KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade
Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

School of Computing

Department of Computer Science and Engineering

Digital Forensics

Integrated Course-Theory

Lab Record (213CSE4307)

Student Name: -

Register Number: -

Section & slot: -

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade
Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

School of Computing

Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

Bonafide record of work done by _____

of_____ in _____

_____during even/odd semester in academic year _____.

Staff In-charge

Submitted to the practical Examination held at Kalasalingam University, Krishnankoil on _____

REGISTER NUMBER

Internal Examiner **External Examiner**

School of Computing

Department of Computer Science and

EngineeringAcademic Year: 2025-2026

Subject Name: Digital Forensics **Year / Sem: III Year – V Semester**

Course Handling Faculty: Dr. K. Venkatesh

Slot / Section:

S.NO	DATE	TITLE OF THE PROGRAM	MARKS	SIGN
1		Create a forensic image of a storage device using FTK Imager or EnCase, and verify its integrity with hash values.		
2		Recover deleted or damaged files from a storage device using Test Disk and Foremost		
3		Capture and analyze network traffic using Wireshark.		
4		Analyze email headers and detect email spoofing using MHA.		
5		Use Autopsy to create a case and import evidence.		
6		Use Sleuth Kit to analyze digital evidence.		
7		Extract and analyze data from Android devices using LiME for memory acquisition.		
8		Use zsteg / Steg Secret to detect hidden data in images.		
9		Use Procmon to identify suspicious processes.		
10		Using REMnux for the Malware analysis.		

Ex.No.1 Evidence Acquisition with FTK Imager

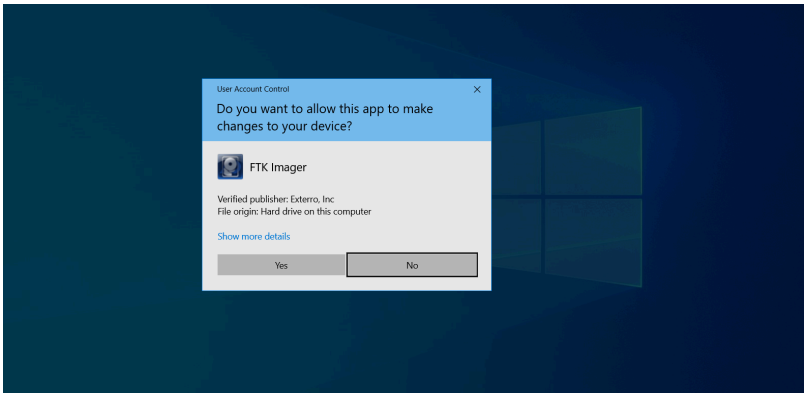
Aim

To acquire **volatile memory (RAM)** and **non-volatile memory (disk image)** from a target system using **AccessData FTK Imager**, while preserving integrity through hashing and proper documentation.

Acquiring Volatile Memory (RAM)

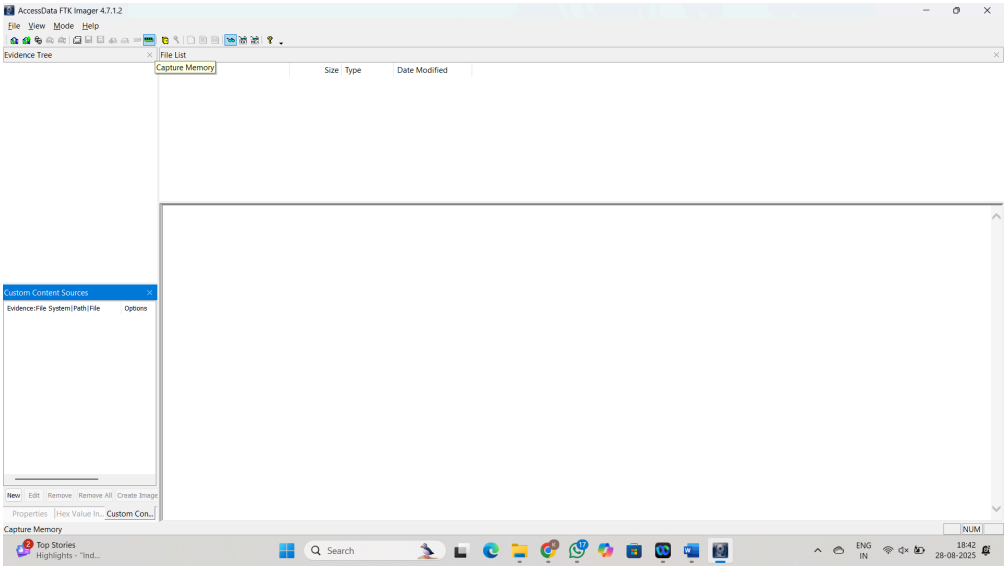
Step 1: Launch FTK Imager as Administrator

- Right-click on **FTK Imager** and select **Run as Administrator**.
- This ensures the tool has sufficient privileges to access system memory.



Step 2: Open Capture Memory Utility

- Below the **Menu Bar**, click **Capture Memory...**

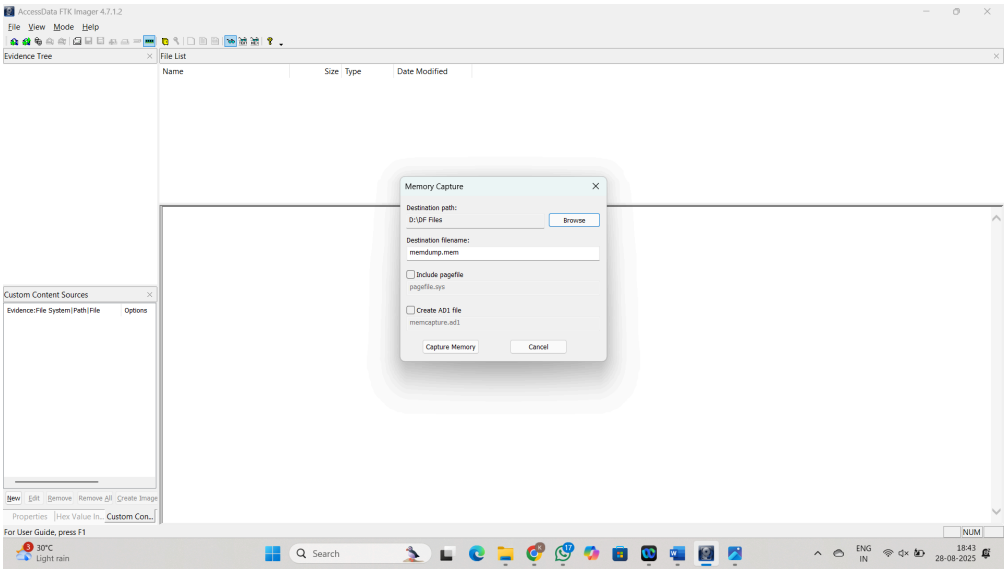


Step 3: Configure Capture Options

In the pop-up dialog:

