Digital Forensics (IFS4102)
Lab 4: Disk & File Forensics

Lab Objectives

In this lab, you will perform several ***disk & file forensics tasks*** on a target machine's acquired disk image and its contained files.

More specifically, in this lab you want to:

1. Inspect an acquired disk image file by **identifying** the disk & file-system information, and listing deleted files using **The Sleuth Kit (TSK)**.
2. Manually analyze a **file signature**, and also to fix a file concealed with **extension mismatch**, i.e. its correct extension has been modified.
3. (*Optional*) Manually extract and view the **metadata** of:
    a. Microsoft Office files;
    b. Images files.
4. Use **Autopsy** to perform the following automated file related tasks:
    a. Browse and extract **deleted files** in the examined file system;
    b. Perform **file type identification** and explore the identified files;
    c. Find out files with **extension mismatch**;
    d. Perform a **hash-lookup analysis** of your target file system;
    e. (*Optional*) View the **Exif data** of image files;
    f. (*Optional*) Extract **archive formats**, including Microsoft Office file formats, so that Autopsy can subsequently analyze the files inside of archive files.

Task 1 (Lin-FWS): Inspecting a Disk Image using TSK

Important Notes:

- In this exercise, you will use a CLI-based yet very powerful tool called **The Sleuth Kit (TSK)**.
- You can run the steps below in a **Linux terminal**. Kali Linux already has TSK pre-installed.
- Use the disk image file of your thumb drive, which you created by using either FTK Imager or dd* tools in your previous labs. Alternatively, you can use a previously shared sample disk image "SuspectDrive1.E01" from https://drive.google.com/file/d/1SiJOaWEmKYyXXKI-PEJP3-I2j1iOC_p3/view?usp=sharing. Its MD5 value is b66270513117670d11ebe2191e947a6d.

Steps:

1. To find out more about the disk image's **file format**, run TSK's commands under the "**Image Management**" (img) tool category. You can run `img_stat`, which shows the details of the image format as follows:
   `# img_stat <image-file>`
   Check the reported image type, e.g. `raw`, and also the image's size shown in bytes.
2. To find out the image's **partition structures**, use commands under the "**Media/Volume Management**" (mm) tool category. Run `mmstat` to display the details about a volume system:
   `# mmstat <image-file>`

3. Run `mmls` to display the **layout of the disk**, including the unallocated spaces:

   `# mmls <image-file>`

   Check the reported type of a target partition type (e.g. FAT16) and its **offset** (e.g. 32).

4. Subsequently, use some commands under the "**File System**" (fs) tool category to discover more about the file system structure of a target partition.

   Run the following `fsstat` command to show the list of accepted arguments for **image type**:

   `# fsstat -i list`

   Notice the available image formats, such as `raw`, `ewf`, `aff`.

5. Run the following `fsstat` command to show the list of accepted arguments for **file-system type**:

   `# fsstat -f list`

   Notice the available file system formats, such as `fat12`, `fat16`, `fat32`, `ntfs`, `ext3`, `ext4`.

6. Now, run `fsstat` to **identify a target file system** as follows (assuming the image type is `raw`, the file system type is `fat16`, and the offset of the target partition is 32):

   `# fsstat -i raw -f fat16 -o 32 <image-file>`

7. Observe the output, and answer the following questions:

   a. What are the Volume ID and Volume Label of the partition?

   b. What is the sector size of the partition?

   c. What is the cluster size of the partition?

8. Run the `fls` command under the "File" Name (f) tool category:

   `# fls -i raw -f fat16 -o 32 <image-file>`

9. Observe the output. Can you notice:
    a. Directory entries, which are listed with "`d/d`"? (Note: the first letter is from the corresponding item's entry type, while the second letter is the type according to its inode).
    b. File entries, which are listed with "`r/r`"?
    c. Deleted entries, which are marked with "`*`"?
10. Run the last command above with the following additional flags:
    a. `-d`: for **showing deleted entries** only;
    b. `-l`: for **long listing**
    c. `-r -p`: for **recursive listing** together with full pathnames

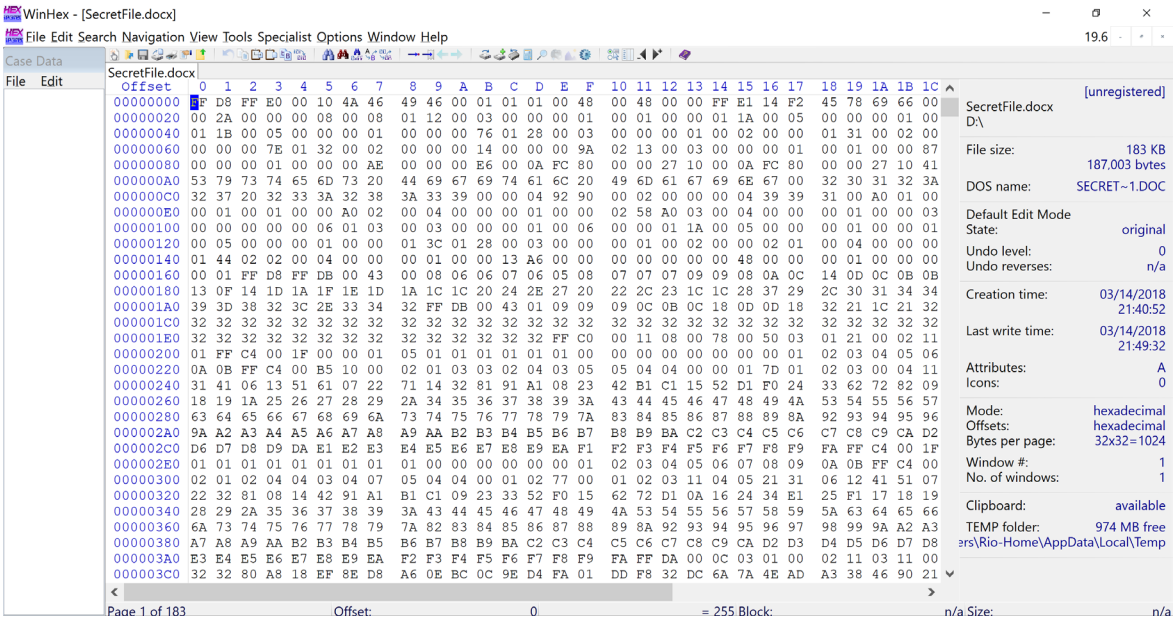
Task 2 (Win-FWS): Performing File Signature Analysis and Fixing a Concealed File with Extension Mismatch

Notes:

- Now you want to manually inspect the **file signature** of a file, whose extension has been **intentionally altered** in order to conceal the file. Once you find out the correct extension, you want to **fix** the file by renaming its extension accordingly, so that the file can be opened.
- Please download a sample file "SecretFile.docx" from: https://drive.google.com/file/d/1JIqXu4yG1SOsXFrhG-eImZadJZsDs8-l/view?usp=sharing. Its MD5 value is 9fb26f5619bd8993d762dfe9060bc7b9.

