CSE 406

Computer Security Sessional

Documentation Report
of
AUTOPSY
A Forensic Analysis Tool

Muhammad Ehsanul Kader

1805067

Md. Erfan Jahangir Chowdhury

1805080

Contents

1 Basic overview of Autopsy

1.1 Introduction

Autopsy is a powerful and versatile digital forensic tool that assists investigators in analyzing and processing evidence. With a user-friendly interface and robust capabilities, Autopsy simplifies the complexity of forensic investigations. From data acquisition to reporting and exporting, Autopsy provides comprehensive support throughout the entire investigative process hence trusted by law enforcement, military personnel, and corporate examiners.

Some of the things one can do using Autopsy:

- Investigate files
- Search keywords
- Parse archives
- Filter hashes
- Check integrity
- Recover data
- Recover files
- Examine pictures
- Analyze images
- Explore databases
- Analyze registries
- Explore emails
- Detect malware
- View events
- Review history
- Add bookmarks
- Create reports

1.2 Autopsy Workflow

Analyzing a data source in Autopsy uses the following workflow:

1. **Create a case:** Container for data sources and reports.
2. **Add a data source to the case:** One or more data sources can be added to a case.
3. **Run ingest modules on the data source:** Ingest modules operate in the background to analyze the data.
4. **Manual analysis:** Navigate through the data and ingest module results to Identify evidence.
5. **Report generation:** A final report based on selected tags or results.

1.3 Cases and Data Sources

Autopsy takes a case-centric approach, organizing and managing data sources with utmost precision. Each case needs at least one data source that we need to analyze. Supported types of data sources in Autopsy are:

- A disk image or VM File
- Local Disk
- Logical Files
- Unallocated Space Image Files
- Autopsy Logical Imager Results
- XRY Text Export

To analyze a data via Autopsy, we need to create a case and add the data source to the case. Every data source is associated with a host. After the data source is added to the case, we select the ingest modules to run on the data source. The ingest modules are run in the background and the results are stored in the case. We can then analyze the results and generate a report.

1.4 Ingest Modules

Ingest Modules analyzed the data in the data sources. They perform all of the analysis of the files and parse their contents.

The Ingest Modules in Autopsy are:

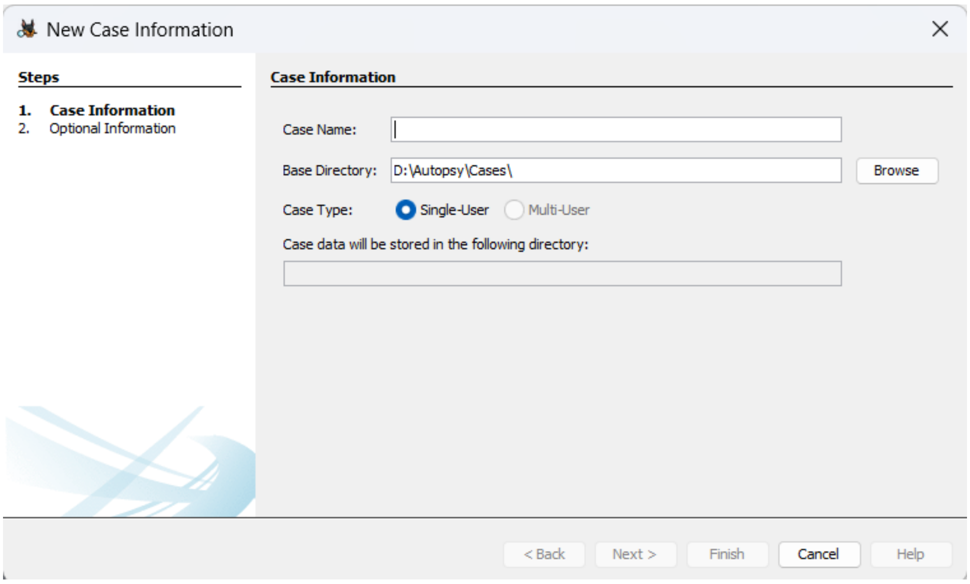
- Hash Lookup
- File Type Identification
- Embedded File Extraction
- Picture Analyzer
- Keyword Search
- Email Parser
- Extension Mismatch Checker
- Encryption Detection
- Android Analyzer
- Interesting Files Identifier
- PhotoRec Carver
- Data Source Integrity
- GPX parser

2 Creating a Case and Analysis

Launching Autopsy will display options to create a new case or open an existing case.

To create a new case, click on the **New Case** button. Add a case name and a base directory for the case.

Additionally, we can add a Case type: **Single User** or **Multi User**. Multi User cases are used when multiple users are working on the same case.



Case Name:

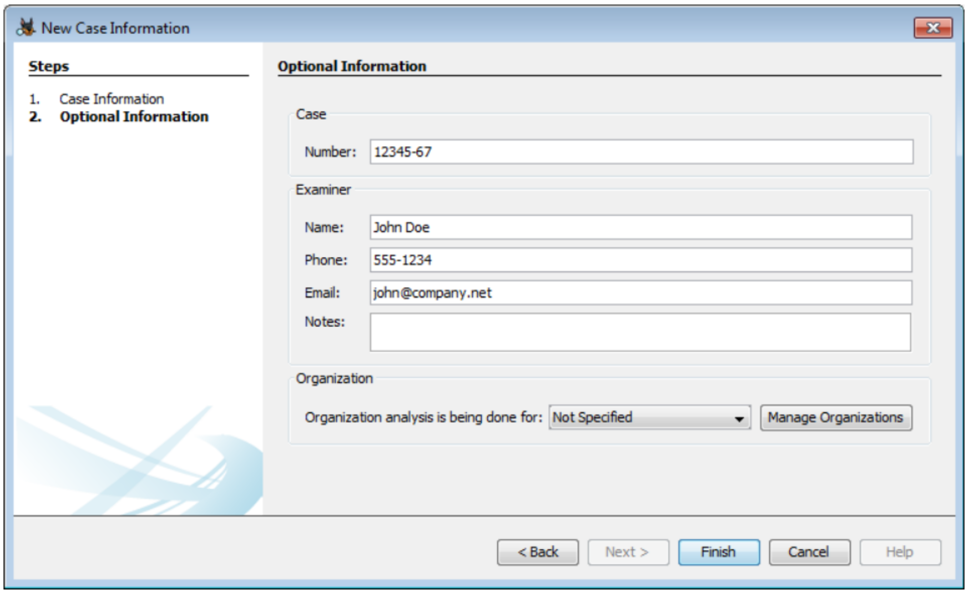
Base Directory: D:\Autopsy\Cases\

Case Type: ● Single-User ○ Multi-User

Case data will be stored in the following directory:

New Case Information

Then we will be prompted to add additional information about the case shown below:



Number: 12345-67

Name: John Doe

Phone: 555-1234

Email: john@company.net

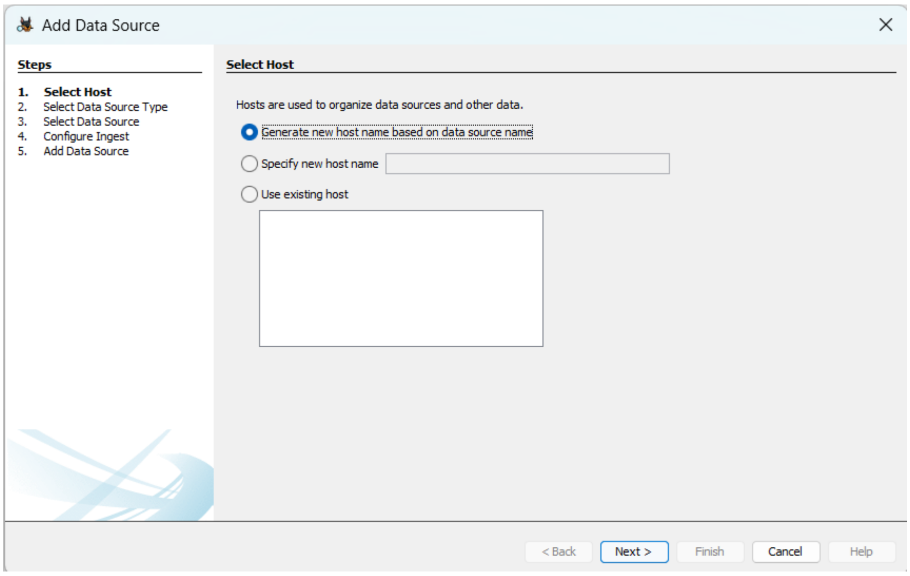
Notes:

Organization analysis is being done for: Not Specified

Additional Information

All fields on this panel are optional. Additionally, the Organization section will only be active if the central repository is enabled.

After creating the case, we will be prompted to add a data source shown below:



Hosts are used to organize data sources and other data.

● Generate new host name based on data source name

○ Specify new host name

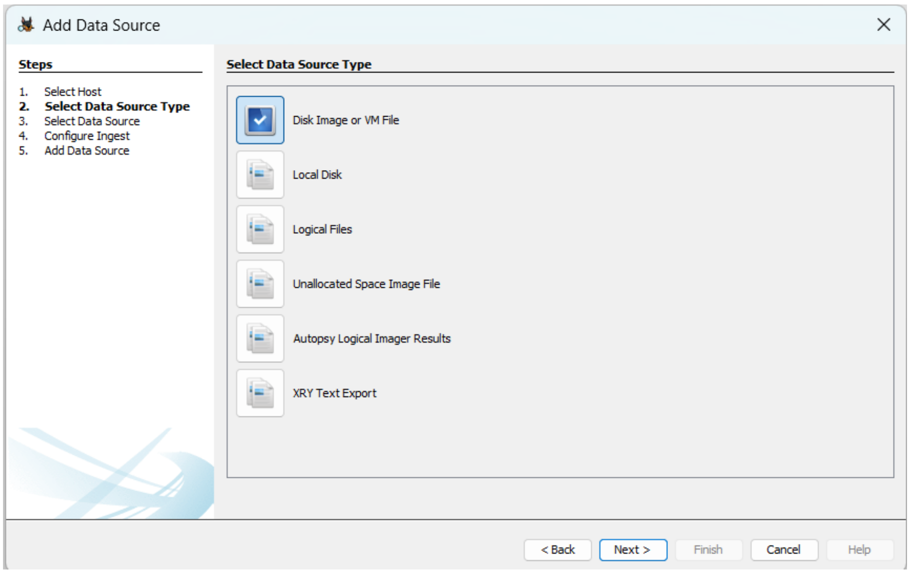
○ Use existing host

Add Data Source

We need to select the host for the data source. There are three options:

- **Generate new host based on data source name** - this will typically create a host with a name similar to our data source with the ID used in the database appended for uniqueness.
- **Specify new host name** - this will allow us to create a new host with a custom name.
- **Use existing host** - this will allow us to select an existing host from the database.

After selecting the host, we need to select the data source type.



☑ Disk Image or VM File

Local Disk

Logical Files

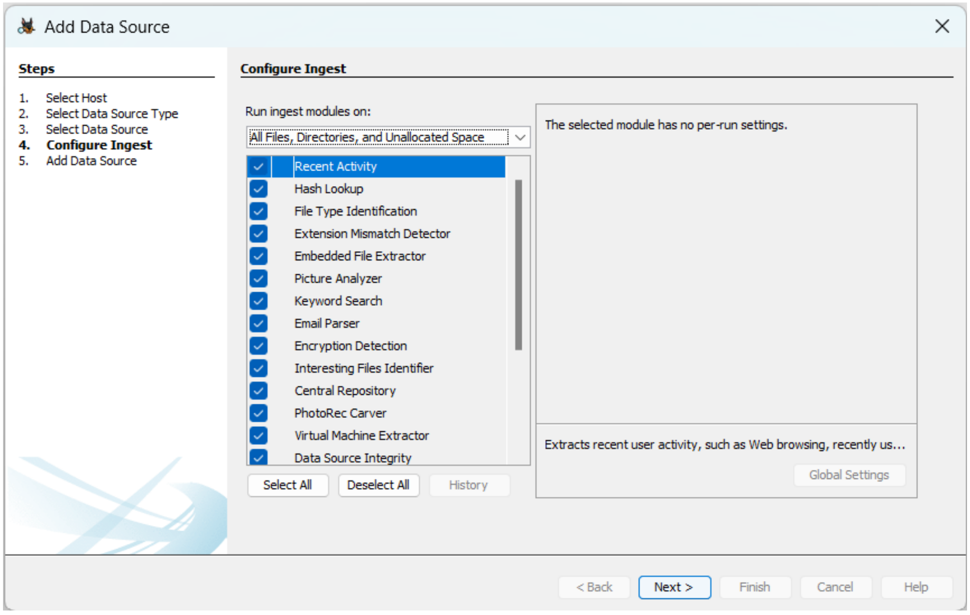
Unallocated Space Image File

Autopsy Logical Imager Results

XRY Text Export

Select Data Source Type

Next we will be prompted to select the data source file. After selecting the data source file, we will be prompted to configure the Ingest Modules:



Add Data Source

Steps

1. Select Host
2. Select Data Source Type
3. Select Data Source
4. **Configure Ingest**
5. Add Data Source

Configure Ingest

Run ingest modules on: All Files, Directories, and Unallocated Space

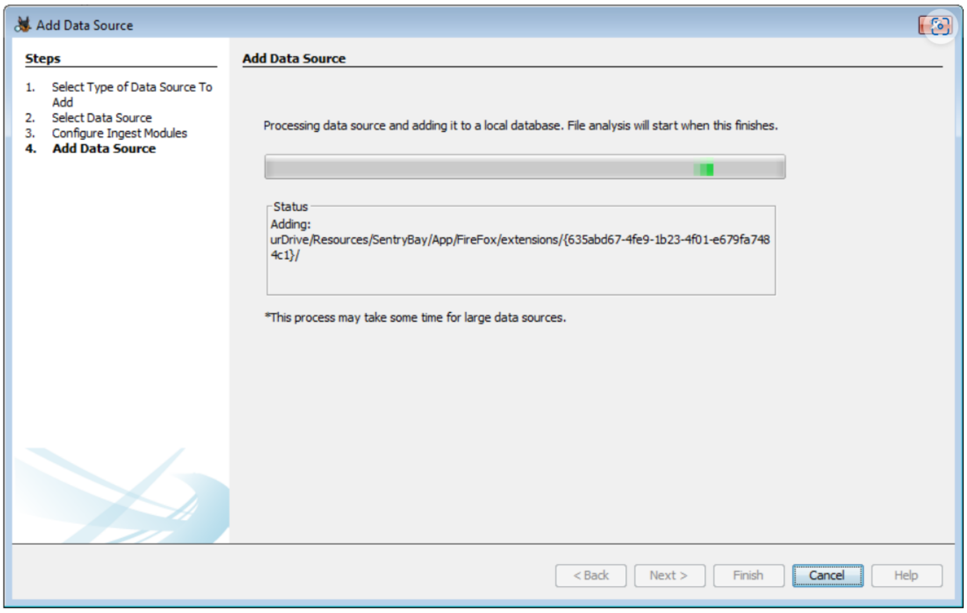
- ☑ Recent Activity
- ☑ Hash Lookup
- ☑ File Type Identification
- ☑ Extension Mismatch Detector
- ☑ Embedded File Extractor
- ☑ Picture Analyzer
- ☑ Keyword Search
- ☑ Email Parser
- ☑ Encryption Detection
- ☑ Interesting Files Identifier
- ☑ Central Repository
- ☑ PhotoRec Carver
- ☑ Virtual Machine Extractor
- ☑ Data Source Integrity

The selected module has no per-run settings.