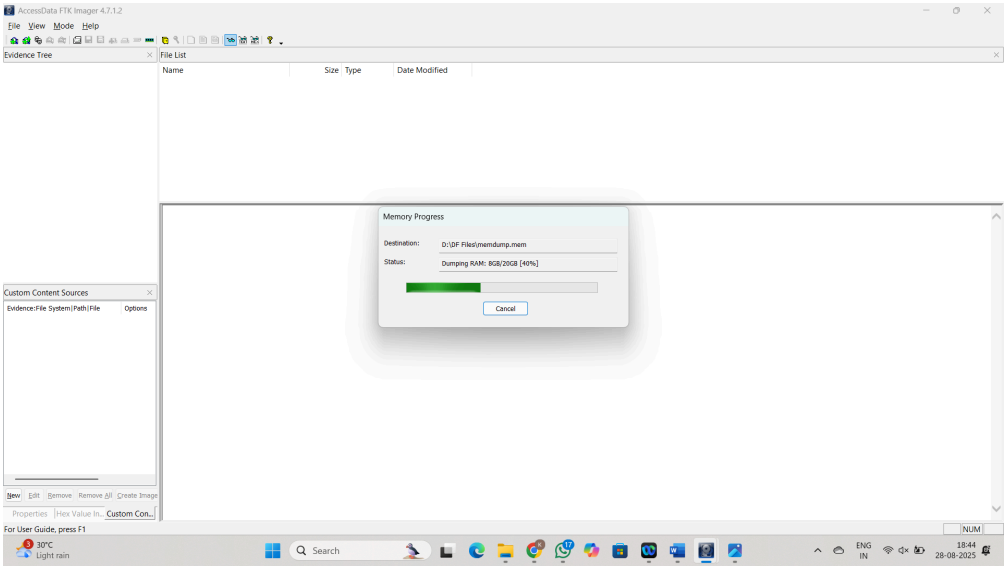
- **Destination Path** → Choose an **external drive** (not the system drive).
- **Destination Filename** → Default is `memdump.mem` (rename if required).
- **Include Pagefile.sys (Optional)** → Captures virtual memory stored on disk.
- **Create AD1 File (Optional)** → Wraps output into an AccessData container.

> *Tip:* Including `pagefile.sys` can reveal hidden processes and artifacts.



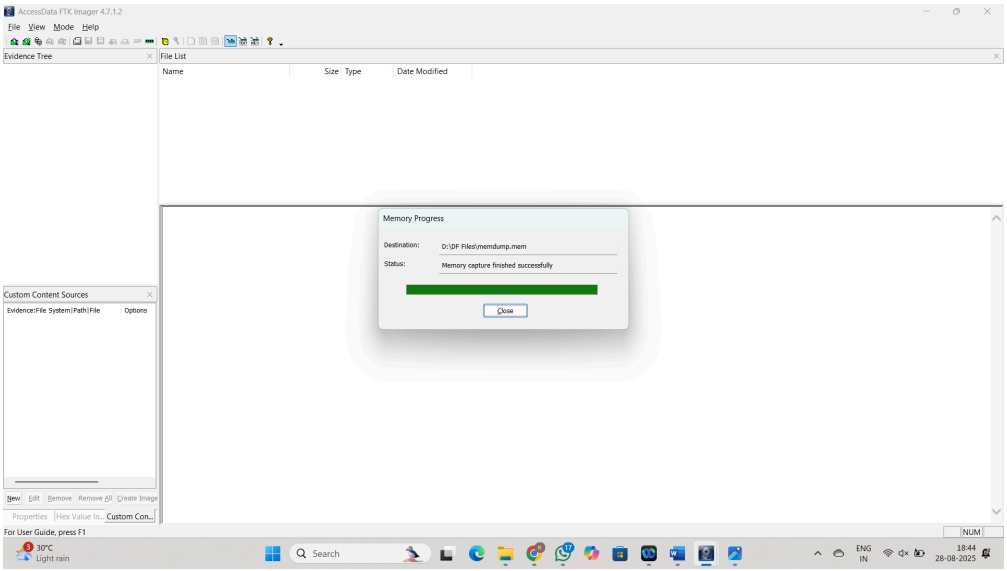
Step 4: Begin Capture

- Click **Capture Memory** to start.
- A progress bar shows the status.



Step 5: Completion

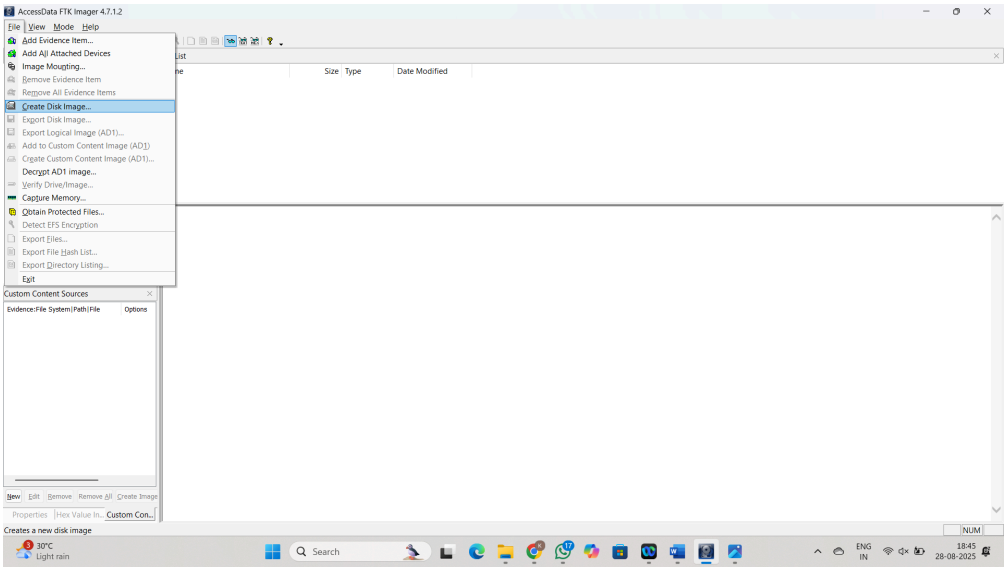
- The `.mem` file will be created in the destination folder.
- Capture time depends on installed RAM size.