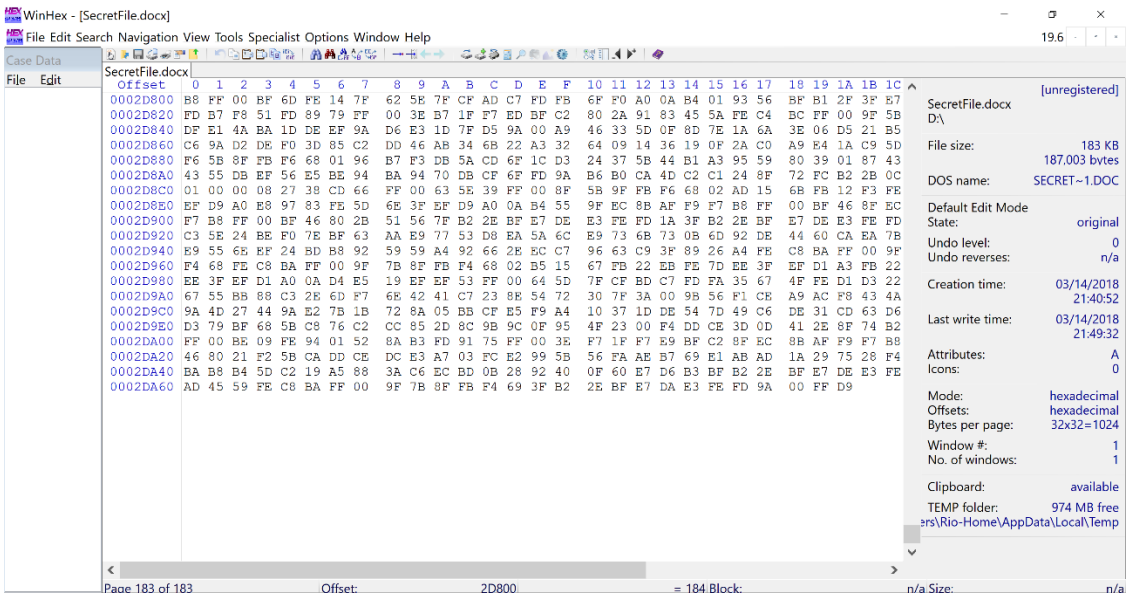
Steps:

1. Try to open the docx file using **Microsoft Word**.
   What is wrong? What error message do you receive?
2. Now you want to check the file by performing a **file signature analysis**.
   For this, you want to use a hex editor, such as WinHex.
   If still needed, download WinHex from https://www.x-ways.net/winhex/.
   It is a fully portable application that can be executed without any installation.
   Launch WinHex simply by invoking its executable `WinHex.exe`.
3. Open the target docx file.
4. Notice the file's **header**, i.e. the *first few bytes* of the file, which can help us identify the correct type of the file:



5. Visit https://www.garykessler.net/library/file_sigs.html to check possible file types.

6. You can also check the file's **footer**, i.e. its *last few bytes* of the file:



7. Based on its file signature, what is the correct extension of the file? Rename its extension accordingly.

8. Open the renamed file and inspect the displayed file now!

[Optional] Task 3-A (Win-FWS): Viewing the Metadata of Microsoft Office Files

Notes:

- Given a Microsoft Office file, you want to check its **metadata** for additional information about the file.

Steps:

