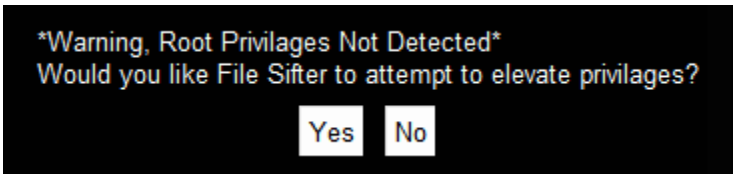
Main Interface



Startup

During initial execution of FileSifter, it will conduct a 'check' to see if it has administrative/root privileges. FileSifter *can operate* without these privileges but depending on the folders/files targeted for collection, may not be able to collect certain files without elevated privileges.

Therefore, depending on the OS Platform version of FileSifter being used, you will receive a 'warning' message if executed without elevated privileges. In Windows, you will be offered the option to attempt to relaunch as an 'Administrator' automatically.



In MacOS or Linux versions, you will receive this warning. However, due to OS specific limitations, you'll need to manually relaunch FileSifter with 'sudo'. A small 'helper' script is included with these versions that will do this automatically when executed from the same folder as FileSifter.

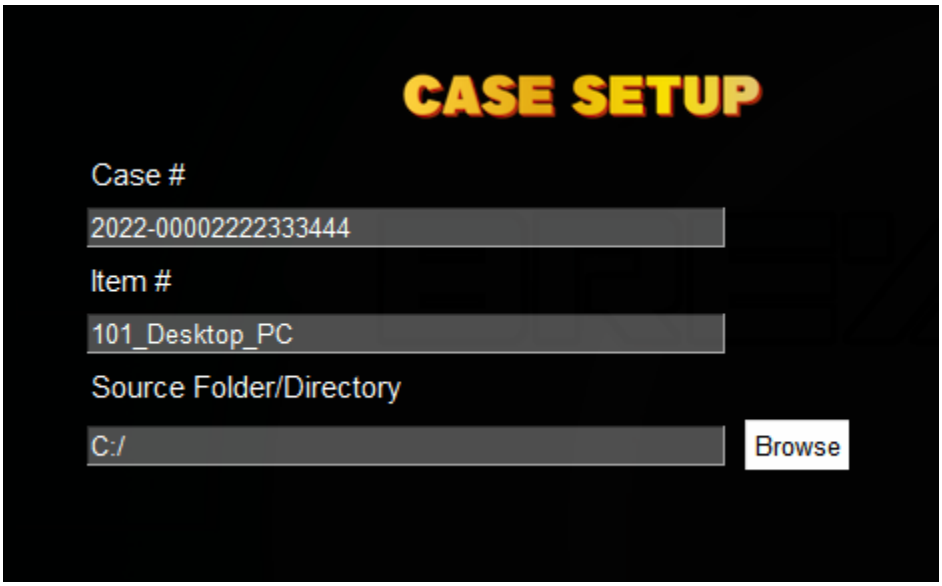
FileSifter

Start-with-sudo.sh

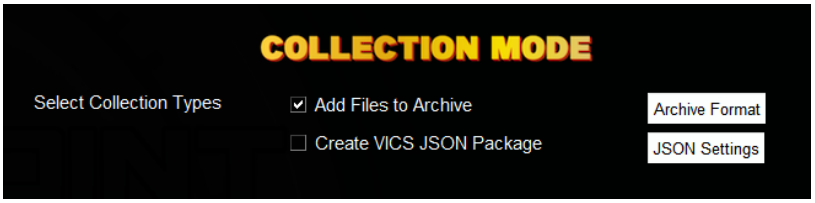
Case Setup

FileSifter organizes collected data based on a Case# and Item# schema. Simply enter the appropriate identifiers in the Case and Item fields.

Source Folder Directory: Use the folder browse button to select the folder that you would like FileSifter to process and collect files from. All FileSifter searches are executed in a recursive fashion, and will collect the selected folder, and subfolders, and all of their enclosed files based on your *Collection Mode* settings.



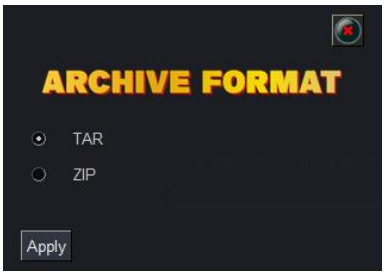
Collection Mode



FileSifter comes with two primary collection modes. They can be executed individually, or combined and run at the same time.

In addition, no matter which Collection Mode is set, all FileSifter jobs will generate a CSV file listing all files and their metadata that were collected by FileSifter.