Acquiring Non-Volatile Memory (Disk Image)

Step 1: Start Disk Imaging

- In FTK Imager, go to **File → Create Disk Image...**

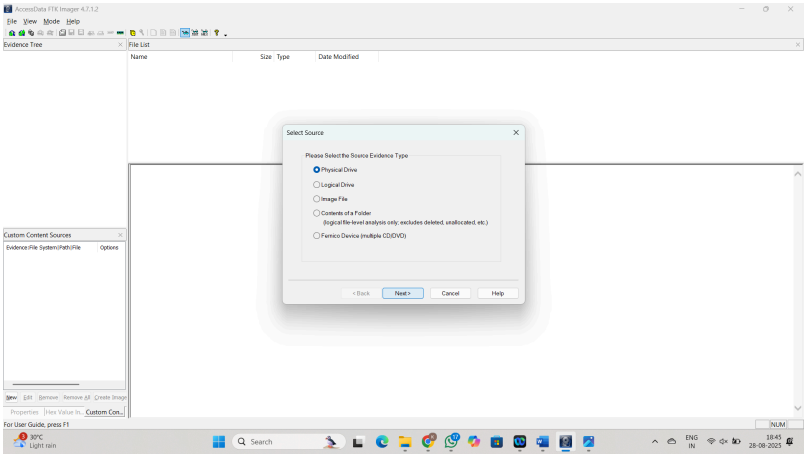


Step 2: Select Source Type

Choose based on requirement:

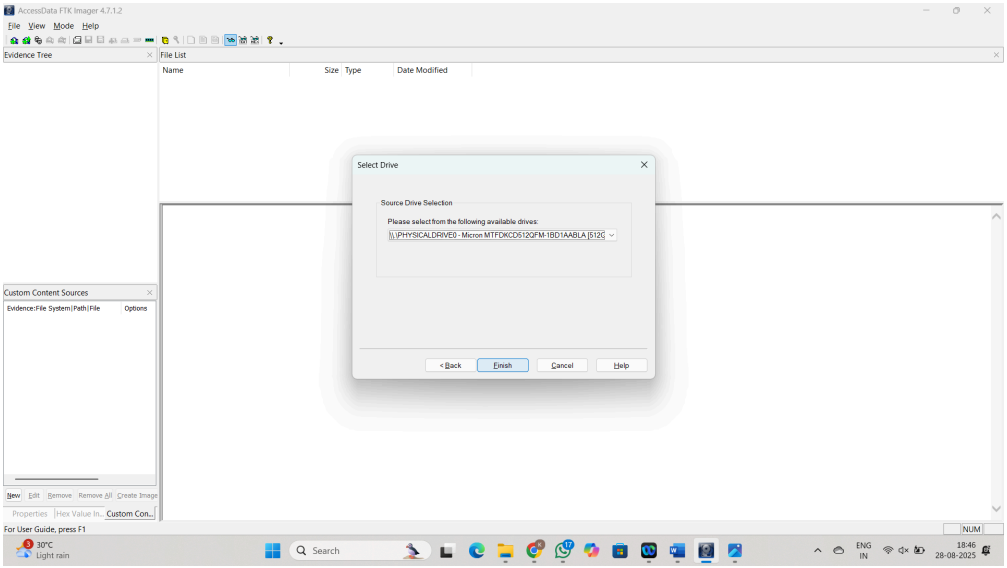
- **Physical Drive** → Entire disk (preferred).
- **Logical Drive** → Single partition (e.g., C:).
- **Image File** → Re-image an existing file.
- **Folder / CD/DVD** → Acquire folder or removable media.

> *Forensic best practice:* Always select **Physical Drive** with a **write blocker**.



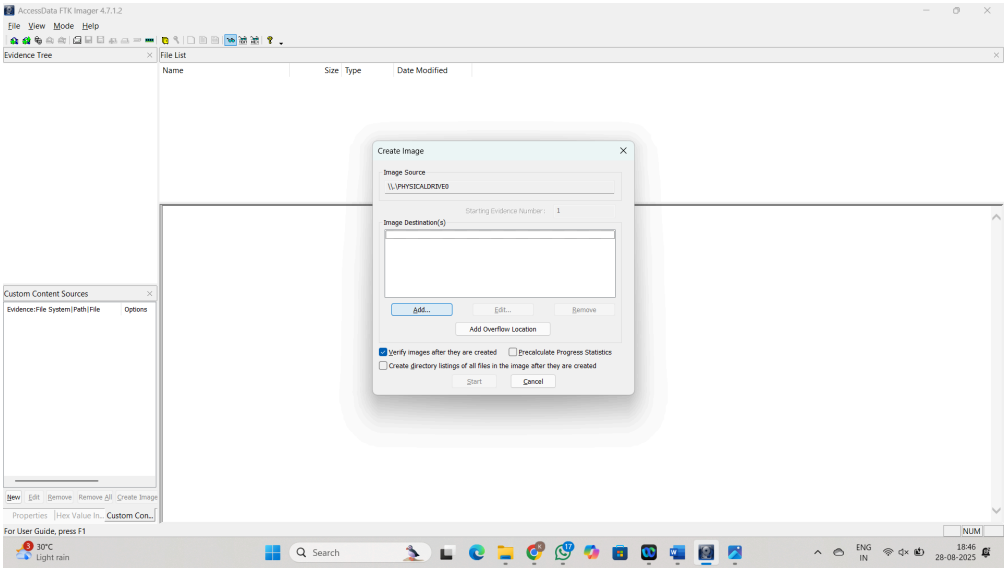
Step 3: Select Drive

- Pick the drive from the list.
- Confirm and click **Finish**.