1. Download a sample doc file named “SampleFile.docx” from: https://drive.google.com/file/d/1SDuA7v80wlLkSit-1D3WbspBAhDfBXQM/view?usp=sharing.
   Its MD5 value is f996cc9b3af5a74ceec29cdc3edb295a.
2. Rename the file into “SampleFile.zip”.
3. Open the renamed file in File Explorer by double-clicking it.
4. Double-click the item named “docProps”. Two files are contained, namely: app.xml and core.xml.
5. Double-click core.xml. You should be able to inspect the file metadata as shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <cp:coreProperties xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dcmitype="http://purl.org/dc/dcmitype/"
xmlns:dcterms="http://purl.org/dc/terms/" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-properties">
    <dc:title/>
    <dc:subject/>
    <dc:creator>Rio-Home</dc:creator>
    <cp:keywords/>
    <dc:description/>
    <cp:lastModifiedBy>Rio-Home</cp:lastModifiedBy>
    <cp:revision>1</cp:revision>
    <dcterms:created xsi:type="dcterms:W3CDTF">2018-03-14T13:55:00Z</dcterms:created>
    <dcterms:modified xsi:type="dcterms:W3CDTF">2018-03-14T13:56:00Z</dcterms:modified>
</cp:coreProperties>
```

6. Inspect the machine name used to create/modify the file, as well as the file creation and last modification times.

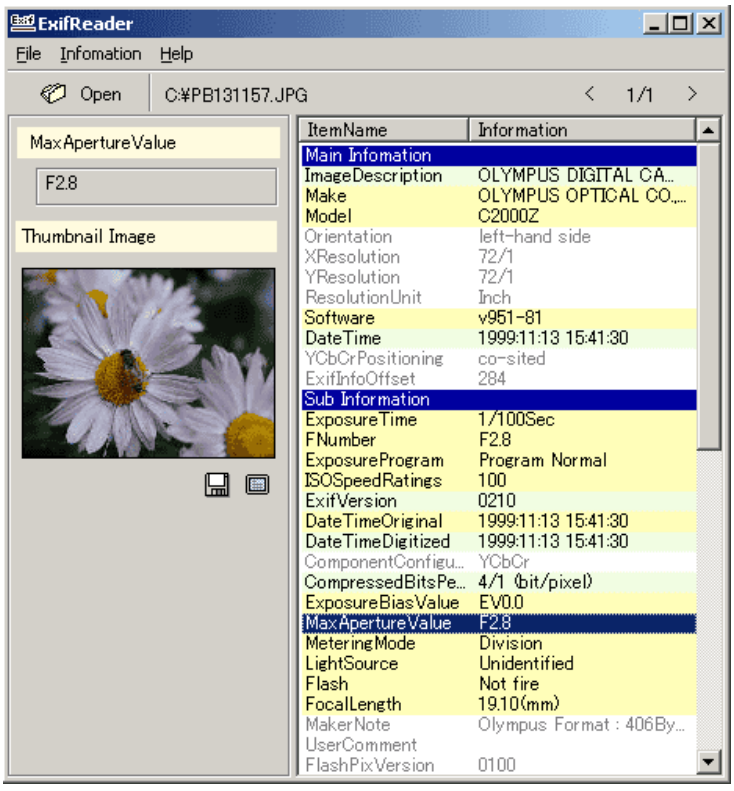
[Optional] Task 3-B (Win-FWS/Lin-FWS): Viewing the Metadata of Image Files

Notes:

- Given an image file, especially those generated by a digital camera or a mobile phone, you want to check its **Exif** (Exchangeable image file format) **metadata** for additional information about the file.

Steps:

1. Download and install an Exif reader software of your choice. Alternatively, you may also utilise a web-based tool, such as http://exif.regex.info/exif.cgi, to check a *non-confidential* image file.
2. Open an image file, which may come from your digital camera, phone or PC/notebook.
3. A sample window output of an Exif reader is shown below:



4. Check various pieces of information regarding the image file, such as:
    a. The **time** when the picture was taken;