Extracts recent user activity, such as Web browsing, recently us...

Configure Ingest Modules

Then we need to wait while Autopsy performs a basic examination of the data source and populates an embedded database with an entry for each file in the data source.



Add Data Source

Steps

1. Select Type of Data Source To Add
2. Select Data Source
3. Configure Ingest Modules
4. **Add Data Source**

Add Data Source

Processing data source and adding it to a local database. File analysis will start when this finishes.

Status
Adding:
urDrive/Resources/SentryBay/App/FireFox/extensions/{635abd67-4fe9-1b23-4f01-e679fa748
4c1}/

*This process may take some time for large data sources.

After the basic examination of the data source is complete, the ingest modules will likely still be running but we can start browsing through the files in our data source.

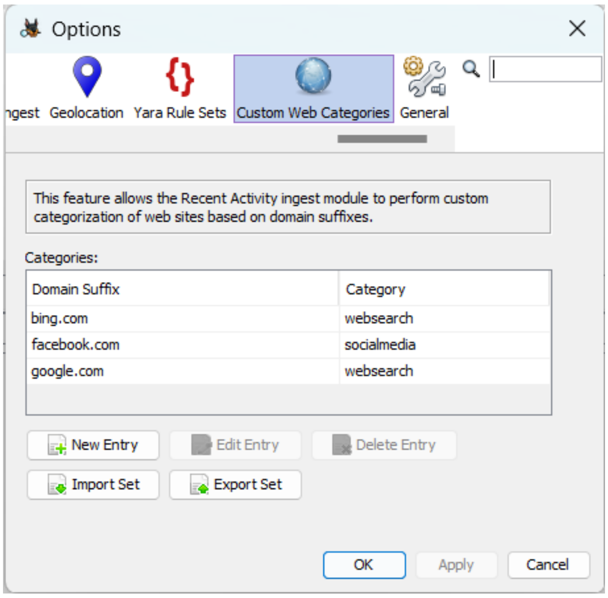
3 Analysis with Ingest Modules

3.1 Recent Activity

The Recent Activity module retrieves user activity from web browsers (including online searches), installed programmes, and the operating system. On the Registry hive, it also launches Regripper.
With this, we can view information on the past seven days of usage, including websites visited, actions taken by the device, and connections made.

Configure

This ingest module comes with some pre-defined settings, from which the result are generated. We can also add our own custom settings for web domains by going to **Tools** → **Options** → **Custom Web Catagories**.

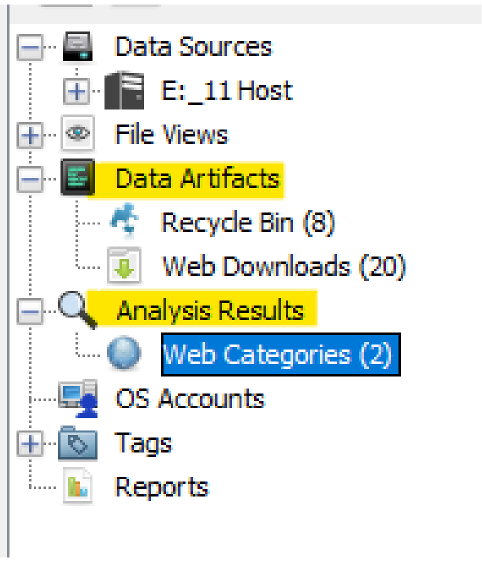


This feature allows the Recent Activity ingest module to perform custom categorization of web sites based on domain suffixes.

Categories:

Domain Suffix	Category
bing.com	websearch
facebook.com	socialmedia
google.com	websearch

Recent Activity Settings



Results of Recent Activity

Results

Results of default settings are shown in **Data Artifacts**
Results of custom settings are shown in **Analysis Results**

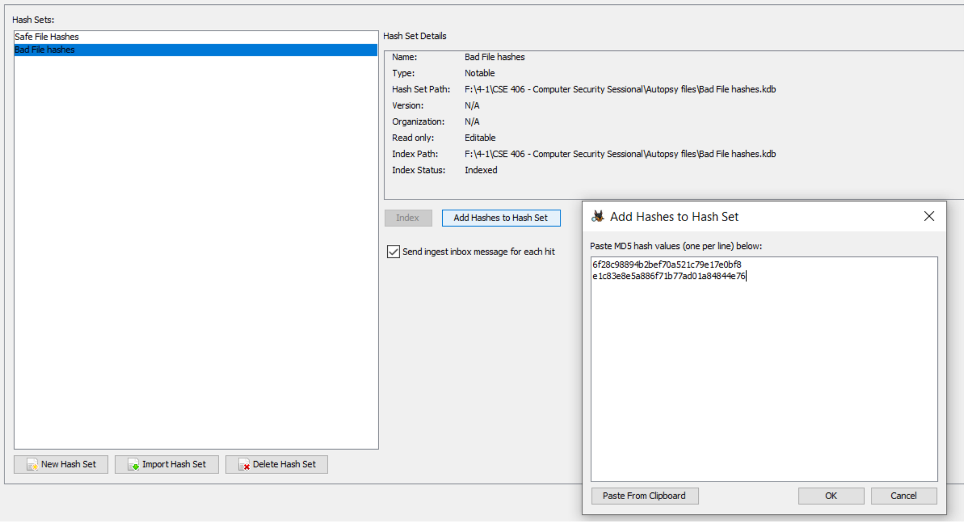
3.2 Hash Lookup

The Hash Lookup Module calculates MD5 hash values for files and looks up hash values in a database to determine if the file is notable, known (in general), or unknown.

The Hash Sets tab allows users to maintain a list of known good files. These are files that are verified to be safe and can be ignored during routine file analysis or scanning processes. By including known good files in a hash set, the software or system can skip checking them repeatedly, saving valuable time and computing resources. In addition to known good files, the Hash Sets can also be used to manage notable or known bad files. These are files that have been identified as malicious, suspicious, or potentially harmful. Including them in a hash set ensures that they are flagged and examined during analysis, helping to identify security threats or criminal activity.

Configuration

In **Tools** → **Options** → **Hash Sets** dialog box, we can add known or notable hash sets. We can add known hash sets like NSRL Reference Data Set (RDS) can be used to identify "known" files on disk image and saves a lot of time and resource in analysis by skipping these files. We can also configure known or notable hash sets using our own dataset.



Hash Sets:

Safe File Hashes
Bad File hashes

Hash Set Details

Name:	Bad File hashes
Type:	Notable
Hash Set Path:	F:\4-1\CSE 406 - Computer Security Sessional\Autopsy files\Bad File hashes.kdb
Version:	N/A
Organization:	N/A
Read only:	Editable
Index Path:	F:\4-1\CSE 406 - Computer Security Sessional\Autopsy files\Bad File hashes.kdb
Index Status:	Indexed

☑ Send ingest inbox message for each hit

Add Hashes to Hash Set

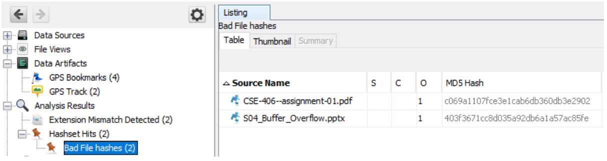
Paste MD5 hash values (one per line) below:

```
6f28c98894b2bef70a521c79e17e0bf8
e1c83e8e5a886f71b77ad01a84844e76
```

Hash Set Custom Configuration

Results

Results are shown in **Analysis Results** → **Hashset Hits**, we can see that two hits detected for files that has hash values matching the hash values in "Bad File hashes"



Bad File hashes

Source Name	S	C	O	MD5 Hash
CSE-406--assignment-01.pdf			1	c069a1107fce3e1cab6db360db3e2902
S04_Buffer_Overflow.pptx			1	403f3671cc8d035a92db6a1a57ac85fe

Results of Hash Lookup ingest module

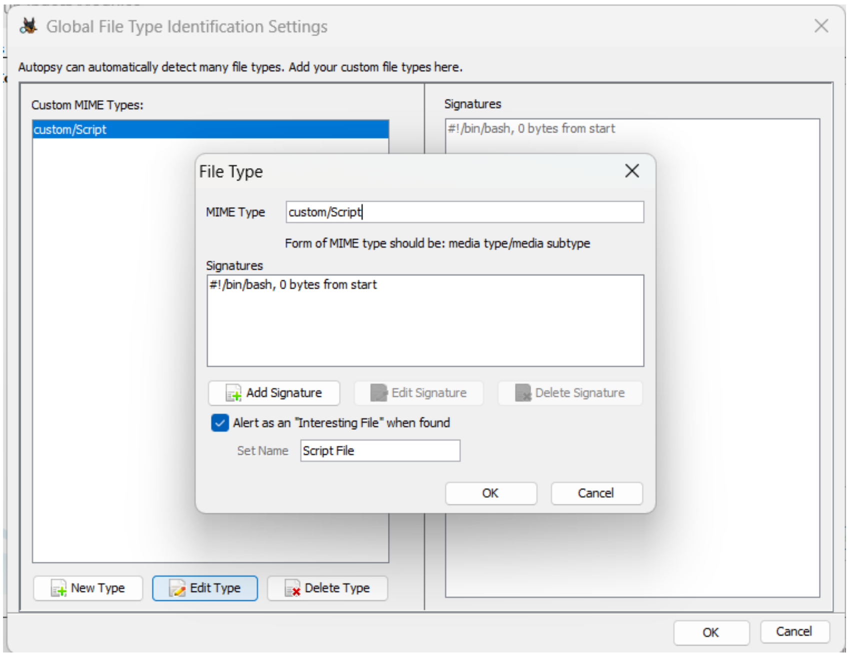
3.3 File Type Identification

The File Type ID module identifies files based on their intrinsic signatures rather than file extensions. Autopsy detects primary file IDs using the Tika library, which can be customised with user-defined criteria.

It is one of the most important modules in Autopsy as it identifies the file types of the files in the data source. It is also used by other modules such as **Extension Mismatch Detector Module** or **Keyword Search Module** to identify the file types of the files in the data source.

Custom Configuration

In global file type identification settings, we can configure custom file type of our interst .



Custom MIME Types: custom/Script

Signatures: #!/bin/bash, 0 bytes from start

MIME Type: custom/Script

Form of MIME type should be: media type/media subtype

Signatures: #!/bin/bash, 0 bytes from start

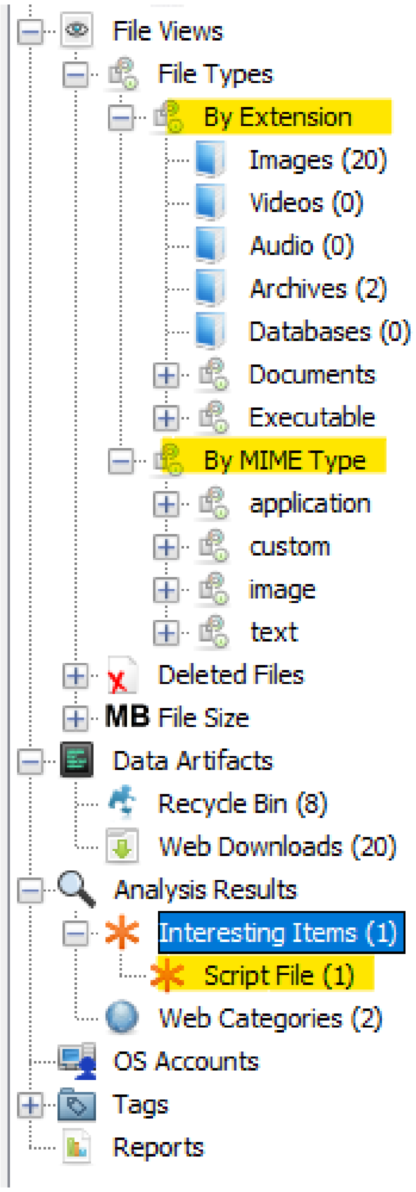
☑ Alert as an "Interesting File" when found

Set Name: Script File

Global File Type Identification Settings

Results

The results of the File Type Identification module are shown in **File Types** tab. The results are shown in two different groups as **By Extension** and **By Mime Type**. By Extension groups all files with the same extension together (may contains wrong files for currupted extension) and By Mime Type groups all files with the same mime type together.
Results of custom settings are shown grouped as given mime name in **Interesting Items** tab .



Custom File Type Identification Settings

3.4 Embedded File Extractor

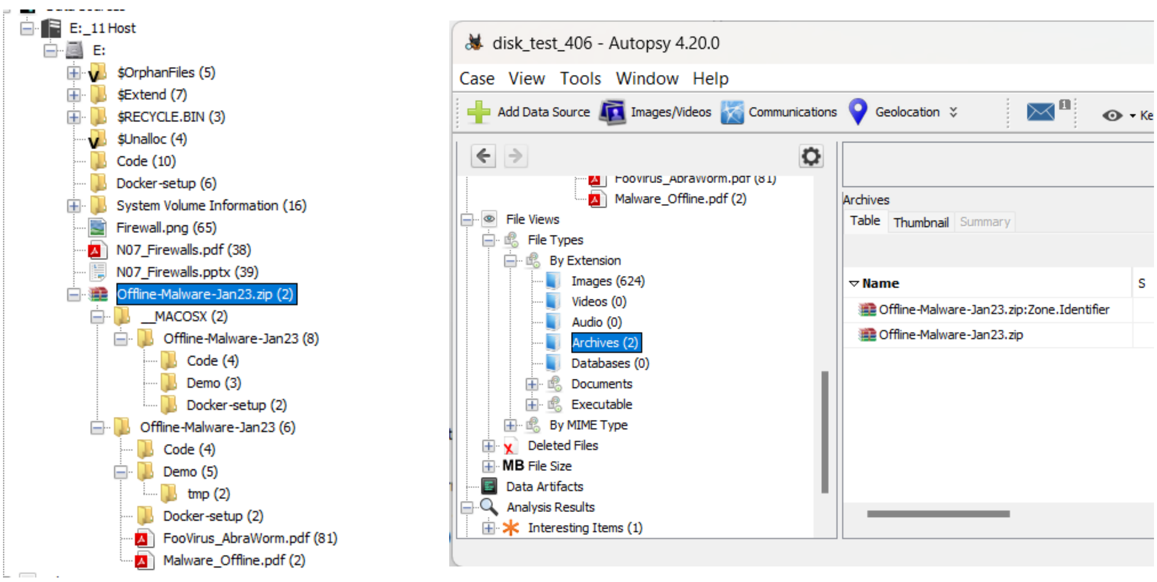
The Embedded File Extractor module opens ZIP, RAR, other archive formats, Doc, Docx, PPT, PPTX, XLS, and XLSX and sends the derived files from those files back through the ingest pipeline for analysis.

This module expands archive files to enable Autopsy to analyze all files on the system. It enables keyword search and hash lookup to analyze files inside of archives

Certain media content embedded inside Doc, Docx, PPT, PPTX, XLS, and XLSX might not be extracted.

Results

Each file extracted shows up in the data source tree view as a child of the archive containing it and as an archive under "Views", "File Types", "Archives".