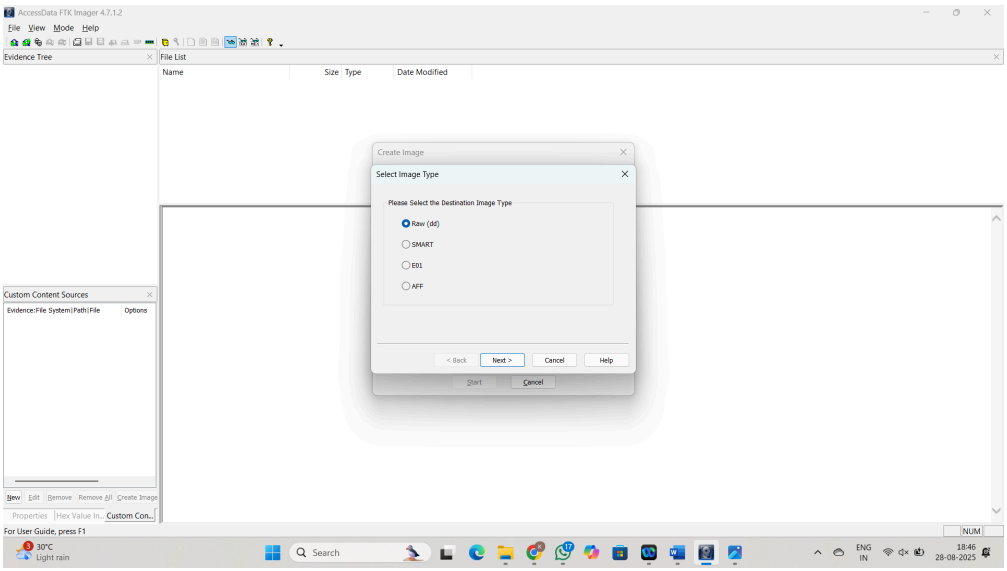


Step 4: Configure Destination & Format

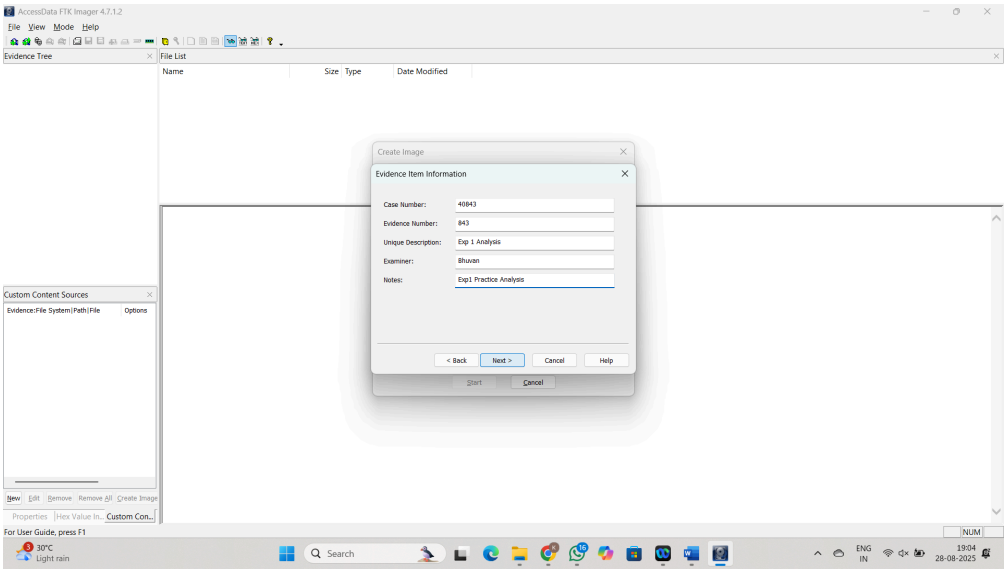
- Click **Add...** to define image format and storage path.



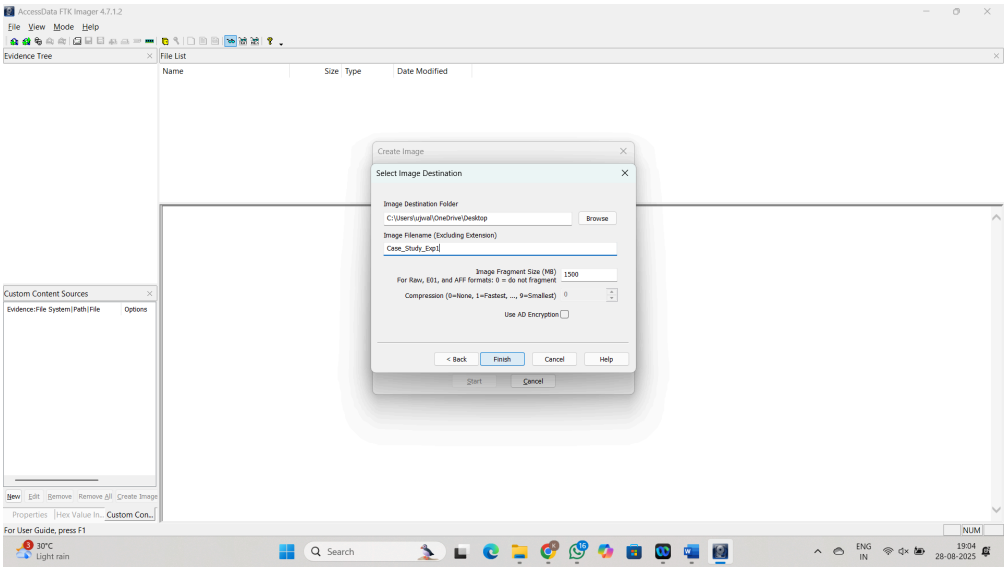
- **Image Type:**
  - `E01 (EnCase)` → Recommended (metadata + compression).
  - `RAW (dd)` → Bit-for-bit copy.



- Enter **Case Info**: Examiner, Case No., Notes.