    b. The **device** used to take the picture;
    c. ***GPS location*** of the camera/phone.
5. For more information on Exif data, you can visit: https://en.wikipedia.org/wiki/Exif.

Task 4 (Win-FWS): Performing Various Automated Forensics Tasks using Autopsy

Let's now use Autopsy to perform more tasks, including those conducted in the previous tasks of this lab, but in an *automated manner*.

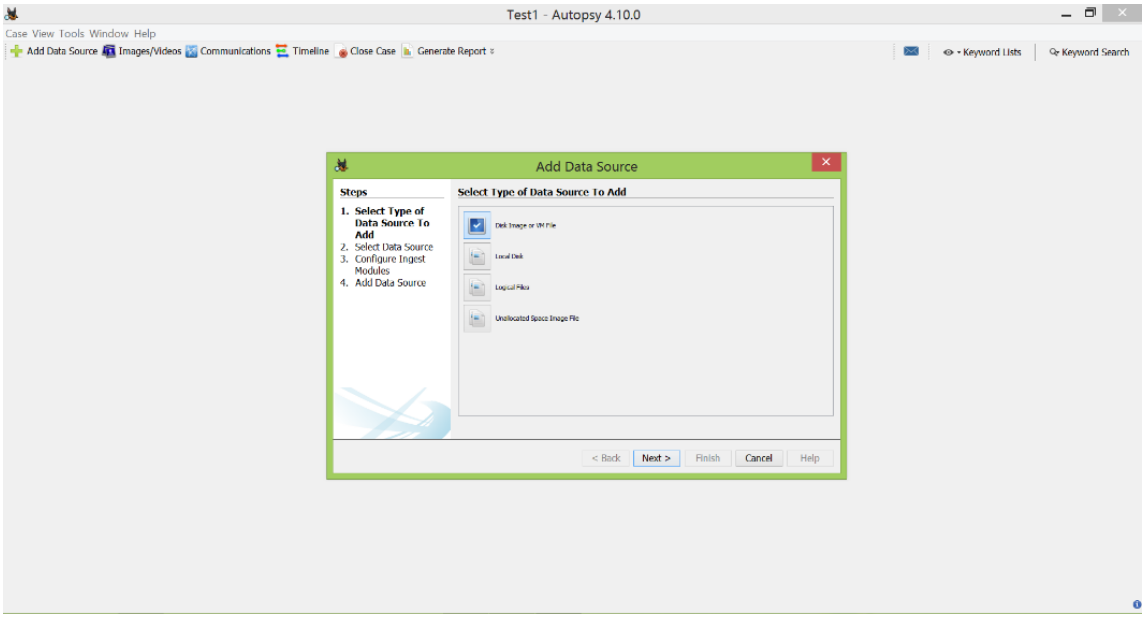
Task 4-A (Win-FWS): Browsing and Extracting All Deleted Files in the Examined File System using Autopsy

Notes:

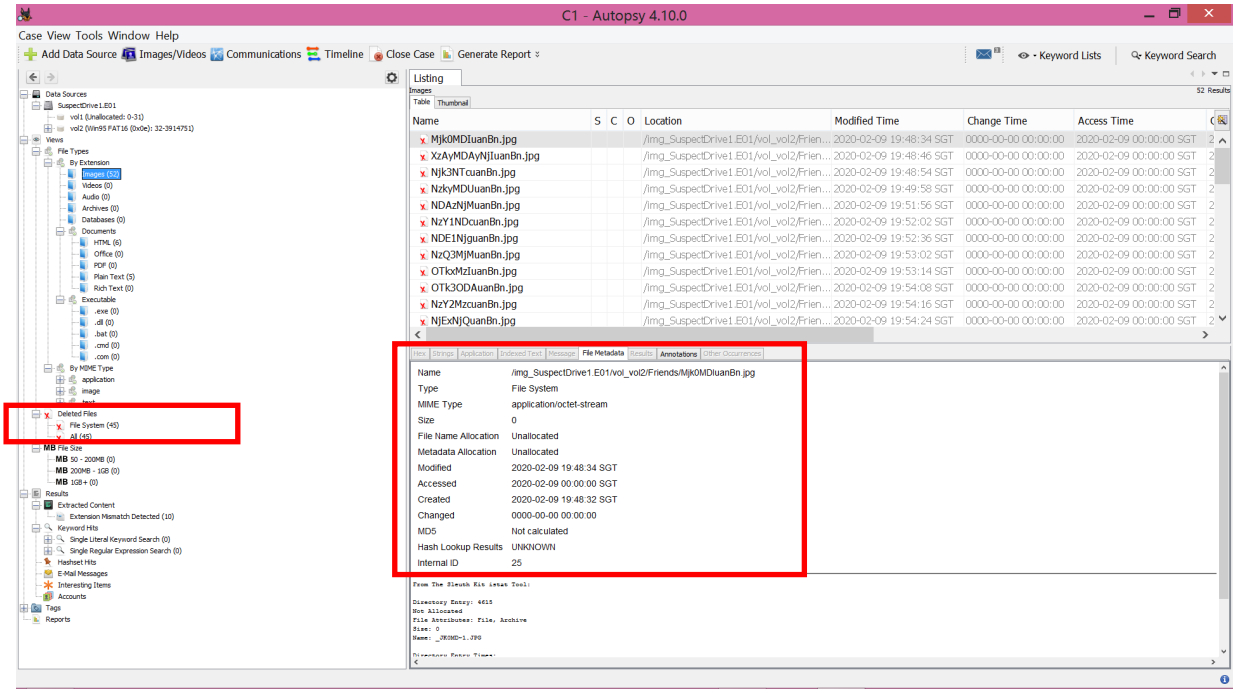
- In this task, you want to use Autopsy to **browse all deleted files** in the examined file system and **extract** some of them.
- You can use the "SuspectDrive1.E01" file that was previously used in Task 1 if needed.

Steps:

1. Launch Autopsy, and then enter your case information.
2. When the "Add Data Source" window appears, select "Disk Image or VM File", and then browse to your downloaded SuspectDrive1.E01 file.



3. In the "Views" section of the tree viewer, you should be able to see a folder named "**Deleted Files**" as shown below.



4. Do explore the listed deleted files, and check their shown *file metadata* as well.

5. Try to additionally extract some deleted files shown on the list.
   Use Windows' File Explorer to check the recovered files,
   including their file metadata.

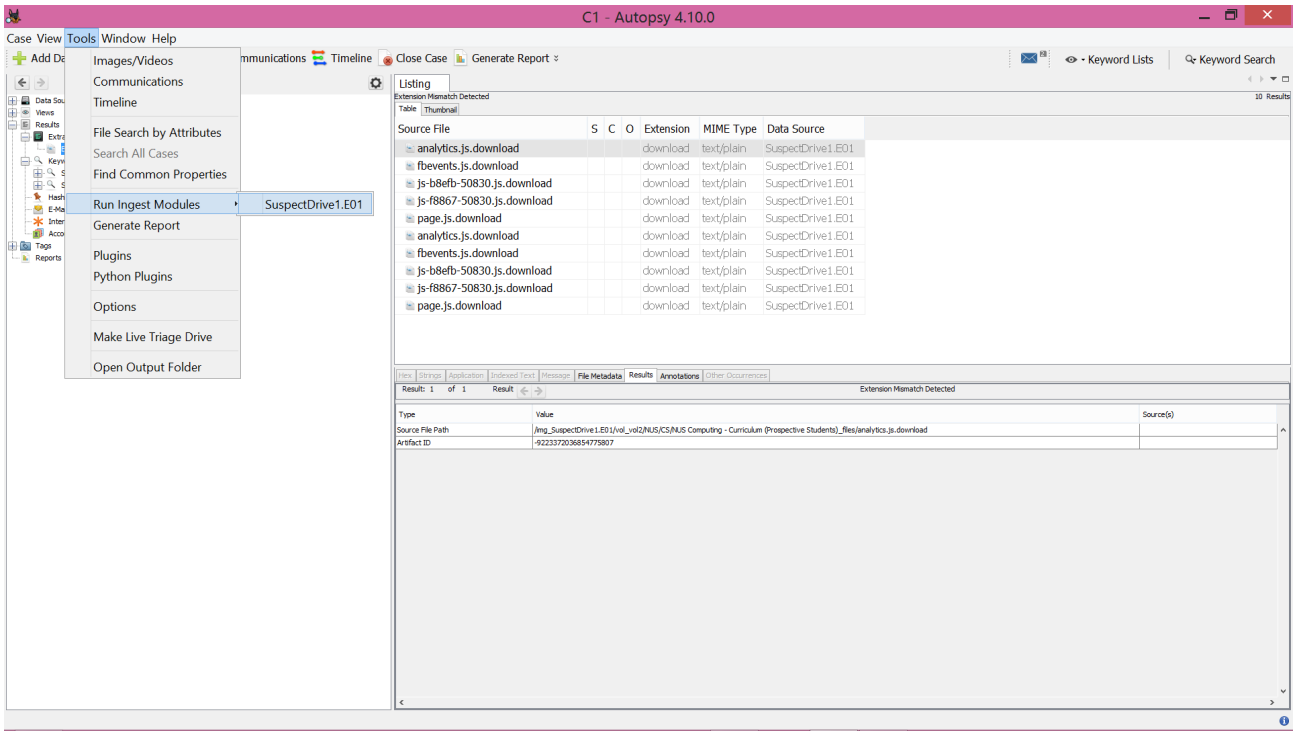
Task 4-B (Win-FWS): Performing File Type Identification using Autopsy

Notes:

- Now, you want to use Autopsy to perform **file type identification** based on the files' *internal signatures*, and then explore the identified files.
- Again, you can use the "SuspectDrive1.E01" file that was previously used.

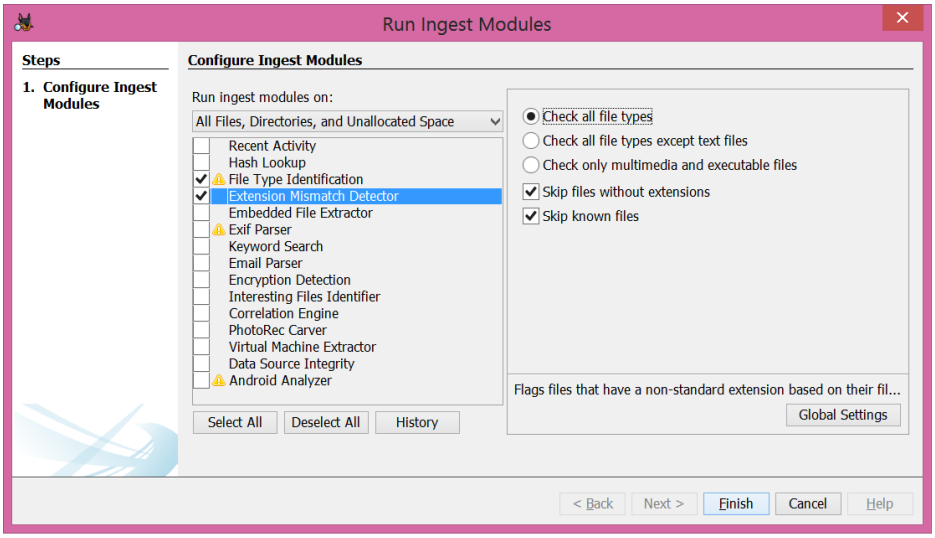
Steps:

1. Launch Autopsy, and add the disk image file as a data source (if still needed).
2. If you need to relaunch the "Run Ingest Modules" dialog box, you can do so by accessing the "Tools" menu item as shown below:

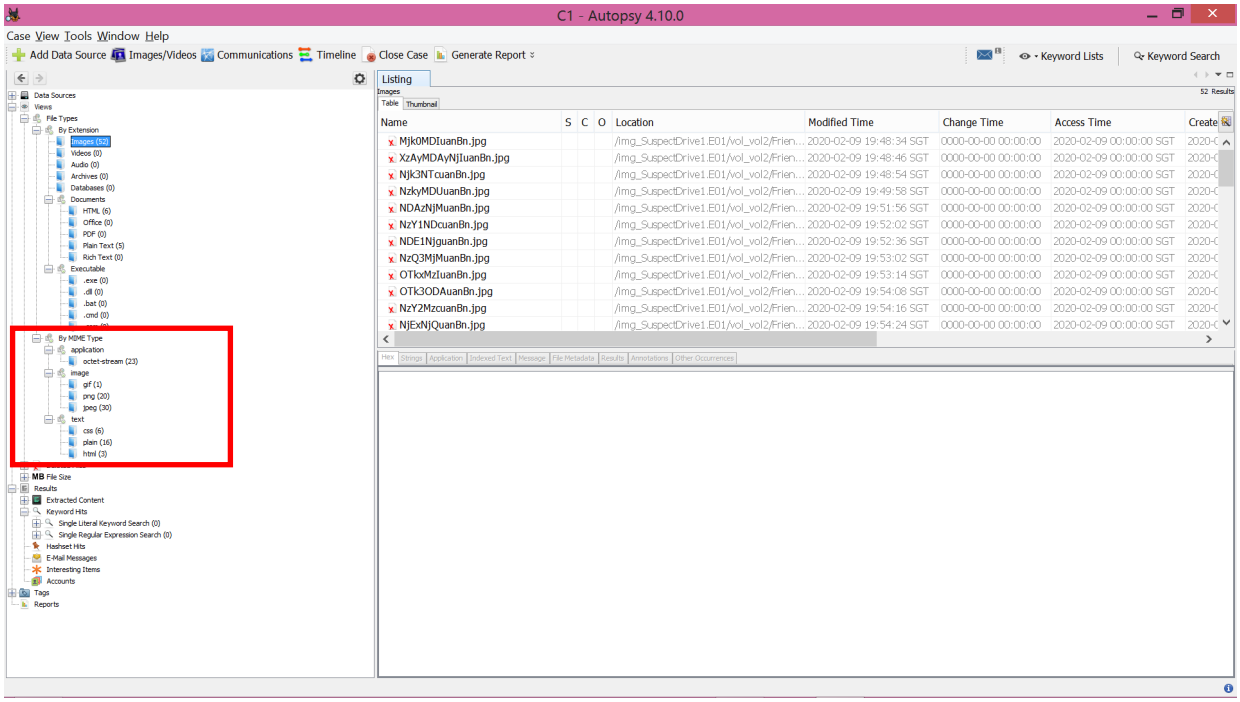