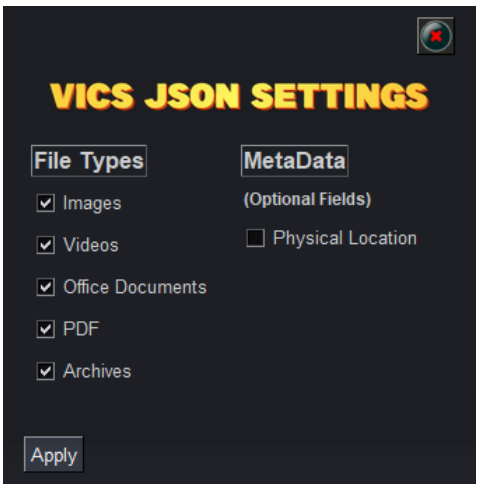
Add Files to Archive



When selected any files identified for collection will be added to an archive file on your collection drive.

Current archive formats include ZIP or TAR.

Create VICS JSON Package



When selected any files identified for collection will be added to a VICS JSON evidence package, and a VICS compliant JSON will be built.

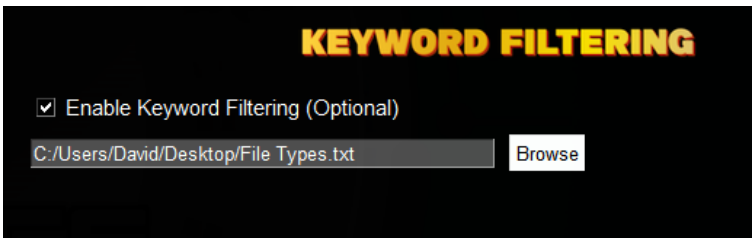
This JSON package can then be imported into a tool such as Griffeye or any other tool supporting VICS evidence packages for efficient review of media data.

The VICS JSON Package will be limited to known/common Image, Video, TXT/Document, and Archives types.

Note, for processing speed considerations the VICS function will only identify files based on their filename extension, and is not conducting real-time file-signature analysis.*

*Supported Extensions: ['.ai', '.bmp', '.cam', '.cr2', '.gif', '.heic', '.heif', '.ind', '.indd', '.j2k', '.jfi', '.jfif', '.jif', '.jp2', '.jpe', '.jpeg', '.jpf', '.jpg', '.jpx', '.k25', '.mj2', '.nrw', '.pct', '.png', '.psb', '.psd', '.psd', '.raw', '.rw2', '.svg', '.svgz', '.tif', '.tiff', '.wdp', '.x3f', '.xcf', 'arw', 'dib', 'dsc', 'eps', 'indt', 'jpm', 'webp', '.dv', '.3g2', '.3gp', '.amv', '.asf', '.avi', '.drc', '.f4a', '.f4b', '.f4v', '.flv', '.gifv', '.m2ts', '.m2v', '.m4p', '.m4v', '.mng', '.mov', '.mp2', '.mp4', '.mpe', '.mpeg', '.mpg', '.mts', '.nsv', '.ogg', '.ogv', '.rm', '.roq', '.svi', '.ts', '.viv', '.vob', '.wmv', '.yuv', 'f4p', 'm4v', 'mkv', 'mpv', 'mxf', 'rmvb', 'webm', '.doc', '.docx', '.xls', '.xlsx', '.ppt', '.pptx', '.odt', '.ods', '.odp', '.pdf', '.zip', '.rar', '.7z', '.dmg', '.tar', '.gz']

Keyword Filtering



If keyword filtering is enabled, the user is prompted to provide a text file containing keywords relevant to the exam, investigation, etc.

Keyword text files can have any number of keywords in them, and should be formatted with 1 keyword per line.

This will then limit the collected files to only those with a keyword match.

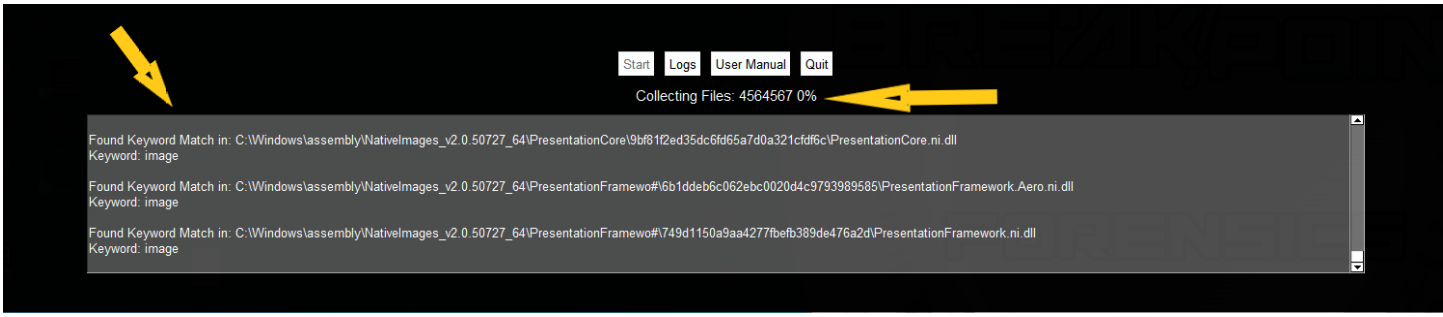
If a keyword match is found in the path or filename of a file, the file will be collected and added to the collection container/archive file. The file will also be processed for relevant metadata, and its MD5 hash-value computed.