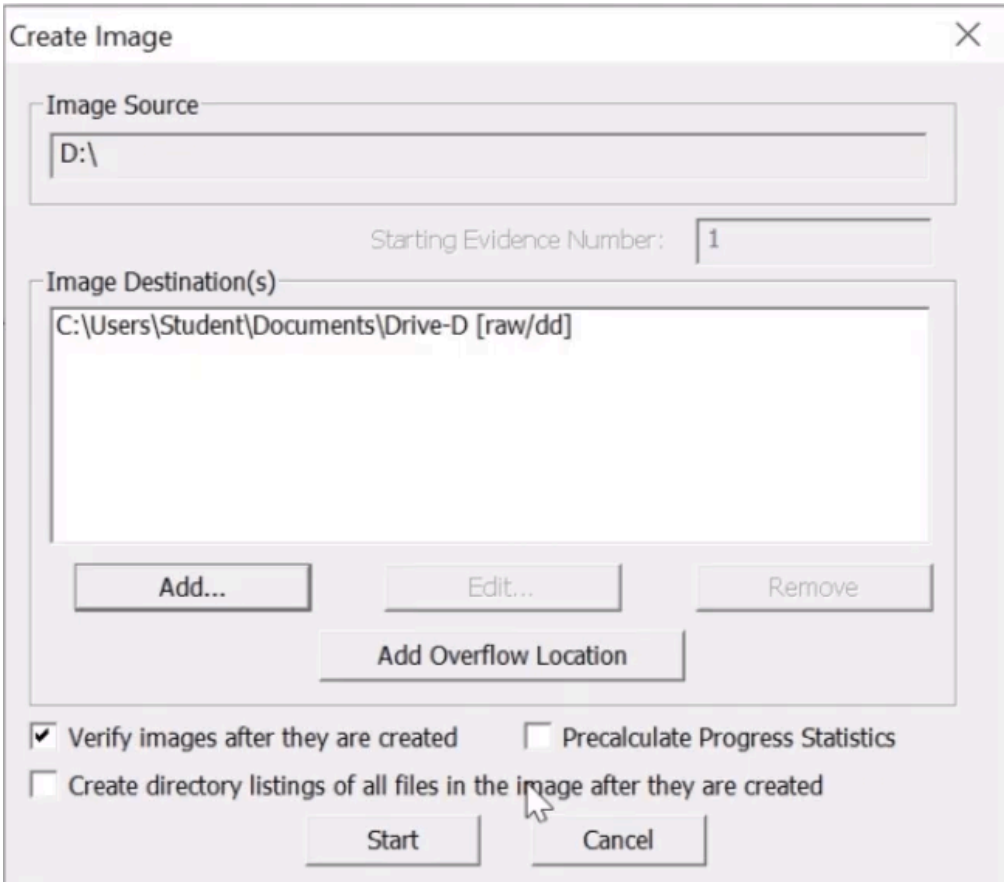


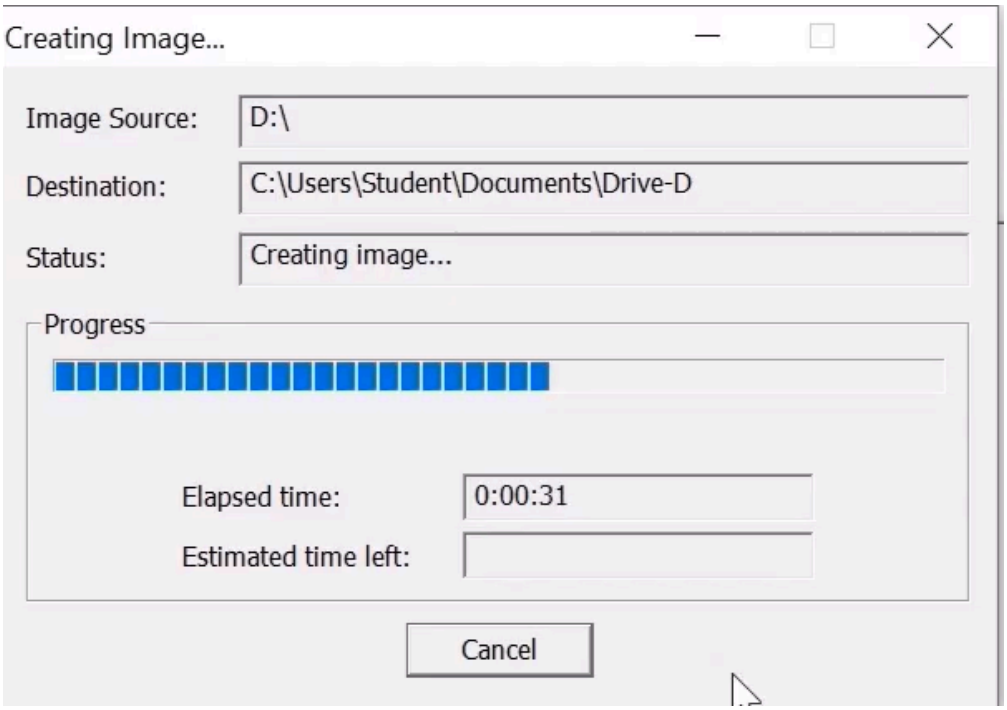
- Set **Destination Folder** (different from source).
- **Fragment Size:** Set `0` for a single file.



Step 5: Imaging Process

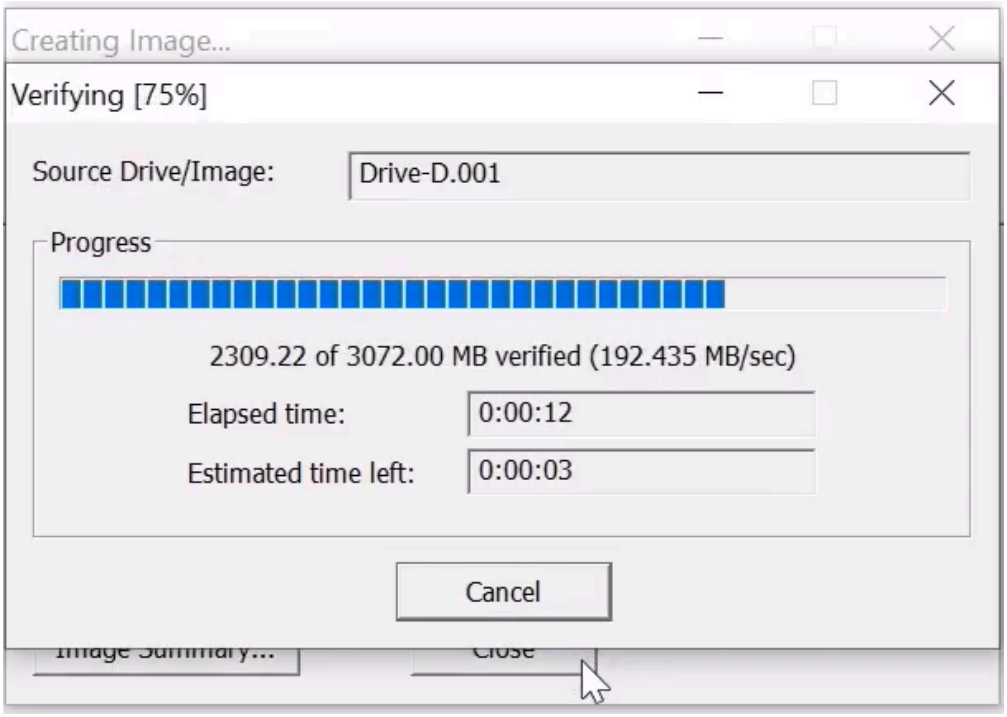
- Check **Verify images after creation** to generate hash values.
- Click **Start** to begin acquisition.

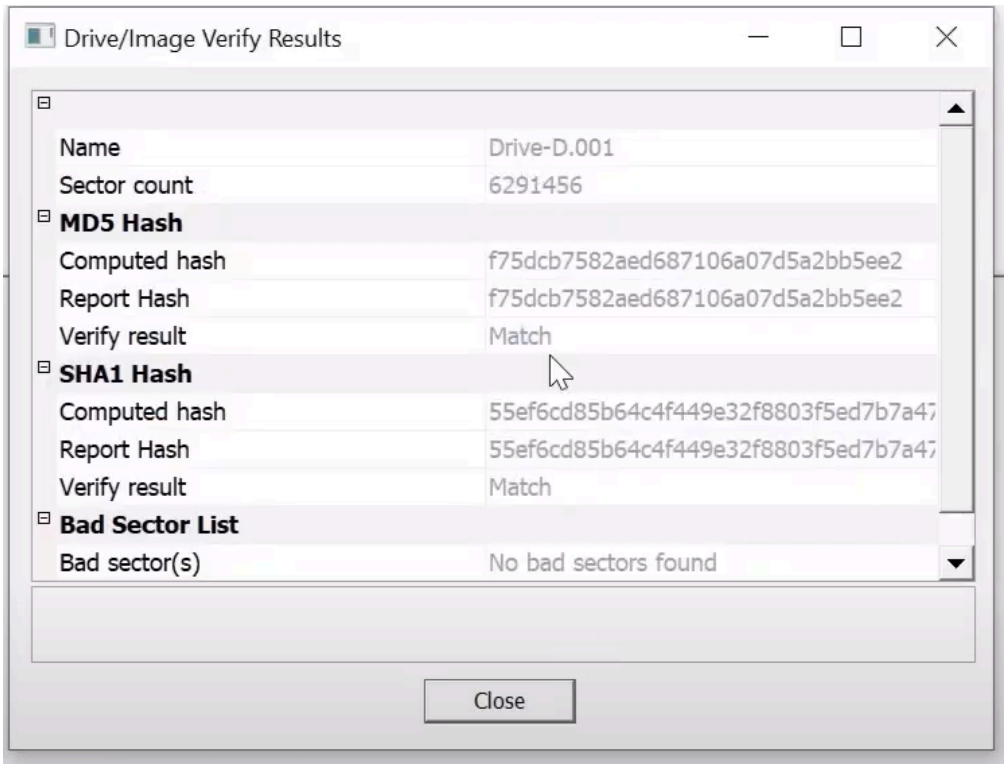




Step 6: Verify Integrity

- On completion, FTK Imager displays **MD5/SHA1** hashes.
- If hashes match, the image is valid and unaltered.





Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	*10*	

Result

Successfully acquired the **RAM dump (.mem)** and **disk image (.E01)** of the target system using **FTK Imager**.
The **MD5/SHA1 hash values** of the acquired images were verified, confirming that the evidence was collected without alteration and is **forensically sound**.

Bhuvaneshwar-Naidu Update Exp_2_Test Disk.md c09b582 · 1 minute ago

123 lines (90 loc) · 4.83 KB

Ex.No.2 Recover Deleted or Damaged Files using TestDisk

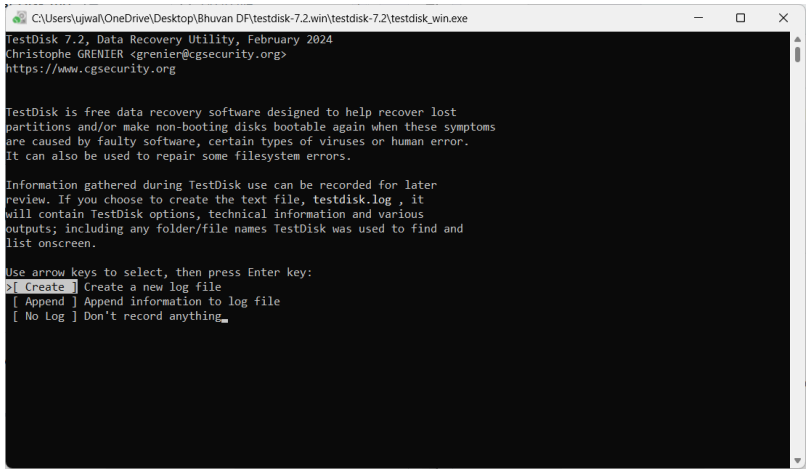
Aim

To use **TestDisk** step-by-step to recover a missing partition, repair a corrupted partition, and restore access to lost files.

Step 1: Log Creation & Disk Detection

Log Creation

- When TestDisk starts, Select the [Create] option to generate a log file of the recovery session. This is helpful for future reference or troubleshooting.



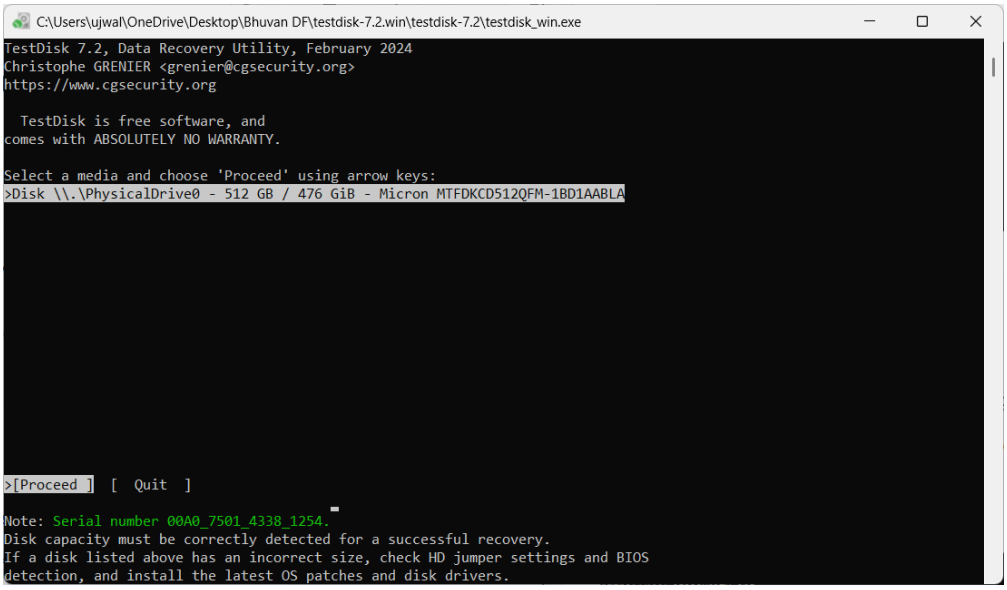
Disk Detection

- All hard drives will be listed with their correct sizes.

Use the **Up/Down arrow keys** to select the target disk.

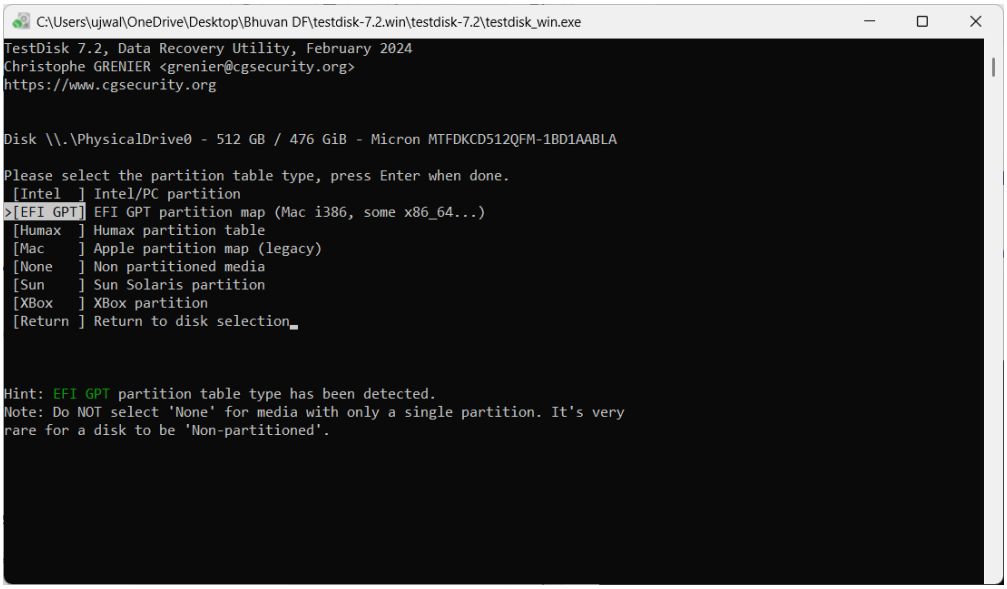
> If available, prefer `/dev/rdisk*` (raw device) over `/dev/disk*` for faster performance

- Select [Proceed] to move to the next step.