Alternatively, right-click the disk entry and select "Run Ingest Modules".

3. In the "Run Ingest Modules" dialog box, do select the "**File Type Identification**" module with its default settings as below:



4. Once the ingest module finishes its identification process files based on the files' *internal signatures* instead of their file extensions, you should be able to see a folder named "**By MIME Type**" in the "Views" section as shown below.



5. Do explore the identified files. You can additionally explore the files as grouped "By Extension" in the Views section, and inspect any differences.

(**Reference**: See https://sleuthkit.org/autopsy/docs/user-docs/4.19.3/file_type_identification_page.html).

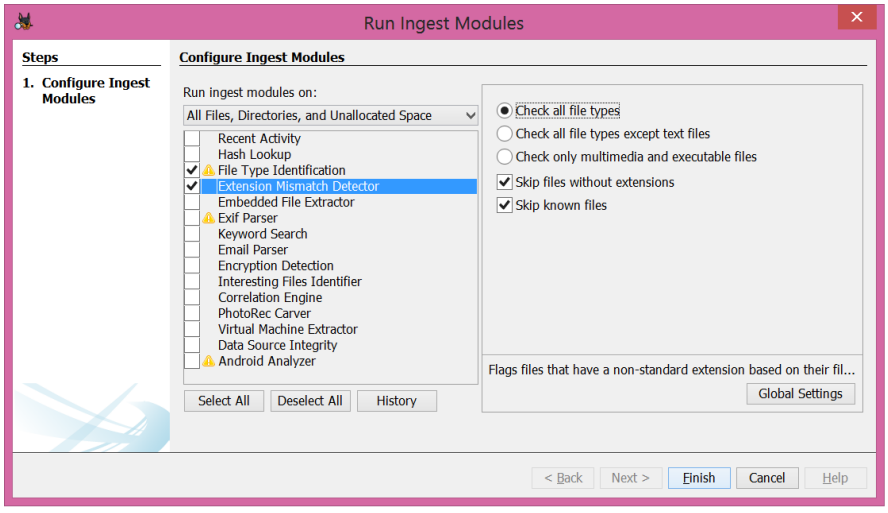
Task 4-C (Win-FWS): Automatically Finding out Files with Extension Mismatch using Autopsy

Notes:

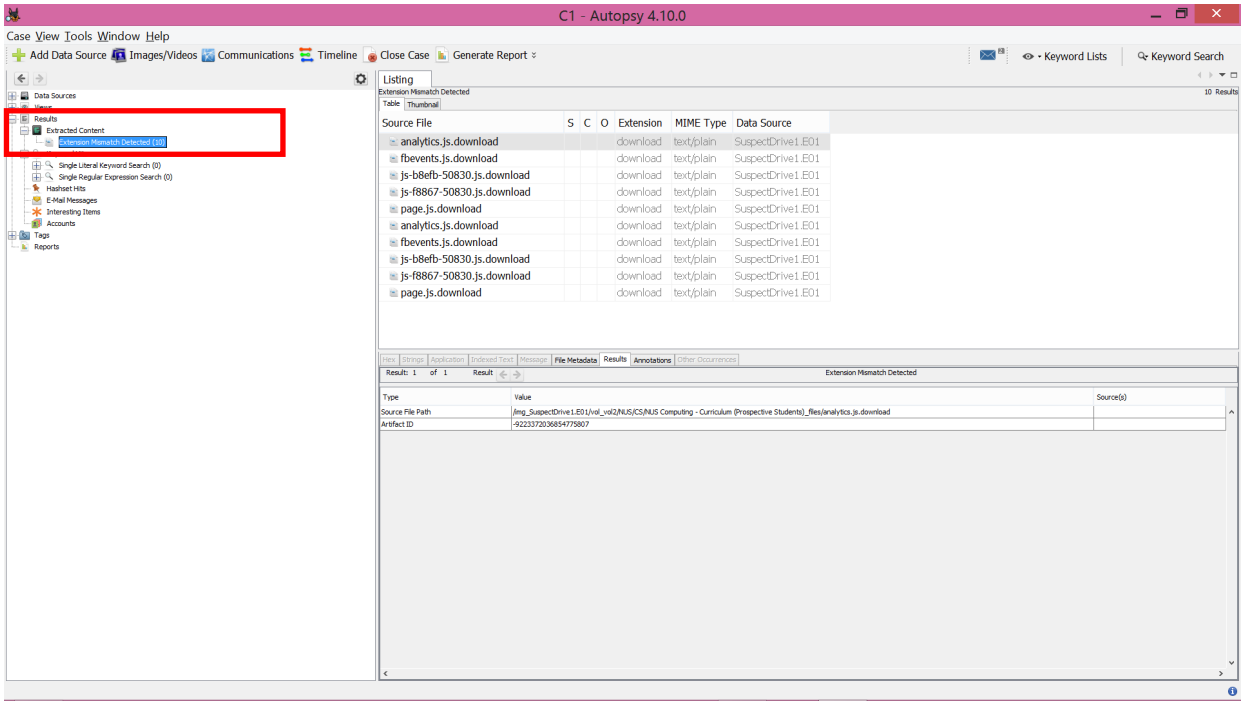
- Now, you want to use Autopsy to automatically find out files with **extension mismatch**.
- Again, you can use the "SuspectDrive1.E01" file that was previously used.

Steps:

1. Launch Autopsy, and add the disk image file as a data source (if still needed).
2. Launch the "Run Ingest Modules" dialog box, then select the "**Extension Mismatch Detector**" module. You can use the default settings as shown below. Note that this ingest also requires the "File Type Identification" module to be additionally enabled or previously run.



3. Once the ingest module finishes its process, you should be able to see a folder named "**Extension Mismatch Detected**" under the "Extracted Content" entry in the "Results" section as shown below.



4. Do explore the detected files, and do manual analysis as in Task 2 if needed. (Note that some identified files are with the ".download" extension. The files were temporarily created by a browser during file-download processes from the Internet.)

(**Reference**: https://sleuthkit.org/autopsy/docs/user-docs/4.19.3/extension_mismatch_detector_page.html).

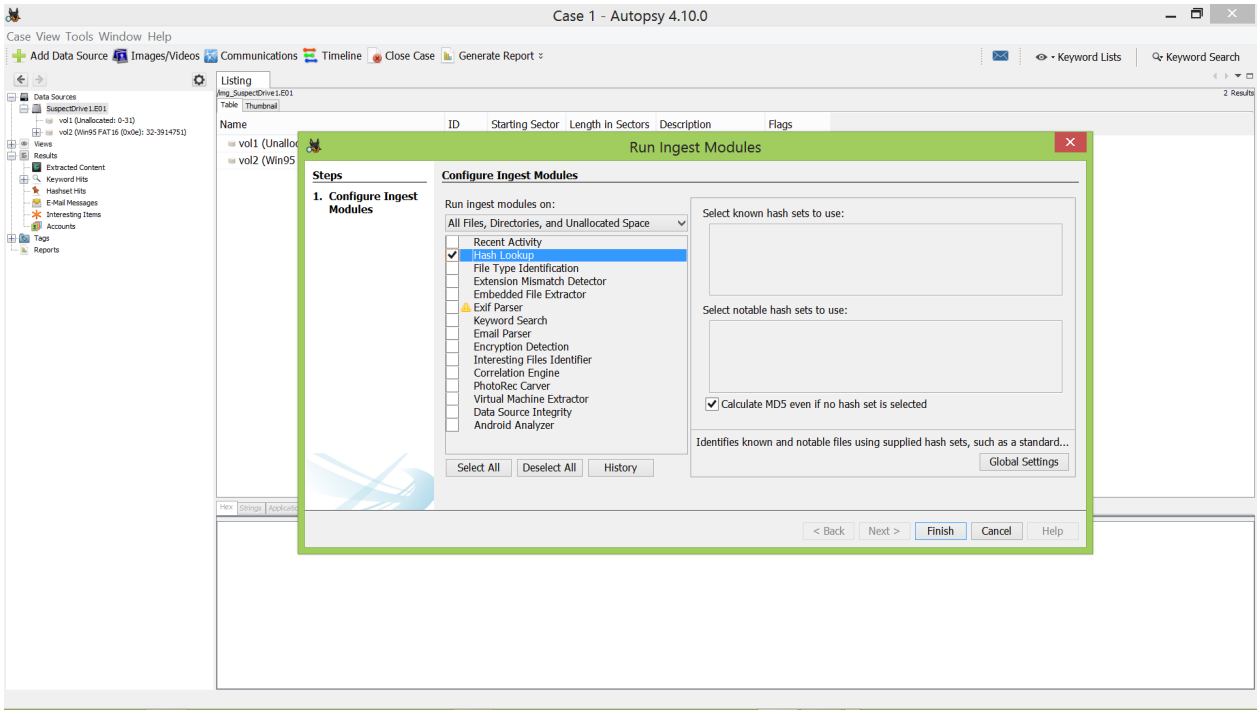
Task 4-D (Win-FWS): Performing Hash-Lookup Analysis of a File System using Autopsy