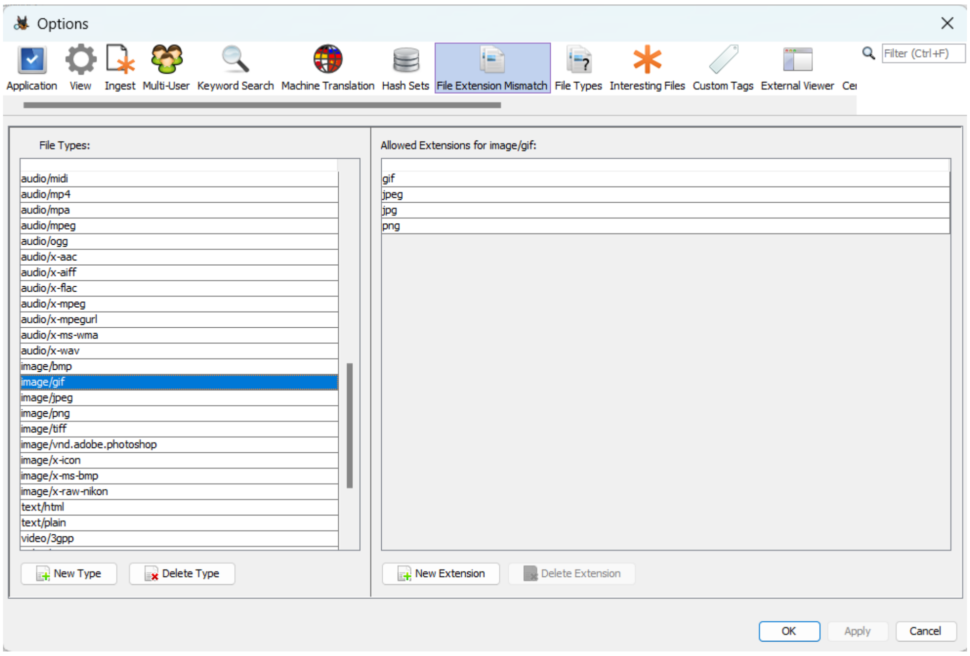
Embedded File Extractor Results

3.5 Extension Mismatch Detector

Extension Mismatch Detector module uses the results from the File Type Identification and flags files that have an extension not traditionally associated with the file's detected type.

Configuration

In **Tools → Options → File Extension Mismatch** dialog box, we can add and remove MIME types and as well as extensions.



File Types:

audio/midi
audio/mp4
audio/mpa
audio/mpeg
audio/ogg
audio/x-aac
audio/x-aiff
audio/x-flac
audio/x-mpeg
audio/x-mpegurl
audio/x-ms-wma
audio/x-wav
image/bmp
image/gif
image/jpeg
image/png
image/tiff
image/vnd.adobe.photoshop
image/x-icon
image/x-ms-bmp
image/x-raw-nikon
text/html
text/plain
video/3gpp

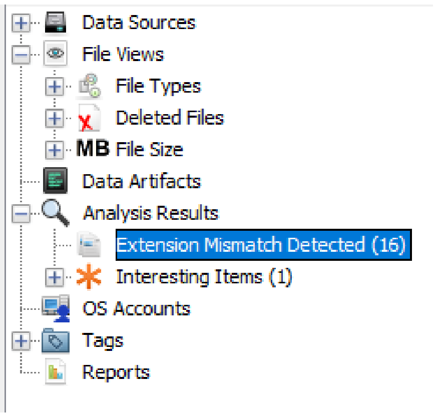
Allowed Extensions for image/gif:

gif
jpeg
jpg
png

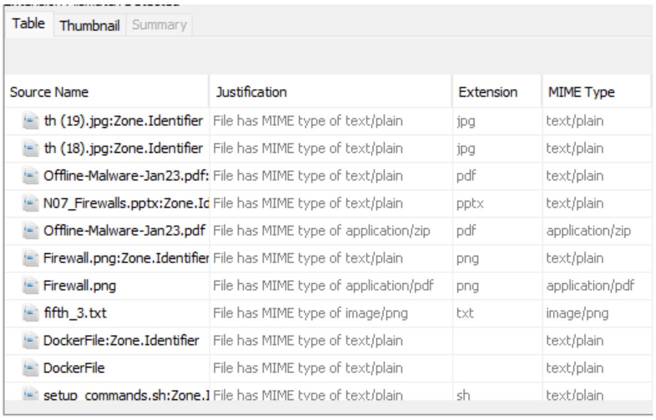
Extension Mismatch Detector Settings

Results

Results are shown in **Analysis Results → Extension Mismatch Detected**



Results



Source Name	Justification	Extension	MIME Type
th (19).jpg:Zone.Identifier	File has MIME type of text/plain	jpg	text/plain
th (18).jpg:Zone.Identifier	File has MIME type of text/plain	jpg	text/plain
Offline-Malware-Jan23.pdf:	File has MIME type of text/plain	pdf	text/plain
N07_Firewalls.pptx:Zone.Ic	File has MIME type of text/plain	pptx	text/plain
Offline-Malware-Jan23.pdf	File has MIME type of application/zip	pdf	application/zip
Firewall.png:Zone.Identifier	File has MIME type of text/plain	png	text/plain
Firewall.png	File has MIME type of application/pdf	png	application/pdf
fifth_3.txt	File has MIME type of image/png	txt	image/png
DockerFile:Zone.Identifier	File has MIME type of text/plain		text/plain
DockerFile	File has MIME type of text/plain		text/plain
setup_commands.sh:Zone.I	File has MIME type of text/plain	sh	text/plain

Details of Mismatched Files

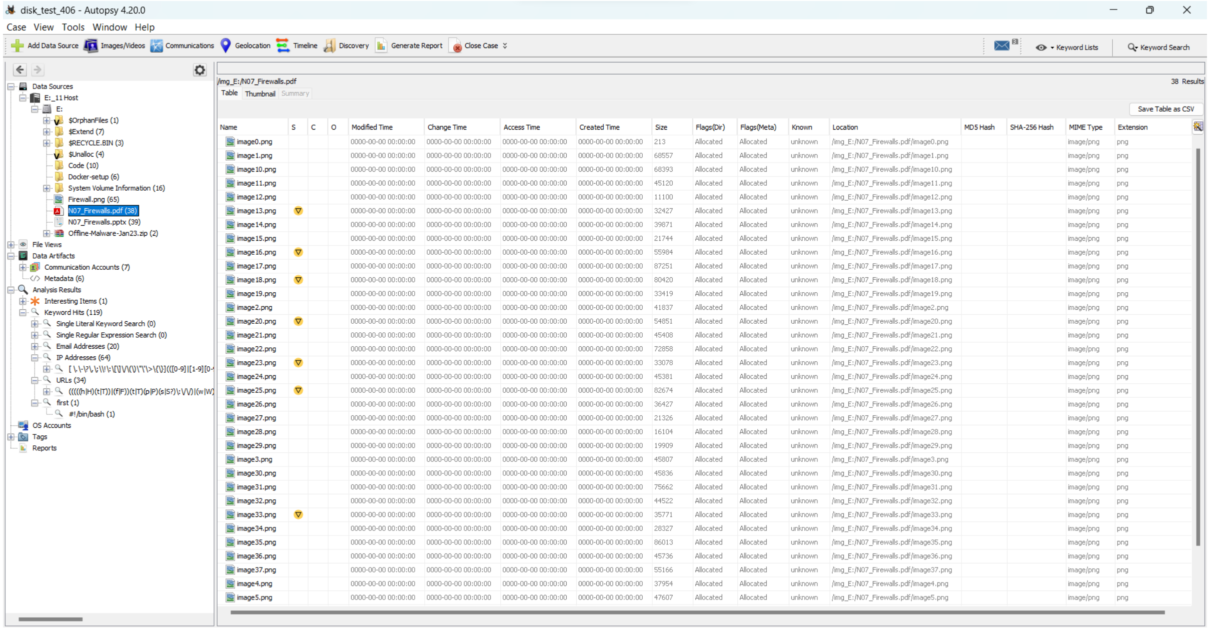
3.6 Picture Analyzer

The Picture Analyzer module retrieves EXIF (Exchangeable Image File Format) metadata from ingested images. The picture's geolocation information, as well as the time, date, camera model, and settings (exposure times, resolution, etc.), may be included in this data. The Blackboard is updated with the newly found properties. This can include information on the photograph's location, timing, and camera used.

Additionally, the module transforms HEIC/HEIF photos to JPG while preserving their EXIF data, which will be handled and saved similarly to regular JPG photographs.

Results

The results are shown in the Result trees.



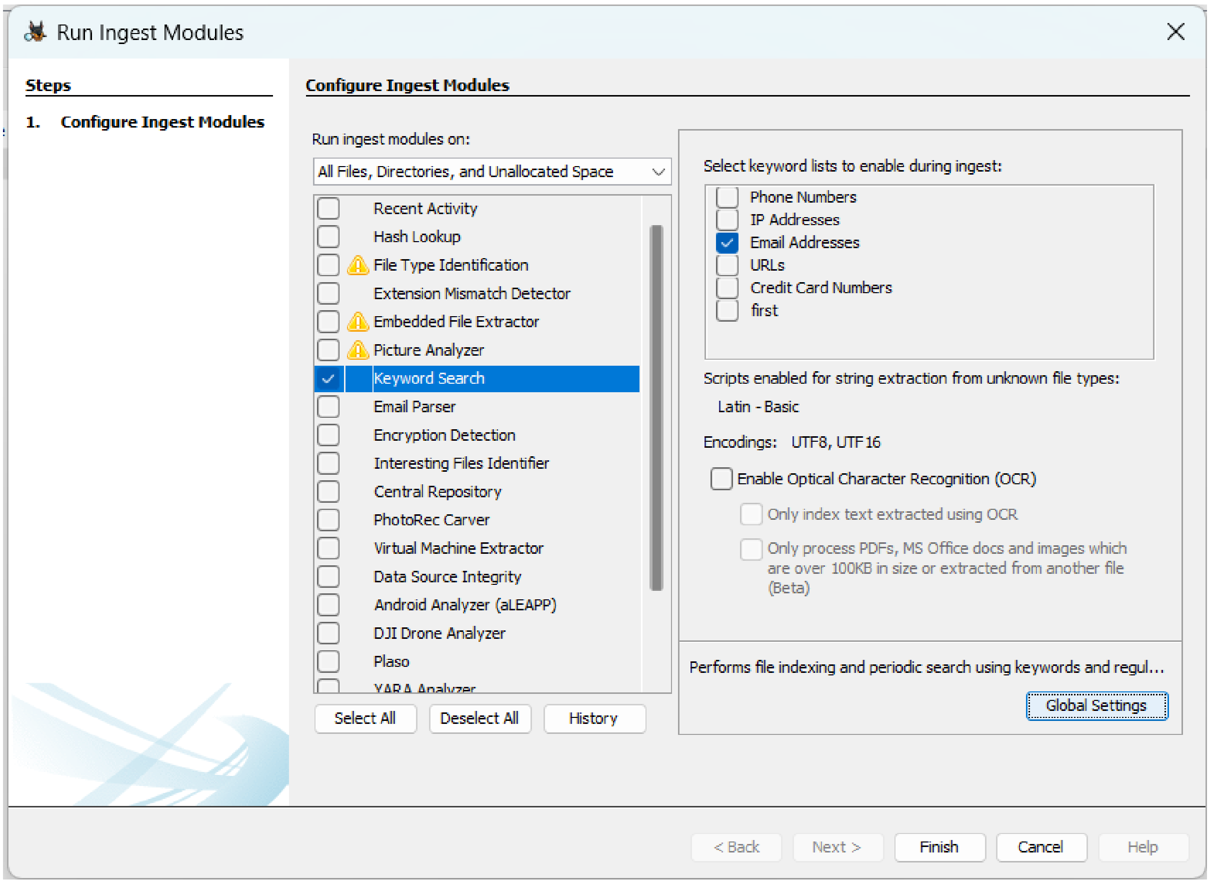
Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location	MD5 Hash	SHA-256 Hash	MIME Type	Extension
image0.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	213	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image0.png			image/png	png
image1.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	68557	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image1.png			image/png	png
image10.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	68393	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image10.png			image/png	png
image11.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45120	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image11.png			image/png	png
image12.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	11100	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image12.png			image/png	png
image13.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	32427	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image13.png			image/png	png
image14.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	39871	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image14.png			image/png	png
image15.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	21744	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image15.png			image/png	png
image16.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	55984	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image16.png			image/png	png
image17.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	87251	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image17.png			image/png	png
image18.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	80420	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image18.png			image/png	png
image19.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	33419	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image19.png			image/png	png
image2.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	41837	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image2.png			image/png	png
image20.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	54851	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image20.png			image/png	png
image21.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45408	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image21.png			image/png	png
image22.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	72858	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image22.png			image/png	png
image23.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	33078	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image23.png			image/png	png
image24.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45381	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image24.png			image/png	png
image25.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	82674	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image25.png			image/png	png
image26.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	36427	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image26.png			image/png	png
image27.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	21326	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image27.png			image/png	png
image28.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16104	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image28.png			image/png	png
image29.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	19909	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image29.png			image/png	png
image3.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45807	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image3.png			image/png	png
image30.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45836	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image30.png			image/png	png
image31.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	75662	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image31.png			image/png	png
image32.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	44522	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image32.png			image/png	png
image33.png	▽			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	35771	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image33.png			image/png	png
image34.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	28327	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image34.png			image/png	png
image35.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	86013	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image35.png			image/png	png
image36.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	45736	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image36.png			image/png	png
image37.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	55166	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image37.png			image/png	png
image4.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	37954	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image4.png			image/png	png
image5.png				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	47607	Allocated	Allocated	unknown	/img_E:/N07_Firewalls.pdf/image5.png			image/png	png

Picture Analyzer Results

An interesting test case of this module can be found in the test cases section 6.4 of this document.

3.7 Keyword Search

The Keyword Search module facilitates both the ingest portion of searching and also supports manual text searching after ingest has completed (see Ad Hoc Keyword Search). It extracts text from files being ingested, selected reports generated by other modules, and results generated by other modules.
When selecting this modules we can add keywords to search for. We can also add regular expressions to search for. We can also add keywords to exclude from the search.



Run Ingest Modules

Steps

1. **Configure Ingest Modules**

Configure Ingest Modules

Run ingest modules on: All Files, Directories, and Unallocated Space

- □ Recent Activity
- □ Hash Lookup
- □ File Type Identification
- □ Extension Mismatch Detector
- □ Embedded File Extractor
- □ Picture Analyzer
- ☑ Keyword Search
- □ Email Parser
- □ Encryption Detection
- □ Interesting Files Identifier
- □ Central Repository
- □ PhotoRec Carver
- □ Virtual Machine Extractor
- □ Data Source Integrity
- □ Android Analyzer (aLEAPP)
- □ DJI Drone Analyzer
- □ Plaso
- □ YARA Analyzer

Select keyword lists to enable during ingest:

- □ Phone Numbers
- □ IP Addresses
- ☑ Email Addresses
- □ URLs
- □ Credit Card Numbers
- □ first

Scripts enabled for string extraction from unknown file types:

Latin - Basic

Encodings: UTF8, UTF16

□ Enable Optical Character Recognition (OCR)

□ Only index text extracted using OCR

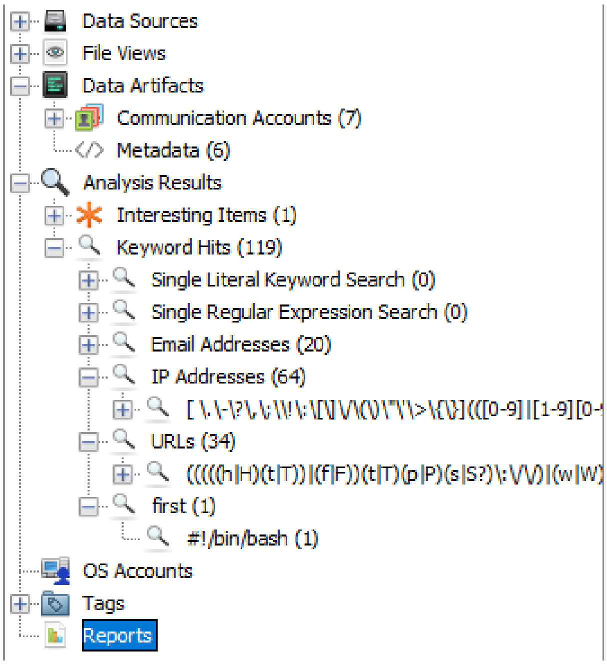
□ Only process PDFs, MS Office docs and images which are over 100KB in size or extracted from another file (Beta)

Performs file indexing and periodic search using keywords and regul...