This information is immediately logged to a CSV file, and the user is provided an immediate notification of the search-hit in the console.

Output Window and Status Bar

FileSifter utilizes an integrated output window. Various confirmation messages, processing status, errors, etc. will be printed here for reference.

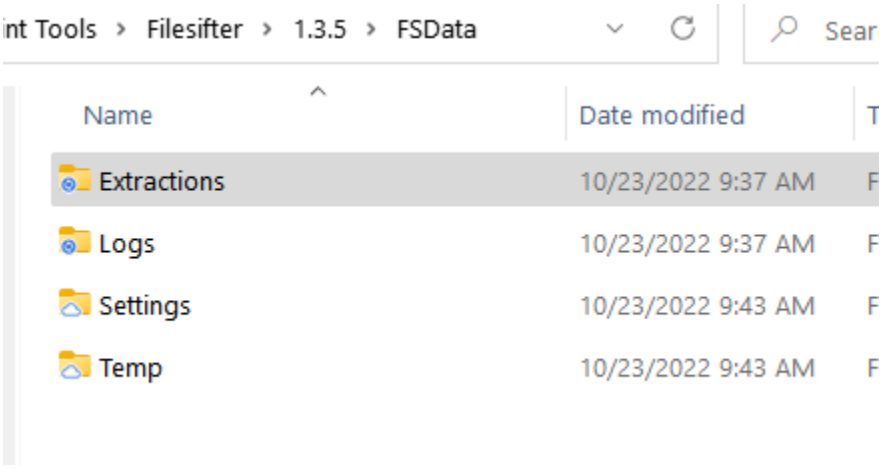
Directly above the window is a dynamic status bar that will occasionally update with processing status, progress indicators and completion percentages.



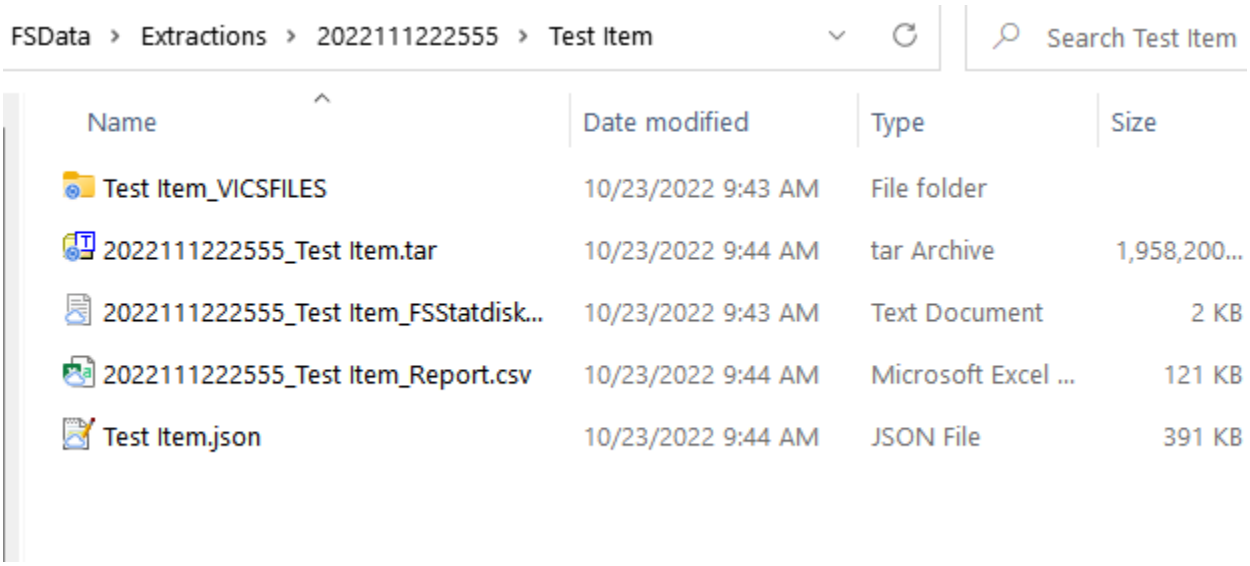
Post Processing

Data Output Locations

Following a FileSifter process you will find the recovered data, logs, and generated JSON's for each Item stored inside the FSData folder.