Notes:

- Now, you want to perform a **hash-lookup analysis** of a target file system using your own defined/downloaded hash-set file.
- Again, you can use the "SuspectDrive1.E01" file that was previously used.
- Also download a sample hash-set file, "target-hash-set.txt", from https://drive.google.com/file/d/18ARP_onDmu6PrRY6MDelF2GEPCJhl_Xp/view?usp=sharing.

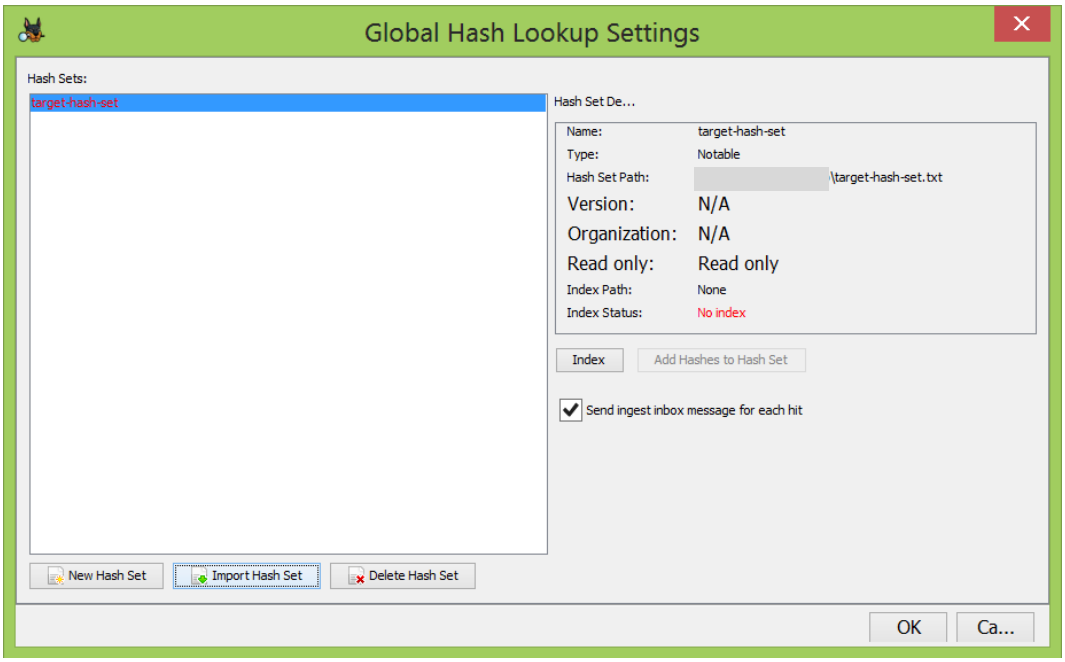
Steps:

1. Launch Autopsy, and add the disk image file as a data source (if still needed).
2. Launch the "Run Ingest Modules" dialog box, and just select the **"Hash Lookup"** module as shown below.



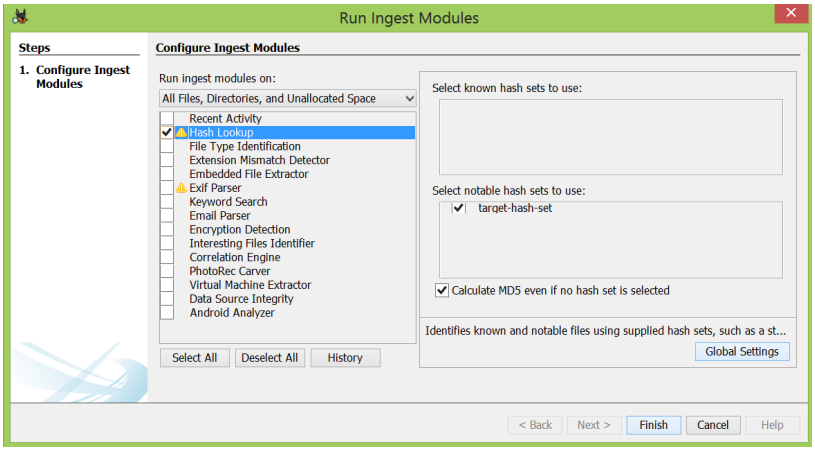
3. Then, click the "Global Settings" button, so that you can access the "Global Hash Lookup Setting" window.

4. Click on the "Import Hash Set" button, and then browse to your downloaded target-hash-set.txt file. You can select the "Notable" option for the "Type of hash set". The imported hash set file is still shown *in red* on the hash sets list as shown below (due to its "No index" *index status* explained more below).



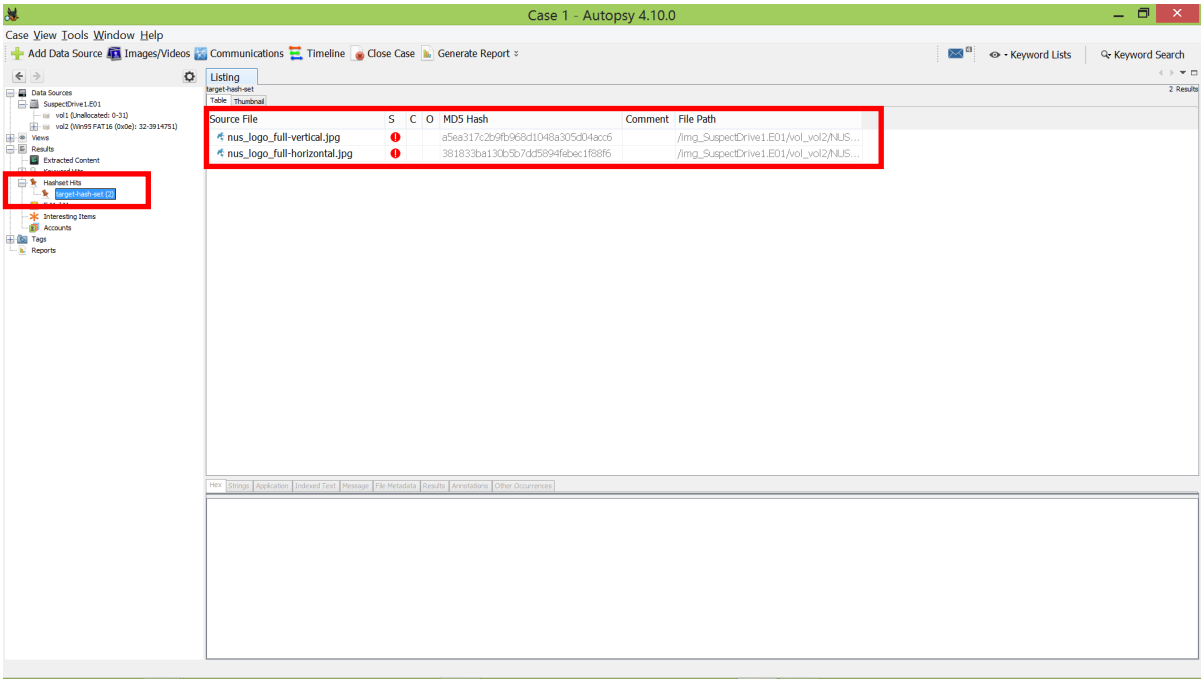
5. Notice that the file's index status is "**No index**". An imported hash set file needs to be indexed first so that Autopsy can perform its hash lookup operations. Hence, click the "Index" button. An index file with ".idx" extension will be created in the folder where you put your hash set file.
   Please ensure that, in the setting window, the hash-set filename is now shown in black. Click the "OK" button to close the setting window.

6. Confirm that the entered hash set file is now selected as a notable hash set to use as shown below.



7. Click the "Finish" button. Autopsy will automatically perform its hash lookup.

8. Once the lookup operation finishes, you can see an entry named "**Hashes Hits**" under the "Results" section in the tree viewer of Autopsy. From the indicated number, you can see how many files are hit by the supplied hash set file. The corresponding files in your target drive are shown in the list pane as below.