Keyword Search Settings

Results

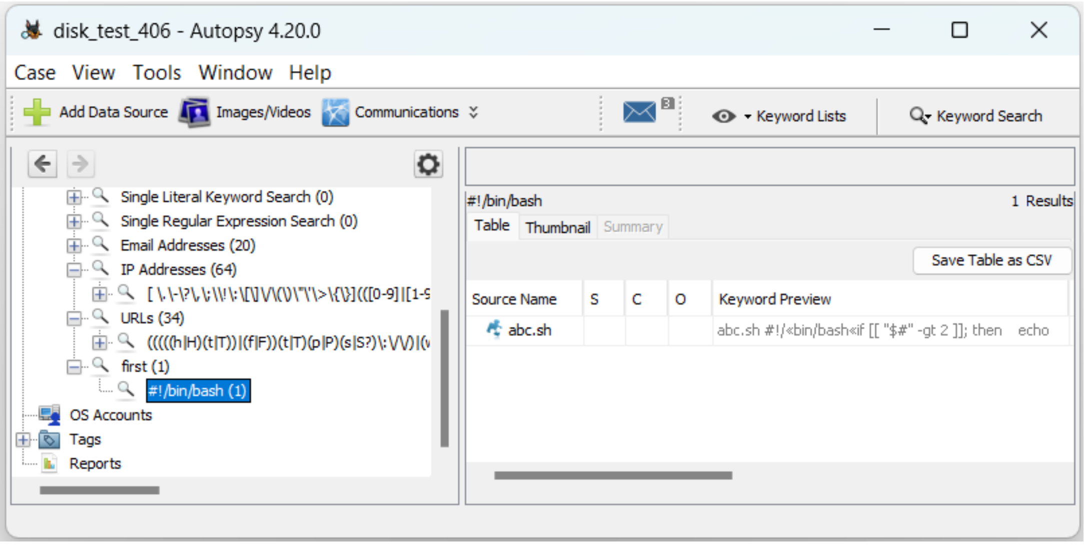
Also in the global setting options we can add our own keyword rule to search for. Running this module will search for the keywords in the files as well as in the file metadata. The results are shown in **Analysis Results** → **Keyword Hits**. Results will be grouped together by the keyword that was found.



```
Data Sources
File Views
Data Artifacts
    Communication Accounts (7)
    Metadata (6)
Analysis Results
    Interesting Items (1)
    Keyword Hits (119)
        Single Literal Keyword Search (0)
        Single Regular Expression Search (0)
        Email Addresses (20)
        IP Addresses (64)
            [ \.\-\?\,\;\!\:\[\]\/\(\)\"\'\>\{\}](([0-9]|[1-9][0-
        URLs (34)
            (((((h|H)(t|T))|(f|F))(t|T)(p|P)(s|S?)\:\/\/)|(w|W)
        first (1)
            #!/bin/bash (1)
OS Accounts
Tags
Reports
```

Keyword Search Results Tabs

Selecting a keyword will show the details of the files that contain the keyword.



disk_test_406 - Autopsy 4.20.0

```
Single Literal Keyword Search (0)
Single Regular Expression Search (0)
Email Addresses (20)
IP Addresses (64)
    [ \.\-\?\,\;\!\:\[\]\/\(\)\"\'\>\{\}](([0-9]|[1-9
URLs (34)
    (((((h|H)(t|T))|(f|F))(t|T)(p|P)(s|S?)\:\/\/)|(
first (1)
    #!/bin/bash (1)
OS Accounts
Tags
Reports
```

#!/bin/bash 1 Results

Source Name	S	C	O	Keyword Preview
abc.sh				abc.sh #!/«bin/bash«if [["$#" -gt 2]]; then echo

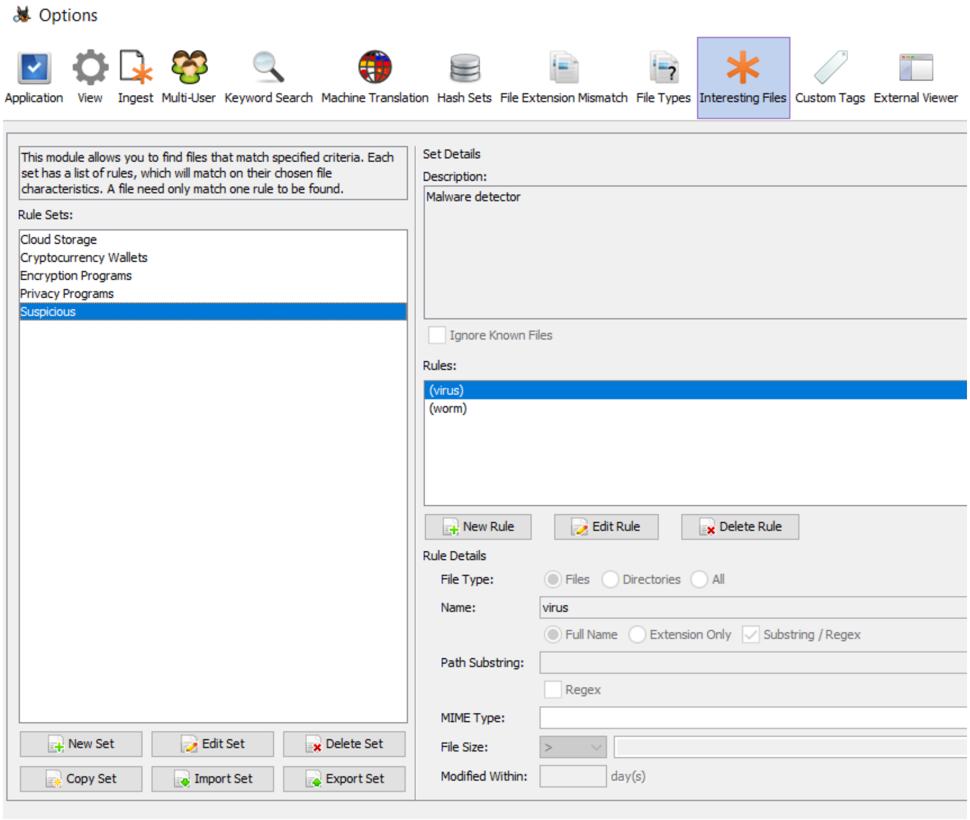
Keyword Search Results Details

3.8 Interesting Files Identifier

The Interesting Files identifier module allows files and directories that follows a set of rules to be automatically marked. This can be useful when there is a need to check whether files with a particular name or path exist in the data source, or when files of a specific type are always of interest.

Configuration

Users can define a set of rules by accessing the **Tools** → **Options** → **Interesting Files** dialog box. For instance, in our demonstration, we established a rule set named "Suspicious." Within this set, we incorporated two specific rules. These rules dictate that if any file contains the substring "Worm" or "virus" in its name, it will be categorized as an "interesting file."



This module allows you to find files that match specified criteria. Each set has a list of rules, which will match on their chosen file characteristics. A file need only match one rule to be found.

Rule Sets:

Cloud Storage
Cryptocurrency Wallets
Encryption Programs
Privacy Programs
Suspicious

Set Details

Description:

Malware detector

□ Ignore Known Files

Rules:

(virus)
(worm)

Rule Details

File Type: ● Files ○ Directories ○ All

Name: virus

● Full Name ○ Extension Only ☑ Substring / Regex

Path Substring:

□ Regex

MIME Type:

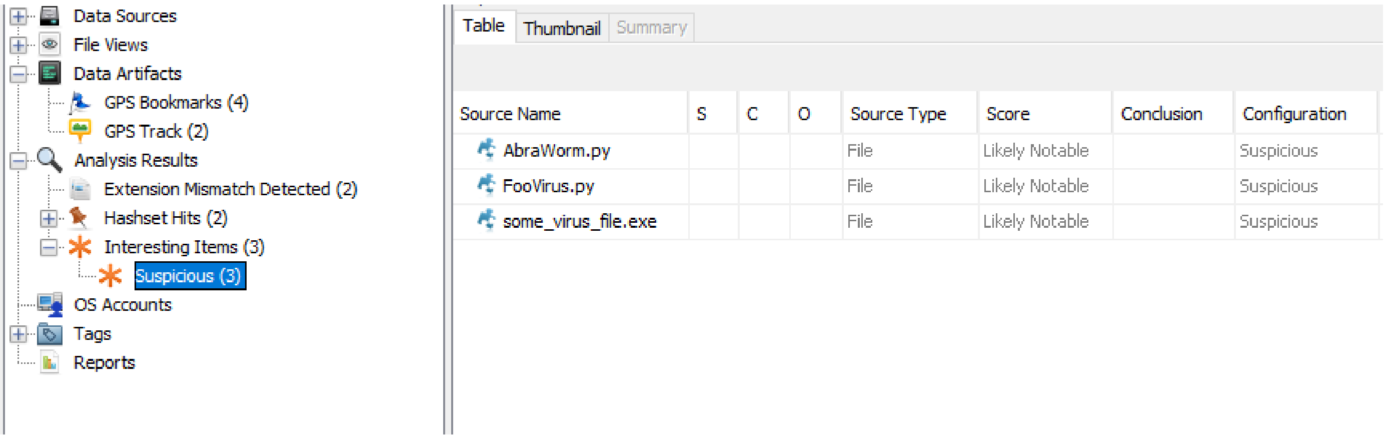
File Size: >

Modified Within: day(s)

Interesting Files Configuration

Results

After running the ”Interesting Files Identifier” ingest module we can see the results in **Analysis Results** → **Interesting Items**, 3 files has been identified as interesting items



Source Name	S	C	O	Source Type	Score	Conclusion	Configuration
AbraWorm.py				File	Likely Notable		Suspicious
FooVirus.py				File	Likely Notable		Suspicious
some_virus_file.exe				File	Likely Notable		Suspicious

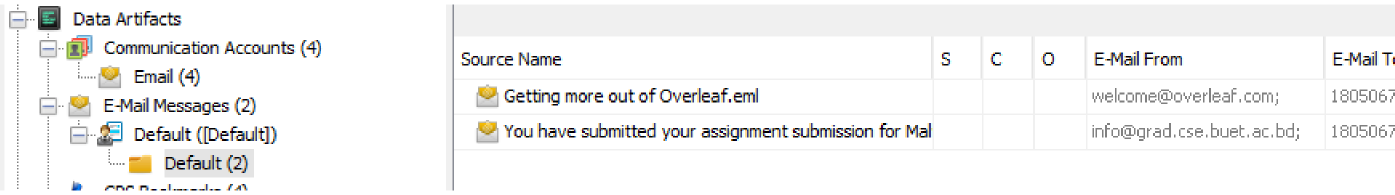
Results of Interesting Files Identifier ingest module

due to their names containing the substrings ”worm” or ”virus,” as defined within the ”Suspicious” set of rules.

3.9 Email Parser

The Email Parser module identifies MBOX, EML and PST format files based on file signatures, extracting the e-mails from them, adding the results to the Blackboard. It also identifies the email accounts used in the discovered emails

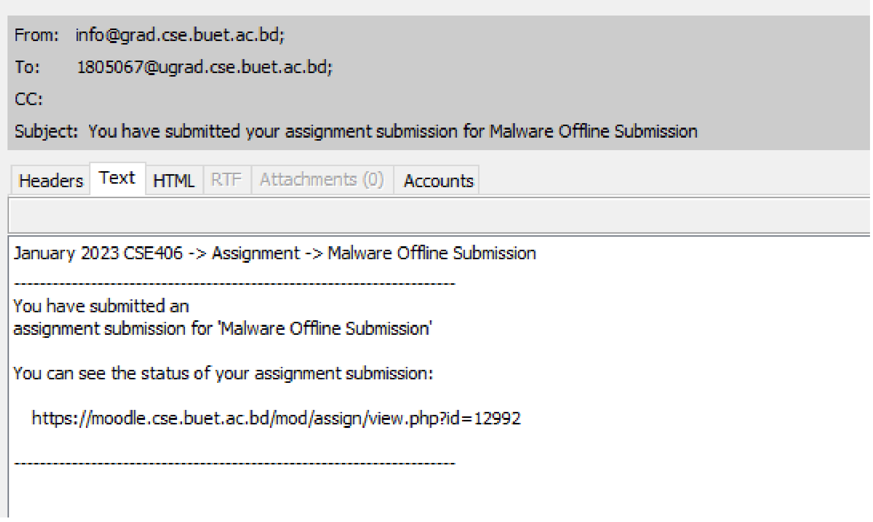
Results



Source Name	S	C	O	E-Mail From	E-Mail T
Getting more out of Overleaf.eml				welcome@overleaf.com;	1805067
You have submitted your assignment submission for Mal				info@grad.cse.buet.ac.bd;	1805067

Results of Email Parser ingest module

After running the "Email Parser" ingest module we can see the results in **Data Artifacts** → **E-mail messages**, 2 email files has been identified by the module.



From: info@grad.cse.buet.ac.bd;
To: 1805067@ugrad.cse.buet.ac.bd;
CC:
Subject: You have submitted your assignment submission for Malware Offline Submission

January 2023 CSE406 -> Assignment -> Malware Offline Submission

----

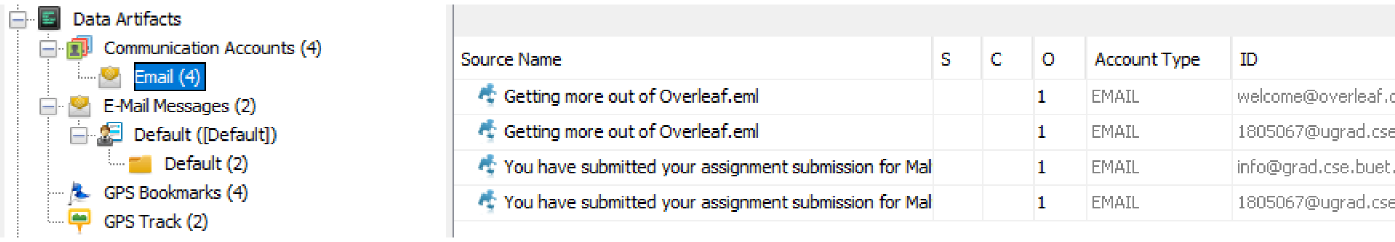
You have submitted an
assignment submission for 'Malware Offline Submission'

You can see the status of your assignment submission:

https://moodle.cse.buet.ac.bd/mod/assign/view.php?id=12992

Details of an email

We can also see the details of a selected email in the bottom part of the result.