Step 2: Partition Table Type Selection

- TestDisk auto-detects the partition table type.
- Usually, the **default value is correct**.
- Press **Enter** to proceed.

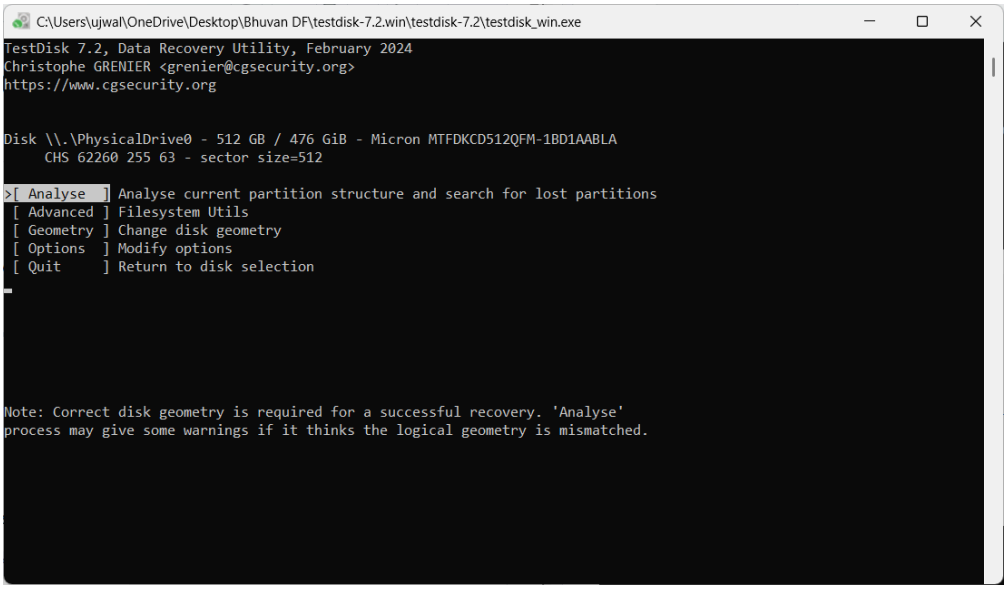


Step 3: Analyse Partition Structure

- Select **Analyse** from the menu to view the current partition structure.
- Missing or corrupted partitions will be shown here.

> Example issues:

- A partition listed twice → indicates corruption.
- "Invalid NTFS boot" → damaged NTFS boot sector.
- Missing logical partition(s).
- Press **Enter** to proceed to **Quick Search**.

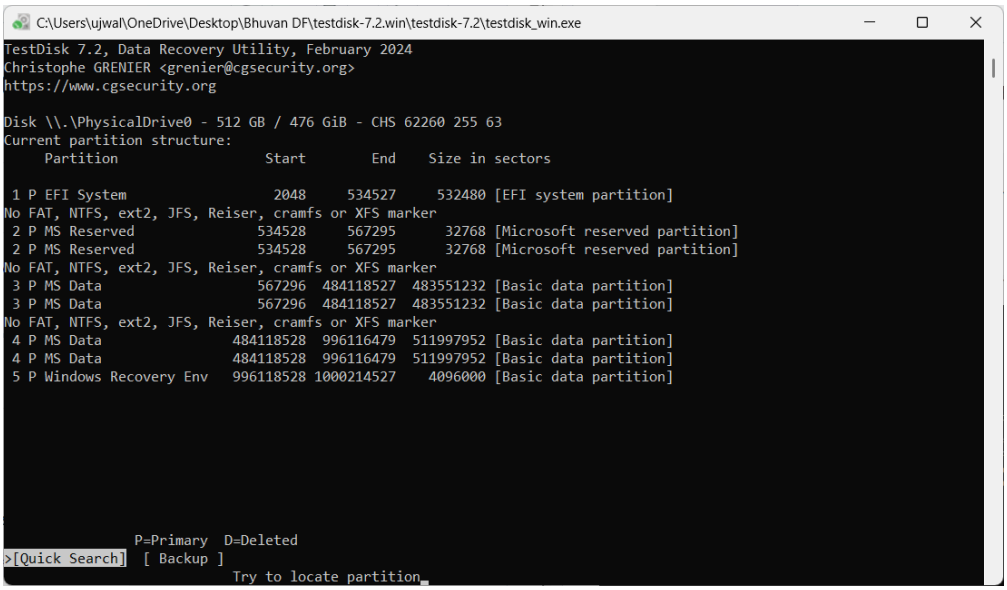


Step 4: Quick Search for Partitions

- TestDisk performs a **Quick Search** and lists found partitions in real-time.
- Highlight the missing partition and press **p** to list its files.

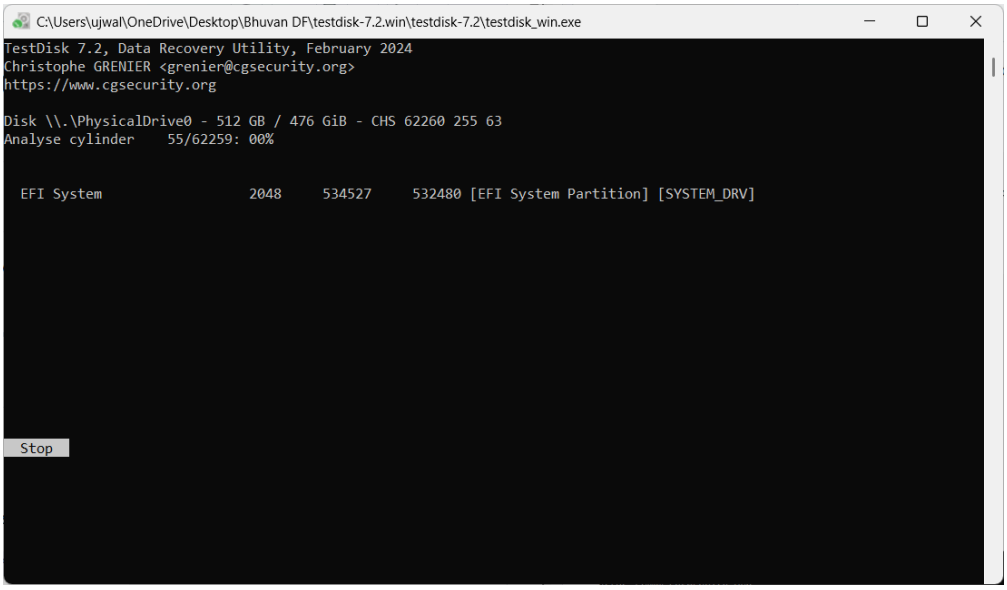
> Files in **red** are deleted entries. Use **q** to go back.