9. If needed, you can extract the file and double-check the hash values using tools like HashMyFiles (https://www.nirsoft.net/utils/hash_my_files.html).

(**Reference**: For more information about the ingest module, see: https://sleuthkit.org/autopsy/docs/user-docs/4.19.3/hash_db_page.html).

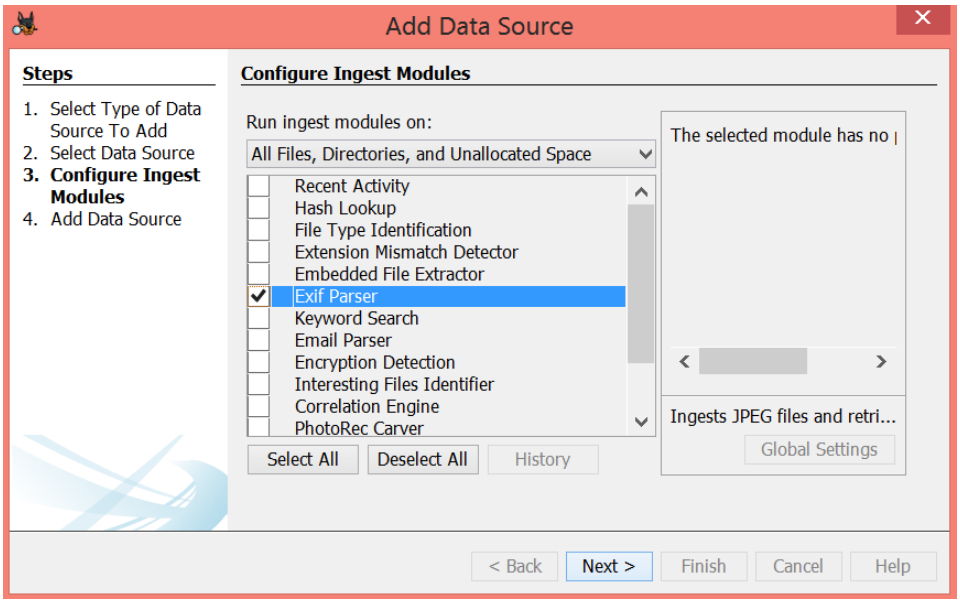
[Optional] Task 4-E (Win-FWS): Viewing Exif Data of Image Files using Autopsy

Notes:

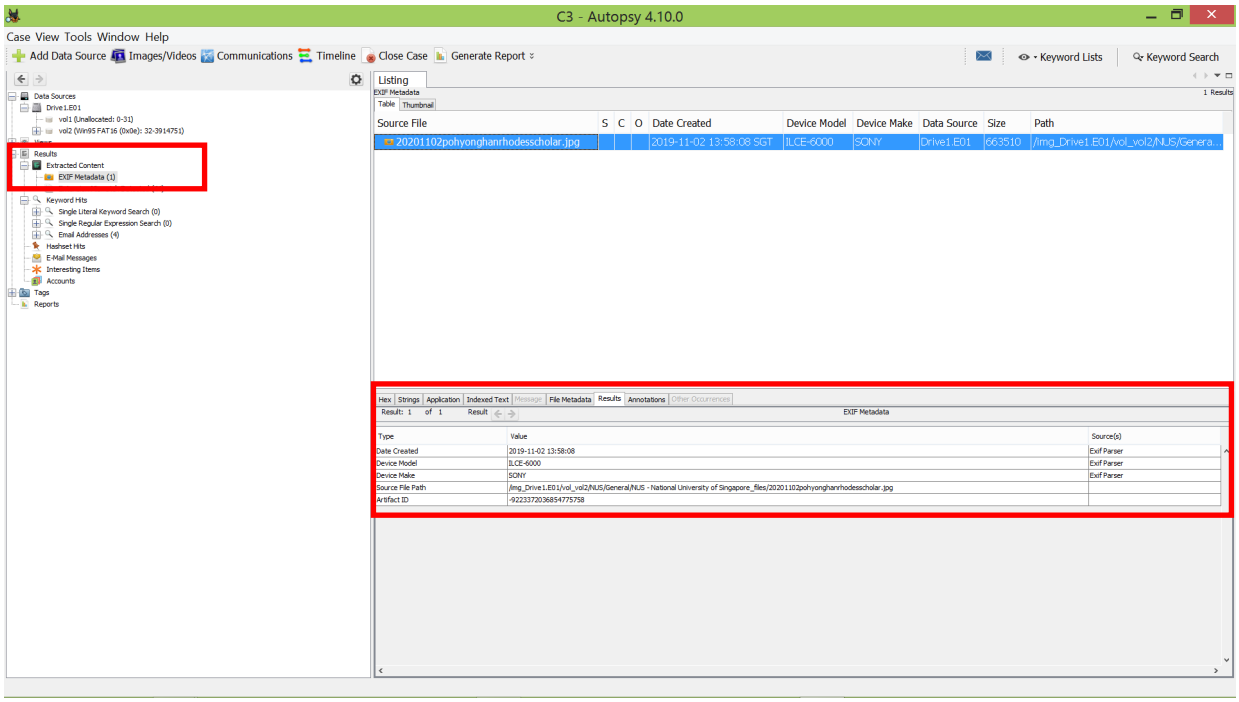
- Now, you want to use Autopsy to view the **Exif data** of image files.
- You can use your disk image file as your target disk.

Steps:

1. Launch Autopsy, and add your disk image file as a data source (if still needed).
2. In the "Run Ingest Modules" dialog box, select the **"Exif Parser"** module as shown below (or the "**Picture Analyzer**" module in the latest version of Autopsy), and then run it.



3. Once the parsing operation finishes, you can find an entry named "**EXIF Metadata**" under the "Extracted Content" entry in the "Results" section. A sample screenshot is shown below.



4. Explore all the listed files. You can also click on the "Results" tab of a listed file to view the parsed Exif metadata of the file.

(**Reference**: For more information about the ingest module, see: https://sleuthkit.org/autopsy/docs/user-docs/4.19.3/_e_x_i_f_parser_page.html).

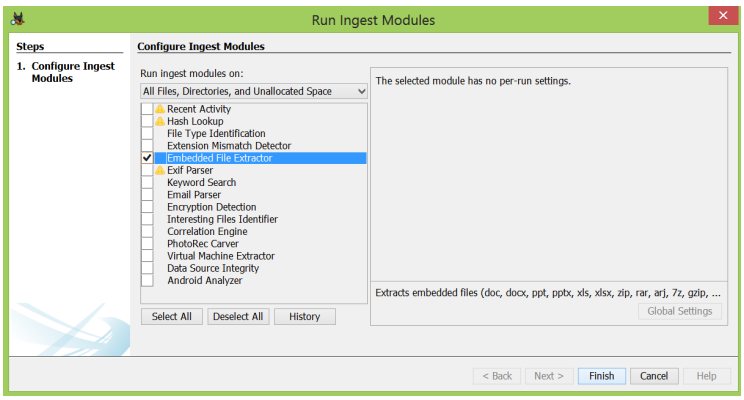
[Optional] Task 4-F (Win-FWS): Extracting Archive Formats for Subsequent File Analysis using Autopsy

Notes:

- Lastly, you want to extract ***archive formats***, such as ZIP, RAR, and Microsoft Office file formats, using the "**Embedded file extractor**" module. Following the extraction, Autopsy can then analyze the files inside of target archive files, for instances, when performing hash lookup and keyword search operations.
- Note that "certain media content embedded inside Doc/Docx, PPT/PPTX, and XLS/XLSX might *not* be extracted" (see also its reference page: https://sleuthkit.org/autopsy/docs/user-docs/4.19.3/embedded_file_extractor_page.html).

Steps:

1. Launch Autopsy, and add your disk image file as a data source (if still needed).