Source Name	S	C	O	Account Type	ID
Getting more out of Overleaf.eml			1	EMAIL	welcome@overleaf.c
Getting more out of Overleaf.eml			1	EMAIL	1805067@ugrad.cse
You have submitted your assignment submission for Mal			1	EMAIL	info@grad.cse.buet.
You have submitted your assignment submission for Mal			1	EMAIL	1805067@ugrad.cse

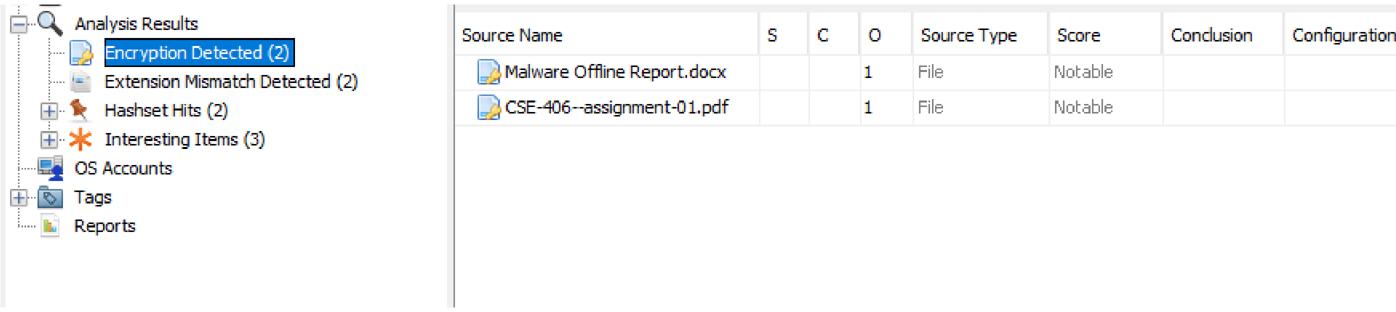
Email Accounts

From **Data Artifacts** → **Communication Accounts** → **Email** we can see the email accounts that has been used in the emails identified by the module

3.10 Encryption Detection

The Encryption Detection Module is designed to identify files that may be encrypted or protected with passwords. It accomplishes this through a combination of general entropy calculations and specialized tests tailored to specific file types.

Results



Source Name	S	C	O	Source Type	Score	Conclusion	Configuration
Malware Offline Report.docx			1	File	Notable		
CSE-406--assignment-01.pdf			1	File	Notable		

Results of Encryption Detection ingest module

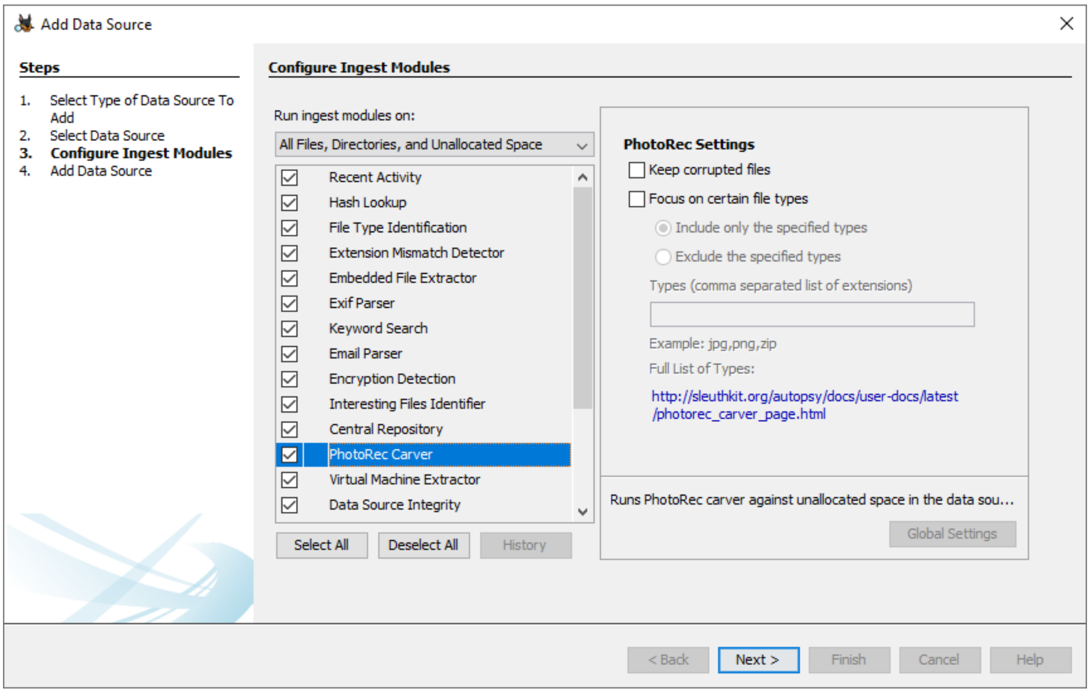
After running the "Encryption Detection" ingest module we can see the results in **Analysis Results** → **Encryption Detection**, 2 files have been identified as we need password to access those two files

3.11 PhotoRec Carver

The PhotoRec Carver module carves files from unallocated space in the data source and sends the files found through the ingest processing chain.

This can help the user to discover more information about files that used to be on the device and were subsequently deleted. These are simply extra files that were found in ”empty” portions of the device storage.

The run-time setting for this module allows user to choose whether to keep corrupted files and to include or exclude certain file types.



Add Data Source

Steps

1. Select Type of Data Source To Add
2. Select Data Source
3. **Configure Ingest Modules**
4. Add Data Source

Configure Ingest Modules

Run ingest modules on:

All Files, Directories, and Unallocated Space

- ☑ Recent Activity
- ☑ Hash Lookup
- ☑ File Type Identification
- ☑ Extension Mismatch Detector
- ☑ Embedded File Extractor
- ☑ Exif Parser
- ☑ Keyword Search
- ☑ Email Parser
- ☑ Encryption Detection
- ☑ Interesting Files Identifier
- ☑ Central Repository
- ☑ PhotoRec Carver
- ☑ Virtual Machine Extractor
- ☑ Data Source Integrity

PhotoRec Settings

☐ Keep corrupted files

☐ Focus on certain file types

● Include only the specified types

○ Exclude the specified types

Types (comma separated list of extensions)

Example: jpg,png,zip

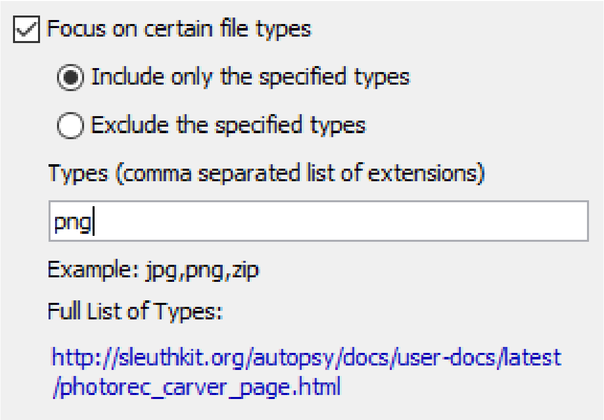
Full List of Types:

http://sleuthkit.org/autopsy/docs/user-docs/latest/photorec_carver_page.html

Runs PhotoRec carver against unallocated space in the data sou...

PhotoRec Carver Settings

For the ”Focus on certain file types” option, user can enter a comma separated list of file types. Depending on which option choosen, PhotoRec will either carve only files of those types or all files except those types.



☑ Focus on certain file types

● Include only the specified types

○ Exclude the specified types

Types (comma separated list of extensions)

png

Example: jpg,png,zip

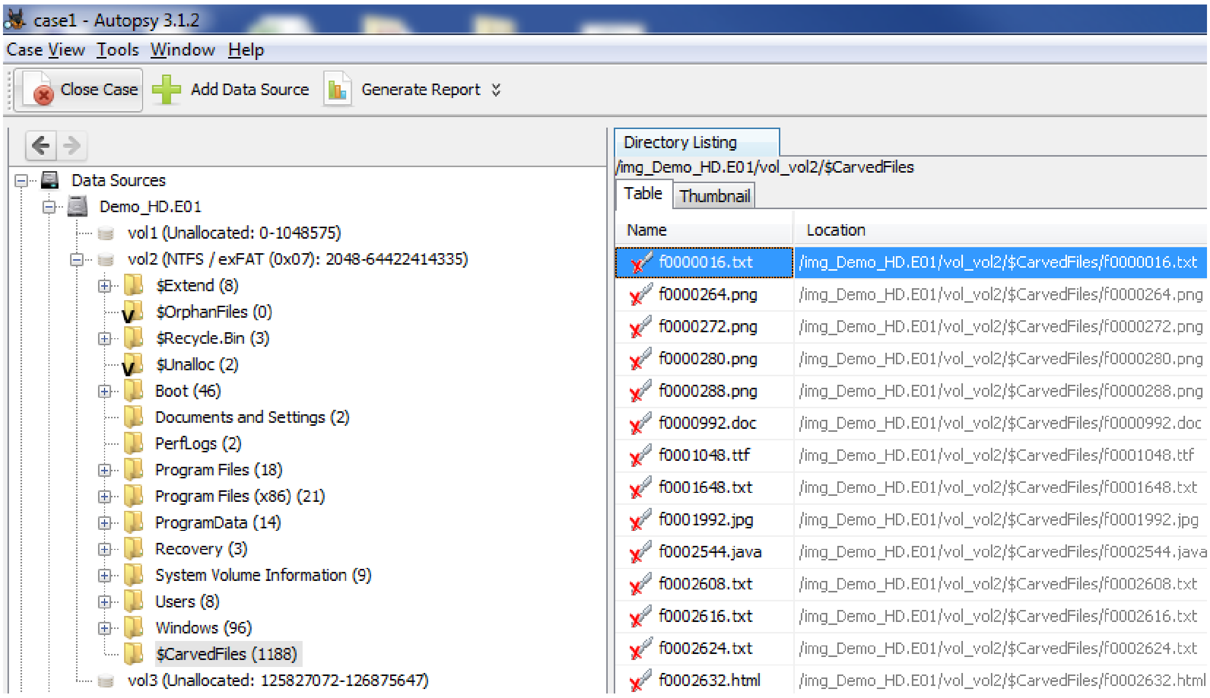
Full List of Types:

http://sleuthkit.org/autopsy/docs/user-docs/latest/photorec_carver_page.html

PhotoRec Carver Results

Results

The results of carving show up on the tree under the appropriate data source with the heading (CarvedFiles)



/img_Demo_HD.E01/vol_vol2/$CarvedFiles

- Data Sources
  - Demo_HD.E01
    - vol1 (Unallocated: 0-1048575)
    - vol2 (NTFS / exFAT (0x07): 2048-64422414335)
      - $Extend (8)
      - $OrphanFiles (0)
      - $Recycle.Bin (3)
      - $Unalloc (2)
      - Boot (46)
      - Documents and Settings (2)
      - PerfLogs (2)
      - Program Files (18)
      - Program Files (x86) (21)
      - ProgramData (14)
      - Recovery (3)
      - System Volume Information (9)
      - Users (8)
      - Windows (96)
      - $CarvedFiles (1188)
    - vol3 (Unallocated: 125827072-126875647)

Name	Location
f0000016.txt	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0000016.txt
f0000264.png	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0000264.png
f0000272.png	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0000272.png
f0000280.png	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0000280.png
f0000288.png	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0000288.png
f0000992.doc	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0000992.doc
f0001048.ttf	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0001048.ttf
f0001648.txt	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0001648.txt
f0001992.jpg	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0001992.jpg
f0002544.java	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0002544.java
f0002608.txt	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0002608.txt
f0002616.txt	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0002616.txt
f0002624.txt	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0002624.txt
f0002632.html	/img_Demo_HD.E01/vol_vol2/$CarvedFiles/f0002632.html

PhotoRec Carver Results

Recover

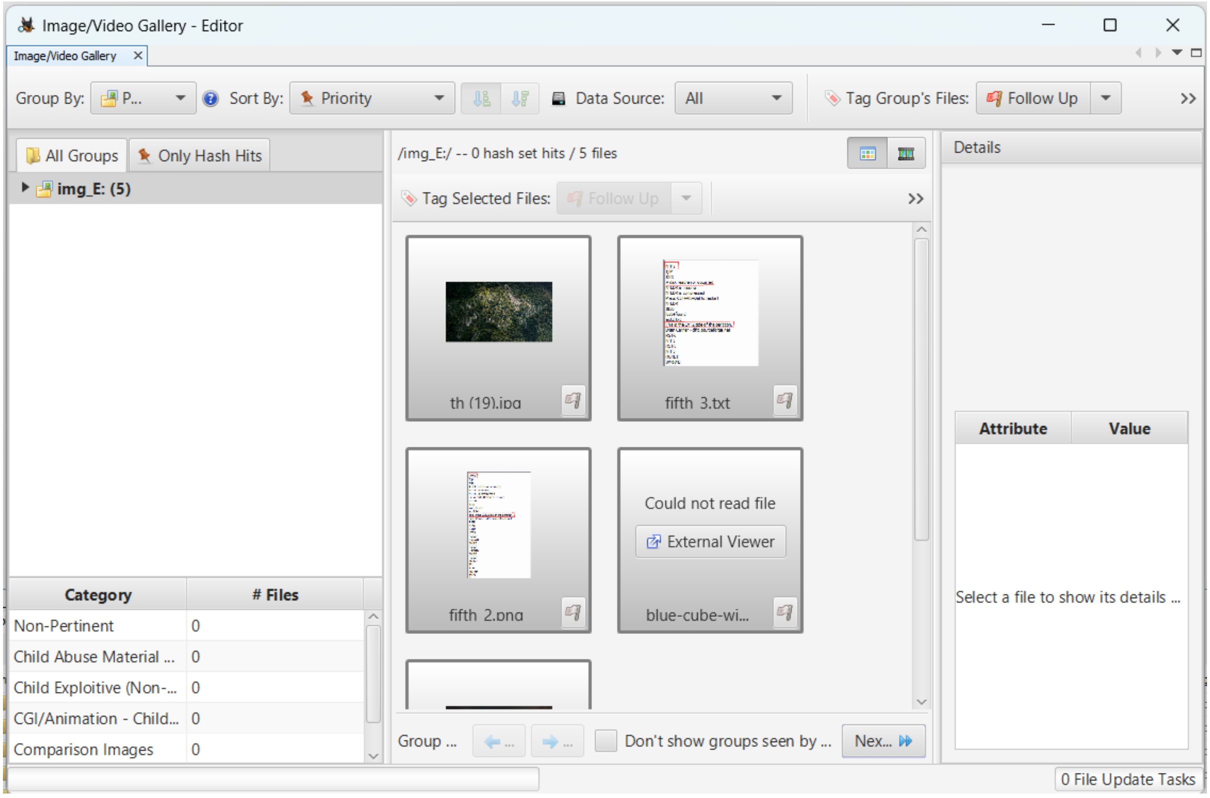
By rightclicking on selected file, we will see options for extracting the file. Extracting the file will recover the data in a given directory.

A test case for this module can be found in the test cases section 6.3 -FTS Undelete Test of this document.