Specifically, they will be located inside a subfolder titled, {CaseID#} Inside that will be an additional subfolder {Item#} for each individual Item processed. Each Item will further have a number of files and subfolders that will vary based on the collection modes selected and may include JSON's, Archives, and other item specific reports.



File Metadata and VICS/CSV Output

Data recovered and imported using FileSifters VICS JSON Package and logged in the generated CSV, will collect and fille several metadata fields, and may vary depending on OS and Filesystem formats.

These will include but are not limited to the following:

SourceID: Based on Item# Field

Mime Type: Identified File Type Based on Extension

File Name/Path: As located on target item.

Create/Written/Accessed Timestamps: The timestamps collect from the original target machine during FileSifter collection.

JSON Example:


```
"value":[
    {
        "CaseID":"c98f0ff2-5dcc-4ec6-95ef-50341e7ae0dd",
        "Media":[
            {
                "MediaID":0,
                "MD5":"052B9CAD59E505CE78C718ADD0C54BA1",
                "MediaSize":2653999,
                "RelativeFilePath":"G:\\My Drive\\Breakpoint
                "MimeType":"image",
                "IsPrecategorized":false,
                "MediaFiles":[
                    {
                        "MD5":"052B9CAD59E505CE78C718ADD0C54E
                        "FileName":"052b9cad59e505ce78c718adc
                        "FilePath":"C:\\Users\\David\\Desktop
                        "Created":"2022-07-08T17:35:44+00:00"
                        "Written":"2022-07-08T17:35:44+00:00"
                        "Accessed":"2022-10-23T16:43:47+00:0C
                        "Unallocated":false,
                        "SourceID":"Test Item"
                    }
                ]
            },
            {
                "MediaID":1,
                "MD5":"0E588B9A05035C2F0208EB86D744F76E",
                "MediaSize":53985,
                "RelativeFilePath":"G:\\My Drive\\Breakpoint
                "MimeType":"image",
                "IsPrecategorized":false,
                "MediaFiles":[
                    {
                        "MD5":"0E588B9A05035C2F0208EB86D744F7
                        "FileName":"0e588b9a05035c2f0208eb86d
                        "FilePath":"C:\\Users\\David\\Desktop
                        "Created":"2022-07-08T17:35:44+00:00"
                        "Written":"2022-07-08T17:35:44+00:00"
                        "Accessed":"2022-10-23T16:43:47+00:0C
                        "Unallocated":false,
                        "SourceID":"Test Item"
```


CSV Example:

	A	B	C	D	E	F	G	H	I
1	Path	File	Size	Modified Time	Accessed Time	Created Time	MD5	Sector	Keyword
2	C:\Users\David\Desktop	1.txt	0	2022-09-18T11:49:16+00:00	2022-09-18T11:49:16+00:00	2022-09-18T11:49:16+00:00	D41D8CD98F00B204E9800998ECF8427E	0	doc
3	C:\Users\David\Desktop	test_DOCS	524288512	2022-06-09T15:07:59+00:00	2022-10-23T16:43:44+00:00	2022-06-09T15:07:21+00:00	E6E8449EB186FB5A7DB7FCD6C8E30F84	0	doc
4	C:\Users\David\Desktop	Analyze DI	82416	2022-07-08T14:06:16+00:00	2022-10-23T16:43:47+00:00	2022-06-08T23:02:36+00:00	5469792595311432B4F941E0A14F23F3	0	doc
5	C:\Users\David\Desktop	INSTRUCTC	124077	2022-09-15T09:59:09+00:00	2022-10-23T16:43:47+00:00	2022-09-15T09:59:08+00:00	F8B061C16420CD3AF90479B19106EE00	0	doc
6	C:\Users\David\Desktop	Scenario Ca	19041	2021-12-29T10:48:14+00:00	2022-10-23T16:43:47+00:00	2022-06-08T23:02:36+00:00	20316C679DC60118D9D7BB16251D6BA9	0	doc

Logs and Troubleshooting

Logging for operations initiated with FileSifter are maintained in 1 primary location.

FileSifter Logs

FileSifter specific logs, such as import status, errors, etc. are under the FSData folder on your collection drive, under 'Logs'. These can be directly accessed by clicking on the 'Logs' button at the bottom of the main interface. A new log is generated for each calendar day.

Default:

[CollectionDrivePath]\FSData\Logs

Case/Item Specific Logs

Additional logs specific to individual item collections may also be located under a specific items extraction folder.

Known Issue / Limitations

- Deciphering Physical Sector Value for collected files requires multiple additional operations and significantly increases collection time.
- Deciphering Physical Sector Value for collected files currently limited to NTFS formatted drives.
- Target Drive Parameters, Size/Formatting, and related stats are currently limited to FileSifter for Windows only. Linux and MacOS support will be added in future update.