2. In the "Run Ingest Modules" dialog box, select the **"Embedded File Extractor"** module as shown below and then run it.



3. Each file extracted shows up in the data-source tree viewer as **a child** of the archive file containing it, and also as **an archive** under the "Views" → "File Types" → "Archives".

Graded Lab Tasks #2 (2 Marks)

From your Lab 4, you will need to submit **3 answers** as follows:

- The selected **3 tasks** in this lab are:
  - (0.75 marks) **Task 1, Step 6 (page 3)**: Please copy and paste the output of running the specified TSK command using the given **sample disk image** "SuspectDrive1.E01". Like your Graded Lab Tasks #1, you can just put the first 30 lines of the command's output.
  - (0.5 marks) **Task 1**, **Step 7-b (page 3):** Answer the given question, i.e. "What is the sector size of the partition?".
  - (0.75 marks) **Task 2, Step 7 (page 6)**: Specify the correct extension.
- From your correct 3 answers, you will earn a total of **2 marks**.
- This graded lab task assignment is an **individual** assignment.
  Hence, you MUST finish the assignment and report **independently**.
- Please prepare your answers in a self-contained **PDF file** by using your **name and matric number** as part of your file name. For example, Jack Lee with Matric No A001 should submit the filename JackLee-A001-**GLT2**.pdf. (*Note*: If you submit multiple files to the Canvas's assignment, a *counter suffix* is automatically added by Canvas for version tracking purposes. We will take your latest submission there.) Your report should also contain your name, matric number, and email address on its first page.
- Upload your PDF file using **`Graded-Lab-Tasks-2`** Canvas Assignment by **Saturday, 11 February 2023, 23:59 SGT**. Note that this deadline is a ***firm & final*** **deadline**. There will be ***no*** deadline extensions.
  As such, you are advised to submit **well before** the cut-off time so as to avoid any technical issues with Canvas or your uploading!

Have fun with your assigned tasks in this lab! :)