4 Specialize Views

4.1 Images/Video Gallery

The Image Gallery option in Autopsy supports investigations involving images and videos. It organises photographs into folders and properties, making it easier to manage vast collections and concentrate on important content. It enables instant image viewing during ingestion, removing the need to wait for the full process to complete.



Group By: P...

Sort By: Priority

Data Source: All

Tag Group's Files: Follow Up

/img_E:/ -- 0 hash set hits / 5 files

Could not read file

Select a file to show its details ...

Category	# Files
Non-Pertinent	0
Child Abuse Material ...	0
Child Exploitive (Non-...	0
CGI/Animation - Child...	0
Comparison Images	0

0 File Update Tasks

Image Gallery

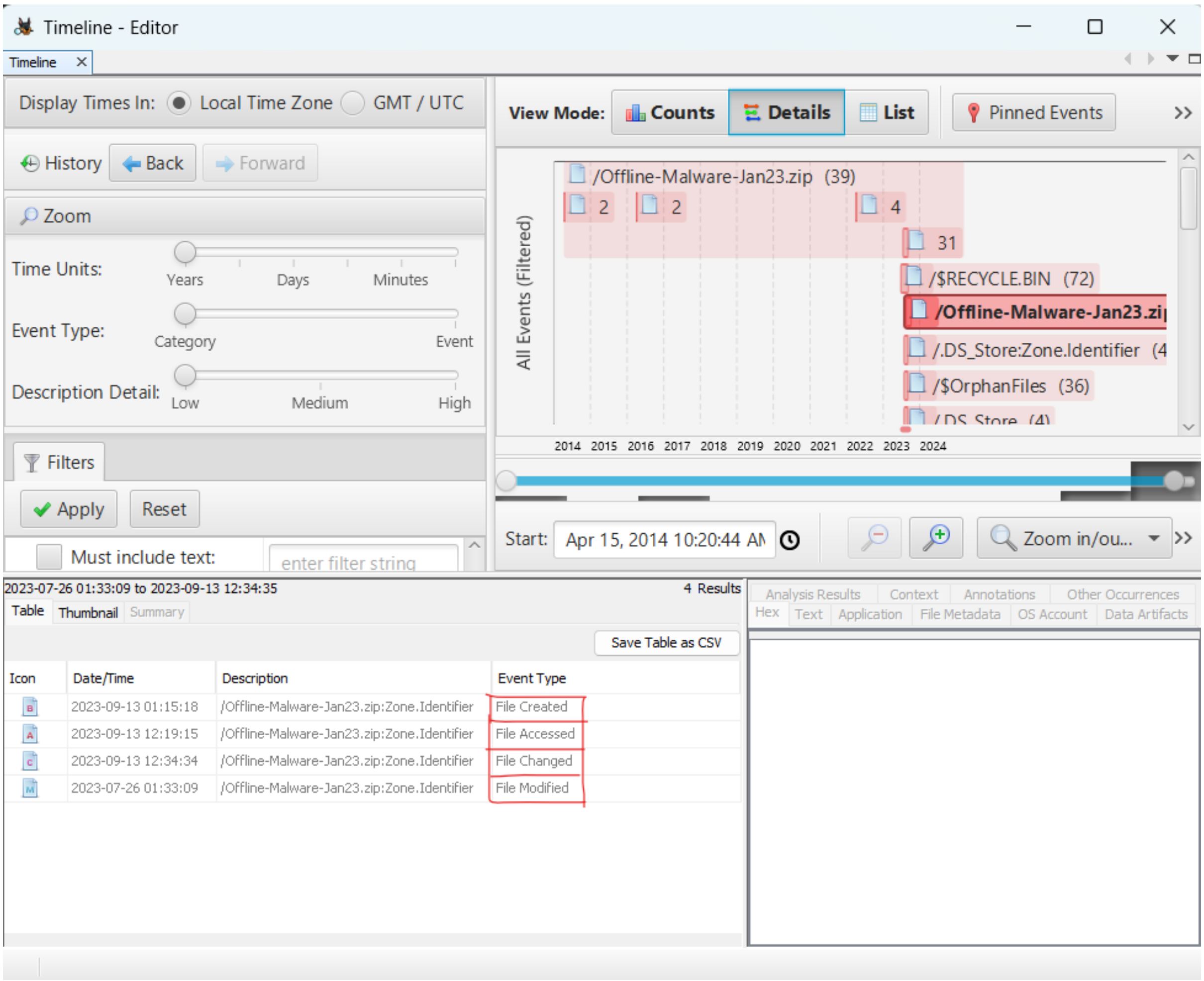
4.2 Timeline

Autopsy's Timeline is a powerful digital forensics tool that tracks and chronicles crucial events such as web activity, external device connections, EXIF photo additions, and their correlations with file system modifications.

Hash Lookup, Recent Activity and Picture Analyzer modules are required to be run before to get most out of Timeline feature.

The timeline tool is based on events. An Event has a timestamp, a category, and a description. ALthough all events are discrete, they can be grouped together manually.

The Timeline feature in Autopsy gathers and categorizes data from various sources into the some event types such as for File System Changes (Access, Creation, Modification, Deletion).



Display Times In: ● Local Time Zone ○ GMT / UTC

/Offline-Malware-Jan23.zip (39)
2
2
4
31
/$RECYCLE.BIN (72)
/Offline-Malware-Jan23.zi
/.DS_Store:Zone.Identifier (4
/$OrphanFiles (36)
/.DS_Store (4)

Start: Apr 15, 2014 10:20:44 AM

□ Must include text: enter filter string

2023-07-26 01:33:09 to 2023-09-13 12:34:35 4 Results

Icon	Date/Time	Description	Event Type
B	2023-09-13 01:15:18	/Offline-Malware-Jan23.zip:Zone.Identifier	File Created
A	2023-09-13 12:19:15	/Offline-Malware-Jan23.zip:Zone.Identifier	File Accessed
C	2023-09-13 12:34:34	/Offline-Malware-Jan23.zip:Zone.Identifier	File Changed
M	2023-07-26 01:33:09	/Offline-Malware-Jan23.zip:Zone.Identifier	File Modified

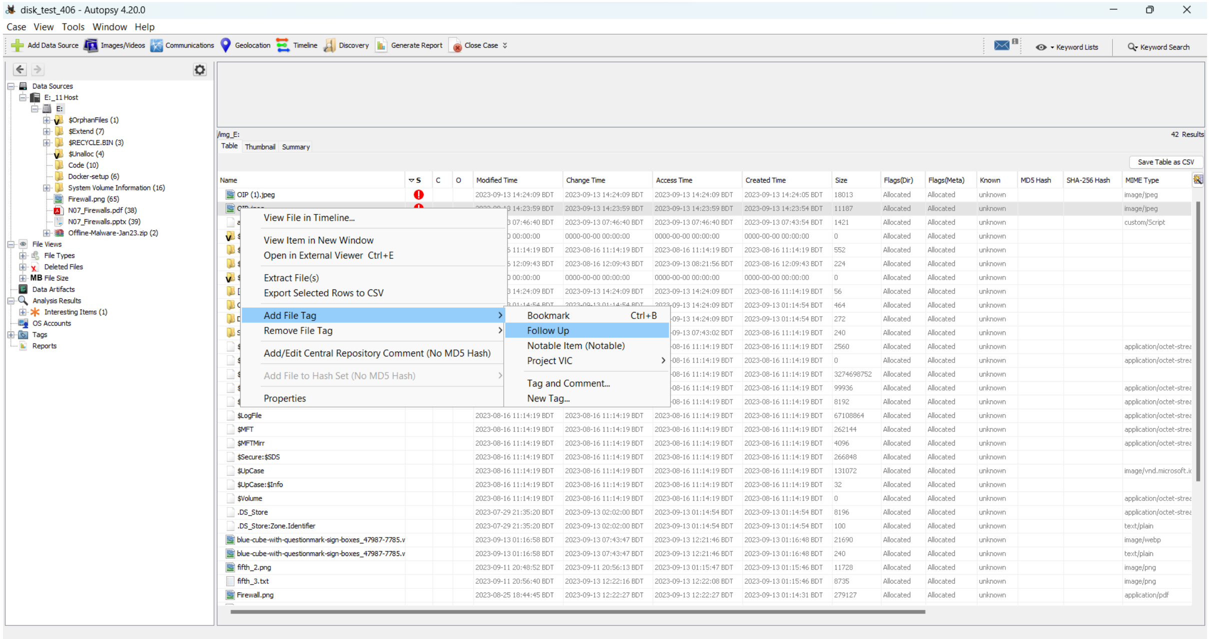
Timeline View

5 Report Generation

5.1 Tagging Files

Upon discovering an interesting thing, an user can tag it by right-clicking the item and selecting one of the tag options.

- **TagFile** file itself is of interest
- **TagResult** results of the analysis are of interest



Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	MD5 Hash	SHA-256 Hash	MIME Type
OIP (1).jpeg	!			2023-09-13 14:24:09 BDT	2023-09-13 14:24:09 BDT	2023-09-13 14:24:09 BDT	2023-09-13 14:24:05 BDT	18013	Allocated	Allocated	unknown			image/jpeg
OIP.jpeg	!			2023-09-13 14:23:59 BDT	2023-09-13 14:23:59 BDT	2023-09-13 14:23:59 BDT	2023-09-13 14:23:54 BDT	11187	Allocated	Allocated	unknown			image/jpeg
a				3 07:46:40 BDT	2023-09-13 07:46:40 BDT	2023-09-13 07:46:40 BDT	2023-09-13 07:43:54 BDT	1421	Allocated	Allocated	unknown			custom/Script
$				0 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0	Allocated	Allocated	unknown			
$				6 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	552	Allocated	Allocated	unknown			
$				6 12:09:43 BDT	2023-08-16 12:09:43 BDT	2023-09-13 08:21:56 BDT	2023-08-16 12:09:43 BDT	224	Allocated	Allocated	unknown			
$				0 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0	Allocated	Allocated	unknown			
[				3 14:24:09 BDT	2023-09-13 14:24:09 BDT	2023-09-13 14:24:09 BDT	2023-08-16 11:14:19 BDT	56	Allocated	Allocated	unknown			
C				3 01:14:54 BDT	2023-09-13 01:14:54 BDT	2023-09-13 14:24:09 BDT	2023-09-13 01:14:54 BDT	464	Allocated	Allocated	unknown			
D						-09-13 14:24:09 BDT	2023-09-13 01:14:54 BDT	272	Allocated	Allocated	unknown			
S						-09-13 07:43:02 BDT	2023-08-16 11:14:19 BDT	240	Allocated	Allocated	unknown			
$						-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2560	Allocated	Allocated	unknown			application/octet-strea
$						-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	0	Allocated	Allocated	unknown			application/octet-strea
$						-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	3274698752	Allocated	Allocated	unknown			
$						-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	99936	Allocated	Allocated	unknown			application/octet-strea
$						-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	8192	Allocated	Allocated	unknown			application/octet-strea
$LogFile				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	67108864	Allocated	Allocated	unknown			application/octet-strea
$MFT				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	262144	Allocated	Allocated	unknown			application/octet-strea
$MFTMirr				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	4096	Allocated	Allocated	unknown			application/octet-strea
$Secure:$SDS				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	266848	Allocated	Allocated	unknown			
$UpCase				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	131072	Allocated	Allocated	unknown			image/vnd.microsoft.i
$UpCase:$Info				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	32	Allocated	Allocated	unknown			
$Volume				2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	2023-08-16 11:14:19 BDT	0	Allocated	Allocated	unknown			application/octet-strea
.DS_Store				2023-07-29 21:35:20 BDT	2023-09-13 02:02:00 BDT	2023-09-13 01:14:54 BDT	2023-09-13 01:14:54 BDT	8196	Allocated	Allocated	unknown			application/octet-strea
.DS_Store:Zone.Identifier				2023-07-29 21:35:20 BDT	2023-09-13 02:02:00 BDT	2023-09-13 01:14:54 BDT	2023-09-13 01:14:54 BDT	100	Allocated	Allocated	unknown			text/plain
blue-cube-with-questionmark-sign-boxes_47987-7785.v				2023-09-13 01:16:58 BDT	2023-09-13 07:43:47 BDT	2023-09-13 12:21:46 BDT	2023-09-13 01:16:48 BDT	21690	Allocated	Allocated	unknown			image/webp
blue-cube-with-questionmark-sign-boxes_47987-7785.v				2023-09-13 01:16:58 BDT	2023-09-13 07:43:47 BDT	2023-09-13 12:21:46 BDT	2023-09-13 01:16:48 BDT	240	Allocated	Allocated	unknown			text/plain
fifth_2.png				2023-09-11 20:48:52 BDT	2023-09-11 20:56:13 BDT	2023-09-13 01:15:47 BDT	2023-09-13 01:15:46 BDT	11728	Allocated	Allocated	unknown			image/png
fifth_3.txt				2023-09-11 20:56:40 BDT	2023-09-13 12:22:16 BDT	2023-09-13 12:22:08 BDT	2023-09-13 01:15:46 BDT	8735	Allocated	Allocated	unknown			image/png
Firewall.png				2023-08-25 18:44:45 BDT	2023-09-13 12:22:27 BDT	2023-09-13 12:22:27 BDT	2023-09-13 01:14:31 BDT	279127	Allocated	Allocated	unknown			application/pdf

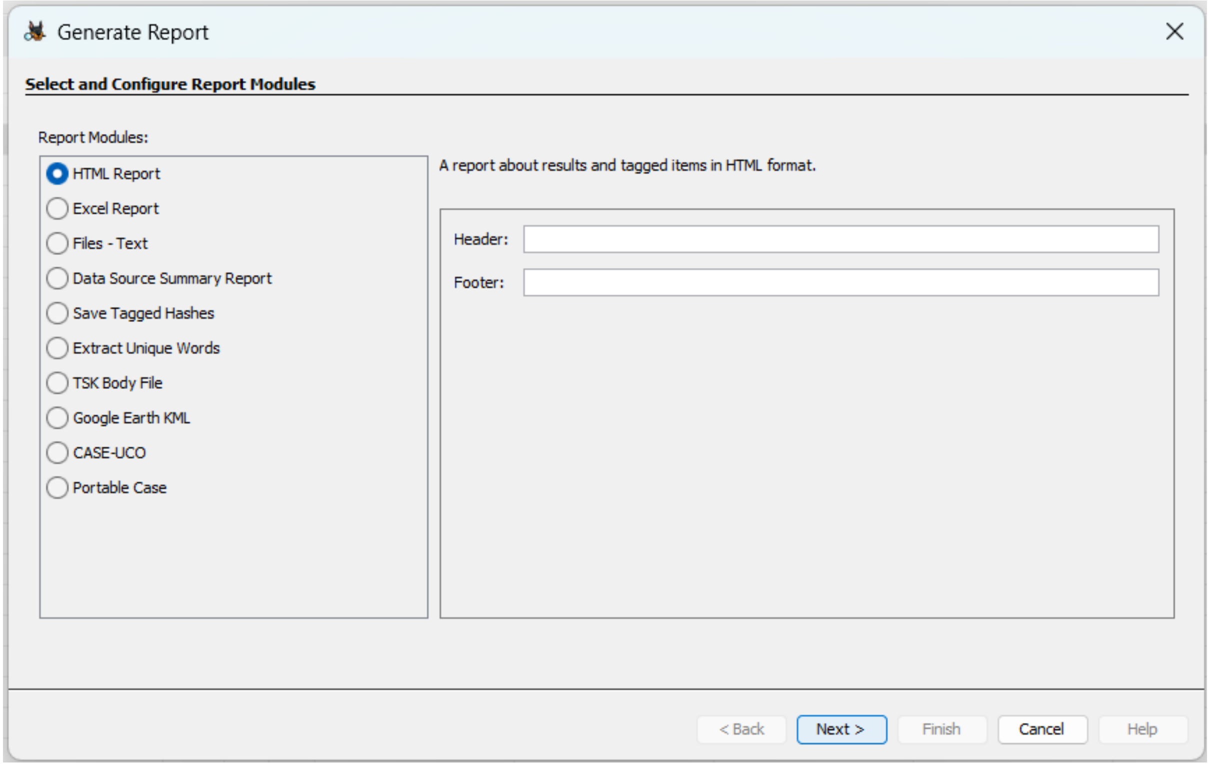
Tagging Files

5.2 Commenting

In Autopsy, commenting involves adding notes and annotations to digital artifacts during a forensic investigation. This aids documentation, collaboration, and context-building among investigators, streamlining evidence organization and enhancing investigative insights and clarity. This helps us to keep multiple tag on a same file and also to keep track of our investigation.

5.3 Reporting

The report modules allow the user to extract key information from a case in a variety of formats. This includes making an HTML or Excel report containing all the extracted content, keyword hits, etc. from a case, or creating a KML file out of any coordinates found to load into software like Google Earth.



Generate Report

Select and Configure Report Modules

Report Modules:

● HTML Report
○ Excel Report
○ Files - Text
○ Data Source Summary Report
○ Save Tagged Hashes
○ Extract Unique Words
○ TSK Body File
○ Google Earth KML
○ CASE-UCO
○ Portable Case

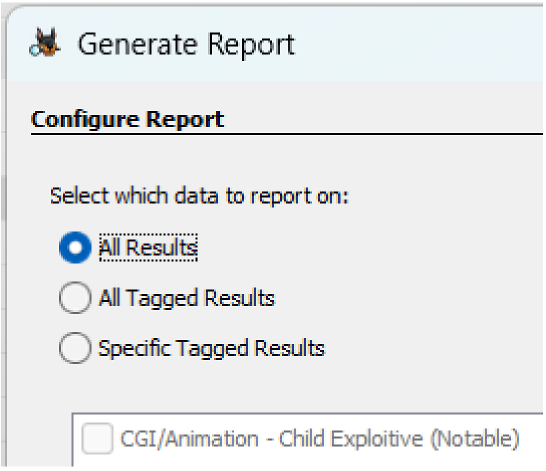
A report about results and tagged items in HTML format.

Header:

Footer:

Report Generation

Using the tagged files, Autopsy can generate a report containing all the information about the files. The report can be generated in multiple format. They can be found in **Reports** tab.



Generate Report

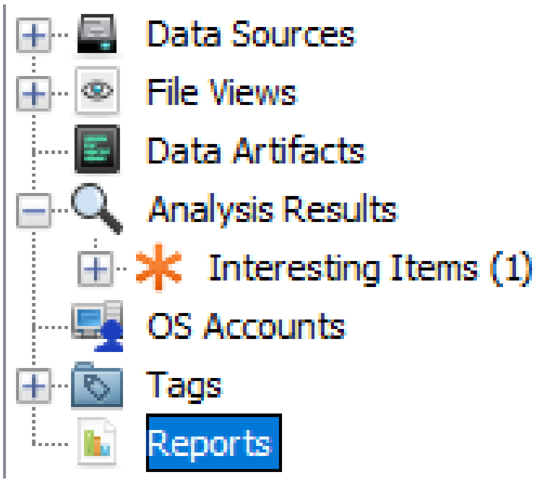
Configure Report

Select which data to report on:

● All Results
○ All Tagged Results
○ Specific Tagged Results

□ CGI/Animation - Child Exploitive (Notable)

Selecting Report Data



Generated Reports

6 Autopsy Functionality Testing

6.1 Extended DOS Partition Test

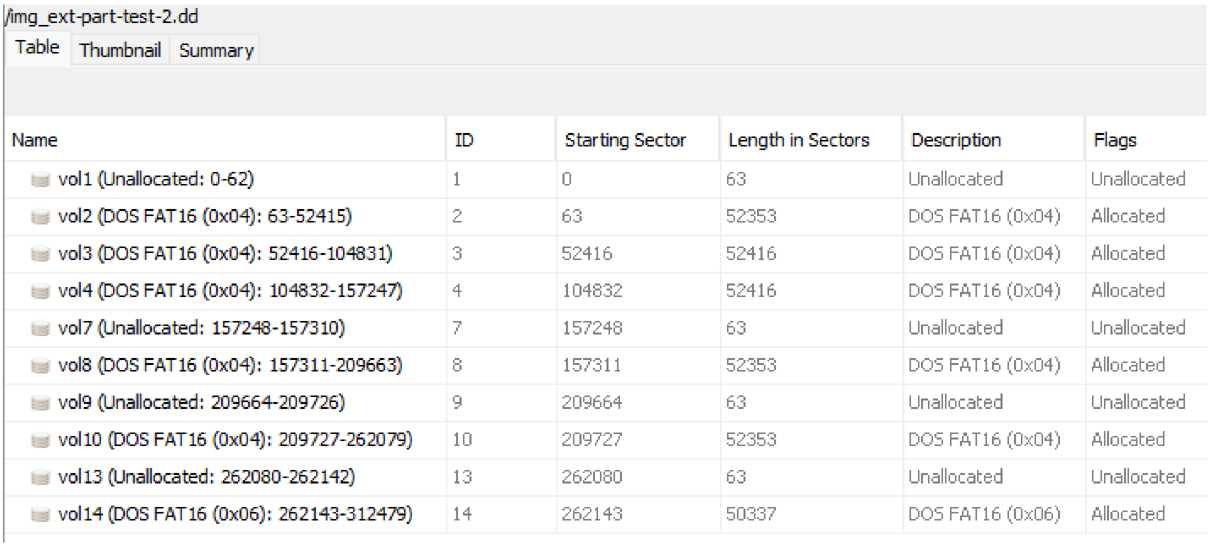
Introduction

Most DOS partition tools will not allow the user to create a third entry in an extended partition. A test image was created by modifying the partition table by hand with a hex editor and the system was booted. Both Windows and Linux read the third entry in the extended partition table and allowed the user to mount the partition. This test was to verify that forensic tools also allowed the investigator to view the partition in the third entry.

Source

The disk image can be found here:
`http://prdownloads.sourceforge.net/dftt/1-extend-part.zip?download`

Outputs from Autopsy



Name	ID	Starting Sector	Length in Sectors	Description	Flags
vol1 (Unallocated: 0-62)	1	0	63	Unallocated	Unallocated
vol2 (DOS FAT16 (0x04): 63-52415)	2	63	52353	DOS FAT16 (0x04)	Allocated
vol3 (DOS FAT16 (0x04): 52416-104831)	3	52416	52416	DOS FAT16 (0x04)	Allocated
vol4 (DOS FAT16 (0x04): 104832-157247)	4	104832	52416	DOS FAT16 (0x04)	Allocated
vol7 (Unallocated: 157248-157310)	7	157248	63	Unallocated	Unallocated
vol8 (DOS FAT16 (0x04): 157311-209663)	8	157311	52353	DOS FAT16 (0x04)	Allocated
vol9 (Unallocated: 209664-209726)	9	209664	63	Unallocated	Unallocated
vol10 (DOS FAT16 (0x04): 209727-262079)	10	209727	52353	DOS FAT16 (0x04)	Allocated
vol13 (Unallocated: 262080-262142)	13	262080	63	Unallocated	Unallocated
vol14 (DOS FAT16 (0x06): 262143-312479)	14	262143	50337	DOS FAT16 (0x06)	Allocated

Autopsy Partition Identification

Conclusion

- **Partiton Identification** Autopsy can correctly identify & disply the partition within disk image.
- **Extended Partition** Autopsy can correctly identify & disply the extended partition (Logical partitions) within primary extende partition.
- **Partition Content** Autopsy can display files & their properties within the partition even if the file has no content.

6.2 FAT Keyword Search

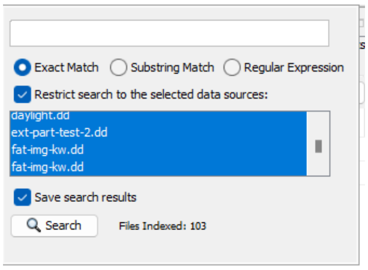
Introduction

This test image is a FAT file system with several ASCII strings. The goal of this test is to identify which tools can find different types of strings. Therefore, not all strings shown in the table below will be found. If one of the below strings is not found by a tool, that does not mean that the tool has an error in it. For example, the '1slack1' string crosses between the end of a file and into the slack space of the file. Some tools will find this and others will not. As long as the functionality of the tool is properly documented, then it is up to the user to use his tools in the needed way to gather the possible evidence.

Source

The disk image can be found here:
`http://prdownloads.sourceforge.net/dftt/2-kwsrch-fat.zip?download`
The MD5 of the image is bac12239bd466fa6c86ceb0b0426da0a.

Outputs from Autopsy



● Exact Match ○ Substring Match ○ Regular Expression

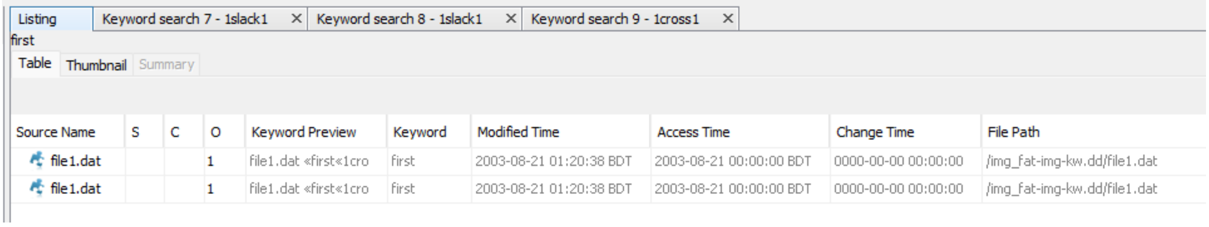
☑ Restrict search to the selected data sources:

daylight.dd
ext-part-test-2.dd
fat-img-kw.dd
fat-img-kw.dd

☑ Save search results

Files Indexed: 103

Autopsy Keyword Search Window



Listing
Keyword search 7 - 1slack1
Keyword search 8 - 1slack1
Keyword search 9 - 1cross1
Table
Thumbnail
Summary

first

Source Name	S	C	O	Keyword Preview	Keyword	Modified Time	Access Time	Change Time	File Path
file1.dat			1	file1.dat «first«1cro	first	2003-08-21 01:20:38 BDT	2003-08-21 00:00:00 BDT	0000-00-00 00:00:00	/img_fat-img-kw.dd/file1.dat
file1.dat			1	file1.dat «first«1cro	first	2003-08-21 01:20:38 BDT	2003-08-21 00:00:00 BDT	0000-00-00 00:00:00	/img_fat-img-kw.dd/file1.dat

Search Results for keyword 'first'

Conclusion

In autopsy we can search for keywords in the data source. Autopsy will search for the keyword in the file name, file content, and file metadata. Autopsy will also search for the keyword in the unallocated space of the data source.

In this test, we conducted a search within a FAT file system disk image. It's worth noting that Autopsy, a powerful forensic tool, can perform similar operations on various other file systems, including NTFS, EXT3FS, and more.

6.3 FTS Undelete Test

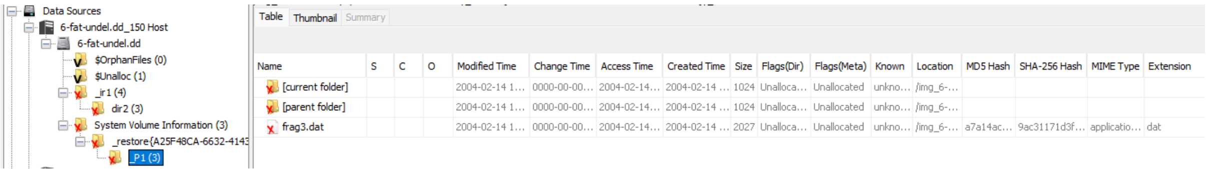
Introduction

This test image is a 6MB FAT file system with six deleted files and two deleted directories. The files range from single cluster files to multiple fragments. No data structures were modified in this process to thwart recovery. They were created in Windows XP, deleted in XP, and imaged in Linux.

Source

The disk image can be found here:
`http://prdownloads.sourceforge.net/dftt/6-undel-fat.zip?download`
The MD5 of the image is 4aeb06ecd361777242ab78735d51ace6.

Outputs from Autopsy



```
Data Sources
  6-fat-undel.dd_150 Host
    6-fat-undel.dd
      $OrphanFiles (0)
      $Unalloc (1)
      _ir1 (4)
        dir2 (3)
      System Volume Information (3)
        _restore{A25F48CA-6632-4143
          _P1 (3)
```

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location	MD5 Hash	SHA-256 Hash	MIME Type	Extension
[current folder]				2004-02-14 1...	0000-00-00...	2004-02-14...	2004-02-14 ...	1024	Unalloca...	Unallocated	unkno...	/img_6-...				
[parent folder]				2004-02-14 1...	0000-00-00...	2004-02-14...	2004-02-14 ...	1024	Unalloca...	Unallocated	unkno...	/img_6-...				
frag3.dat				2004-02-14 1...	0000-00-00...	2004-02-14...	2004-02-14 ...	2027	Unalloca...	Unallocated	unkno...	/img_6-...	a7a14ac...	9ac31171d3f...	applicatio...	dat

Autopsy FTS Undelete

Conclusion

Autopsy can recover & display deleted files & directories within a file system.

6.4 JPEG Search Test

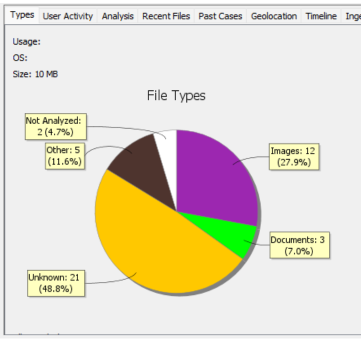
Introduction

This test image is an NTFS file system with 10 JPEG pictures in it. The pictures include files with incorrect extensions, pictures embedded in zip and Word files, and alternate data streams. The goal of this test image is to test the capabilities of automated tools that search for JPEG images.

Source

The disk image can be found here:
`http://prdownloads.sourceforge.net/dftt/8-jpeg-search.zip?download`
The MD5 of the image is 9bdb9c76b80e90d155806a1fc7846db5.

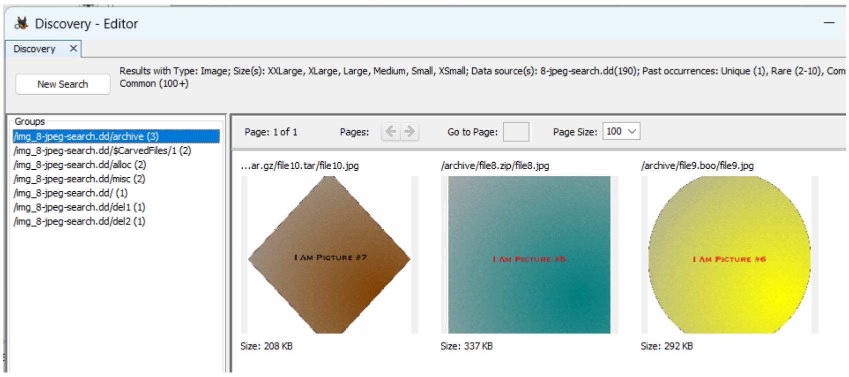
Outputs from Autopsy



Usage:
OS:
Size: 10 MB

Autopsy JPEG Search Summary

From the figure below we can see that Autopsy can find JPEG images within the data source even if the image extention is incorrect. Also if another file has jpeg extension but it's not a jpeg image, Autopsy will detect correct file type & display accordingly.



Results with Type: Image; Size(s): XXLarge, XLarge, Large, Medium, Small, XSmall; Data source(s): 8-jpeg-search.dd(190); Past occurrences: Unique (1), Rare (2-10), Com
Common (100+)

/img_8-jpeg-search.dd/archive (3)
/img_8-jpeg-search.dd/$CarvedFiles/1 (2)
/img_8-jpeg-search.dd/alloc (2)
/img_8-jpeg-search.dd/misc (2)
/img_8-jpeg-search.dd/ (1)
/img_8-jpeg-search.dd/del1 (1)
/img_8-jpeg-search.dd/del2 (1)

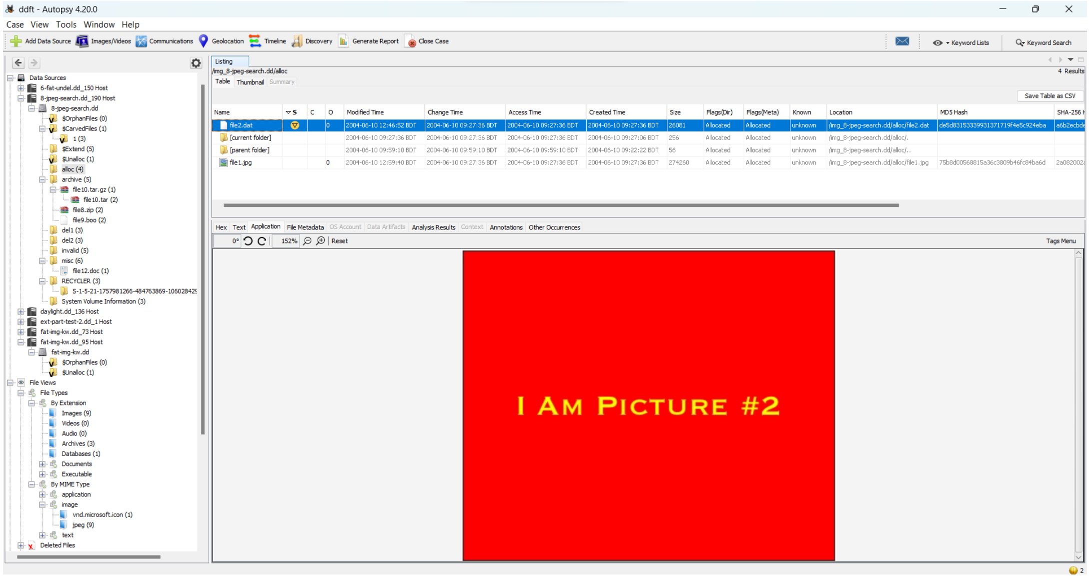
Page: 1 of 1
Go to Page:
Page Size: 100

...ar.gz/file10.tar/file10.jpg
Size: 208 KB

/archive/file8.zip/file8.jpg
Size: 337 KB

/archive/file9.boo/file9.jpg
Size: 292 KB

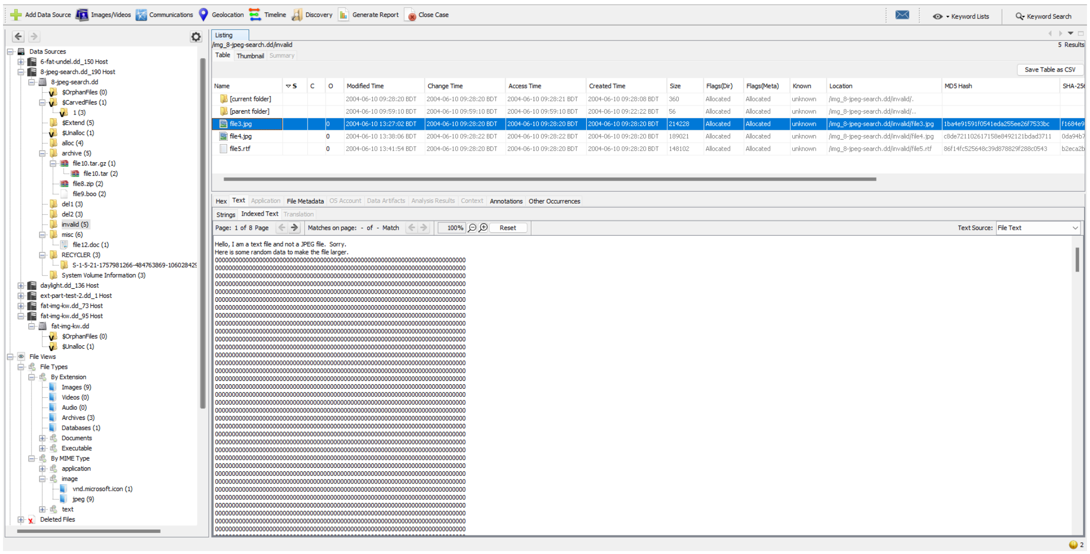
All 12 JPEG Images Found by Autopsy



/img_8-jpeg-search.dd/alloc 4 Results

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location	MD5 Hash	SHA-256 H
file2.dat	▽		0	2004-06-10 12:46:52 BDT	2004-06-10 09:27:36 BDT	2004-06-10 09:27:36 BDT	2004-06-10 09:27:36 BDT	26081	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/alloc/file2.dat	de5d83153339931371719f4e5c924eba	a6b2ecbde
[current folder]				2004-06-10 09:27:36 BDT	2004-06-10 09:27:36 BDT	2004-06-10 09:27:36 BDT	2004-06-10 09:27:06 BDT	256	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/alloc/.		
[parent folder]				2004-06-10 09:59:10 BDT	2004-06-10 09:59:10 BDT	2004-06-10 09:59:10 BDT	2004-06-10 09:22:22 BDT	56	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/alloc/..		
file1.jpg			0	2004-06-10 12:59:40 BDT	2004-06-10 09:27:36 BDT	2004-06-10 09:27:36 BDT	2004-06-10 09:27:36 BDT	274260	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/alloc/file1.jpg	75b8d00568815a36c3809b46fc84ba6d	2a082002

Detecting Images with Corrupted Extension



/img_8-jpeg-search.dd/invalid 5 Results

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location	MD5 Hash	SHA-25
[current folder]				2004-06-10 09:28:20 BDT	2004-06-10 09:28:20 BDT	2004-06-10 09:28:21 BDT	2004-06-10 09:28:08 BDT	360	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/invalid/.		
[parent folder]				2004-06-10 09:59:10 BDT	2004-06-10 09:59:10 BDT	2004-06-10 09:59:10 BDT	2004-06-10 09:22:22 BDT	56	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/invalid/..		
file3.jpg			0	2004-06-10 13:27:02 BDT	2004-06-10 09:28:20 BDT	2004-06-10 09:28:20 BDT	2004-06-10 09:28:20 BDT	214228	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/invalid/file3.jpg	1ba4e91591f0541eda255ee26f7533bc	f1684e9
file4.jpg			0	2004-06-10 13:38:06 BDT	2004-06-10 09:28:22 BDT	2004-06-10 09:28:22 BDT	2004-06-10 09:28:20 BDT	189021	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/invalid/file4.jpg	c8de721102617158e8492121bdad3711	0da94b7
file5.rtf			0	2004-06-10 13:41:54 BDT	2004-06-10 09:28:20 BDT	2004-06-10 09:28:20 BDT	2004-06-10 09:28:20 BDT	148102	Allocated	Allocated	unknown	/img_8-jpeg-search.dd/invalid/file5.rtf	86f14fc525648c39d878829f288c0543	b2eca2b

Page: 1 of 8 Page Matches on page: - of - Match 100% Text Source: File Text

```
Hello, I am a text file and not a JPEG file.  Sorry.
Here is some random data to make the file larger.
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

Detecting Other File Types with JPEG Extension

Conclusion

Autopsy identified standard JPEG files with correct extensions and seamlessly recognized JPEGs with non-standard extensions, treating them as valid images. Notably, it adeptly detected false positives, even when files had JPEG extensions but lacked actual JPEG content. It also managed partial JPEGs and accurately identifying valid signatures, even without complete headers and footers. It also effectively recovered deleted JPEGs, regardless of their extensions, and competently handled deleted JPEGs with incorrect extensions. Autopsy's capabilities extended to locating and extracting embedded JPEGs within archive formats and unconventional file structures. Additionally, it excelled in recognizing and extracting JPEGs embedded within document files and detecting hidden JPEGs within alternate data streams, making it a valuable tool for forensic analysis.

6.5 NTFS Autodetect Test

Introduction

This test has four images. One is a disk image that contains two partitions and the partitions are also included as individual files. The fourth image is an additional partition image. The purpose of this test case is to test the file system detection routines of your analysis tools. A typical partition contains only one file system, but the layout of some file systems allows multiple file systems to exist in a single partition. Each of the partitions in this disk image contain two file systems. The first partition is formatted for NTFS and Ext2, the second is formatted for NTFS and UFS2, and the third is formatted for NTFS and UFS1. Both file systems are valid and can be mounted in their respective operating system. The test is whether your tool will warn you that there are two valid file systems or if it will show you only one and hide the other.

Source

The disk image can be found here:
`http://prdownloads.sourceforge.net/dftt/10b-ntfs-autodetect.zip?download`
This test case has one 'raw' disk image and two 'raw' partition images. In total, the images are 275 MB, but they compress to under 1 MB.
The MD5 of the disk image is 9225ff95b92311a28b2224e9dc324231,
Partition 1 is 6bd741152ccedd50e12623af5eeba803,
Partition 2 is 0b768efb1011b047c9831c1e00d1706c,
Partition 3 is 5984253cad72d4950c15a9e679139daf

Image Details

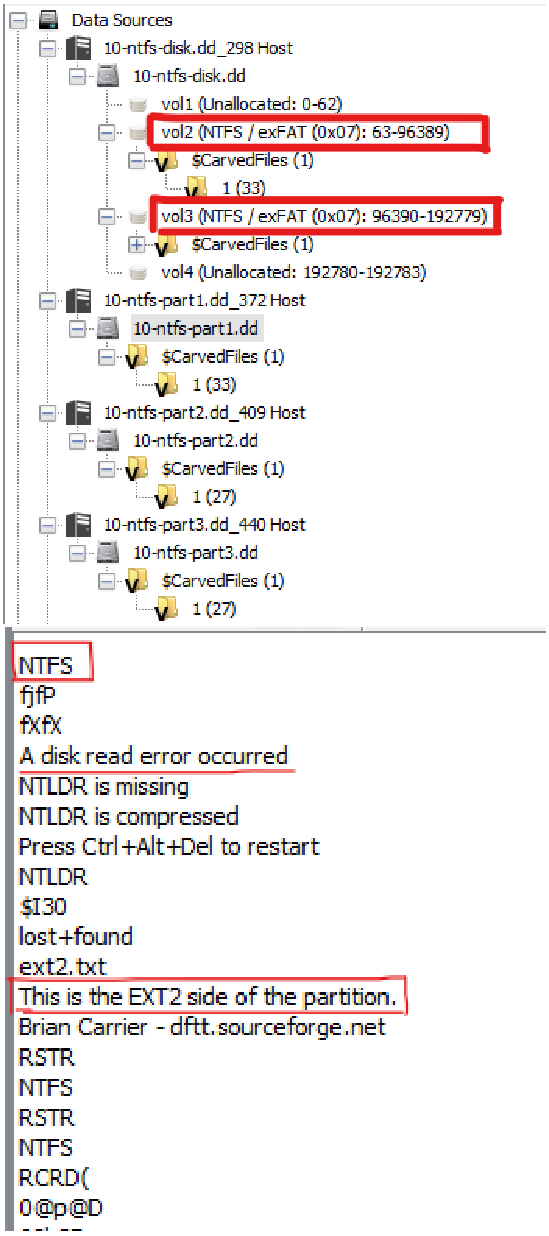
- The 10-ntfs-disk.dd image has a DOS partition table with two partitions (10-ntfs-part1.dd and 10-ntfs-part2.dd). Each partition has a type of 0x07, which the NTFS type. The contents of each partition are described next.

- The 10-ntfs-part1.dd image is from the first partition in 10-ntfs-disk.dd. It was originally formatted as NTFS from within Windows XP and the ntfs.txt file was created. The partition was then formatted as Ext2 and the ext2.txt file was created. The NTFS file system was then checked for errors from within Windows and none were found.

- The 10-ntfs-part2.dd image is from the second partition in 10-ntfs-disk.dd. It was originally formatted as NTFS from within Windows XP and the ntfs.txt file was created. The partition was then formatted as UFS2 and the ufs1.txt file was created. The NTFS file system was then checked for errors from within Windows and none were found. This image was supposed to be UFS1, but I incorrectly formatted it as FreeBSD UFS2.

- The 10-ntfs-part3.dd image is not in the 10-ntfs-disk.dd image and was created because 10-ntfs-part2.dd was incorrectly formatted as UFS2 instead of UFS1. The partition was formatted as NTFS from within Windows XP and the ntfs.txt file was created. The partition was then formatted as UFS1 and the ufs1.txt file was created. The NTFS file system was then checked for errors from within Windows and none were found.

Outputs from Autopsy



```
Data Sources
  10-ntfs-disk.dd_298 Host
    10-ntfs-disk.dd
      vol1 (Unallocated: 0-62)
      vol2 (NTFS / exFAT (0x07): 63-96389)
        $CarvedFiles (1)
          1 (33)
      vol3 (NTFS / exFAT (0x07): 96390-192779)
        $CarvedFiles (1)
      vol4 (Unallocated: 192780-192783)
  10-ntfs-part1.dd_372 Host
    10-ntfs-part1.dd
      $CarvedFiles (1)
        1 (33)
  10-ntfs-part2.dd_409 Host
    10-ntfs-part2.dd
      $CarvedFiles (1)
        1 (27)
  10-ntfs-part3.dd_440 Host
    10-ntfs-part3.dd
      $CarvedFiles (1)
        1 (27)
```

```
NTFS
fjfP
fXfX
A disk read error occurred
NTLDR is missing
NTLDR is compressed
Press Ctrl+Alt+Del to restart
NTLDR
$I30
lost+found
ext2.txt
This is the EXT2 side of the partition.
Brian Carrier - dftt.sourceforge.net
RSTR
NTFS
RSTR
NTFS
RCRD(
0@p@D
```

Autopsy NTFS Autodetect

Conclusion

Autopsy can correctly detect & display multiple file systems within a partition.

7 Demonstration Video

Demonstration of this tool can be found in https://youtu.be/cWCEsyyR7xE

8 References

- To download Autopsy, visit: `https://www.autopsy.com/download/`
- For more information about Autopsy, visit: `http://www.sleuthkit.org/autopsy/`
- For digital forensics test images, visit: `https://dftt.sourceforge.net/`