- If all looks correct, press **Enter** to continue.



Step 5: Save Partition Table / Deeper Search

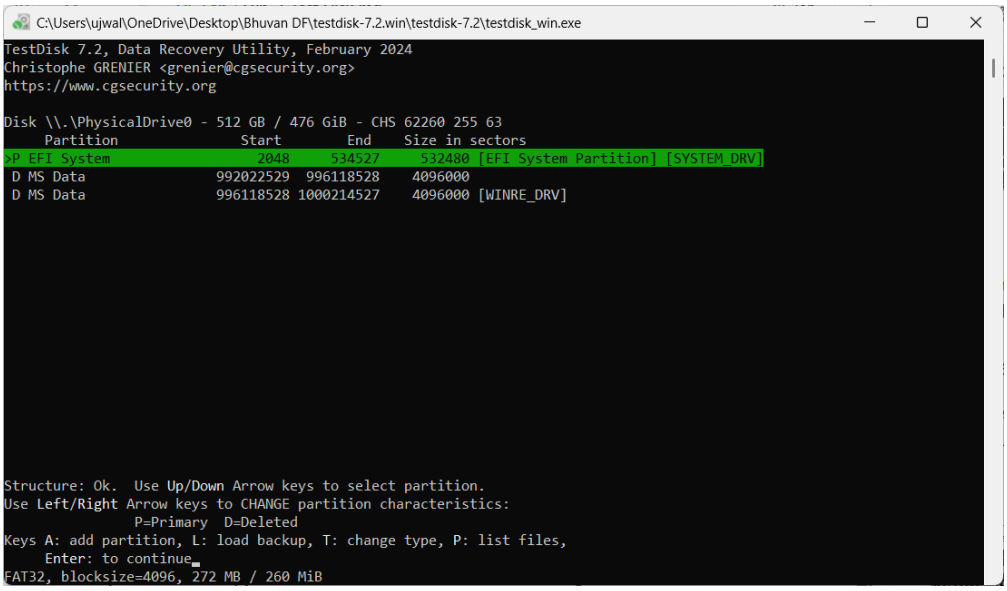
- If not all partitions are visible, select **Deeper Search**.
- This scans for backup boot sectors (FAT32, NTFS, ext2/ext3) cylinder by cylinder.
- This process can take a long time, as it scans the entire drive, block by block, to find remnants of partition structures.
- Again, use p to preview files and confirm if a found partition is the one you are looking for.



After the deeper scan:

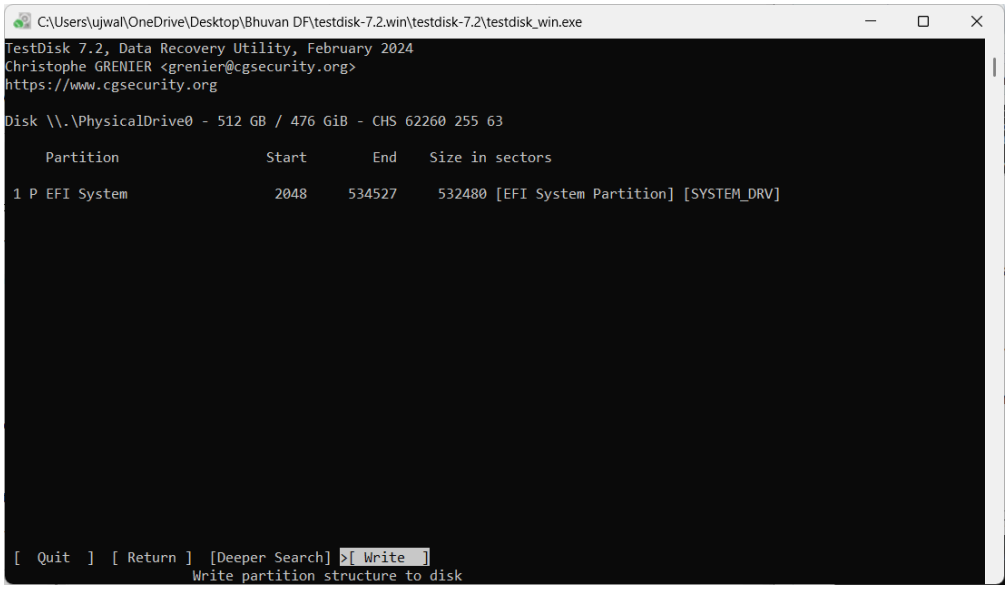
- Partitions found using backup boot sectors are listed.
- Overlapping or corrupted entries will appear as **D (Deleted)**.
- Highlight the correct partition and press **p** to verify its files.

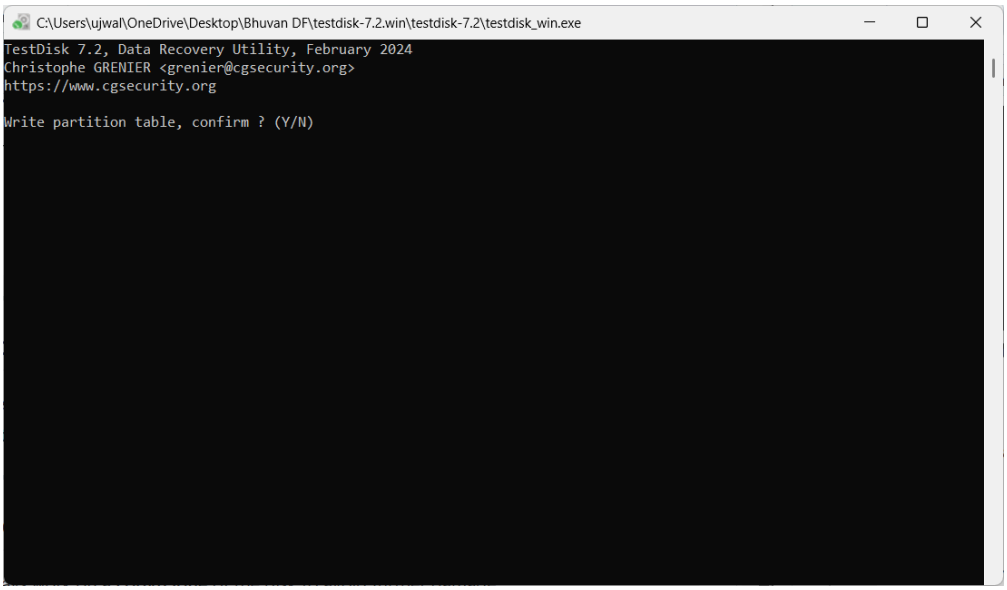
- Use **Left/Right arrow keys** to change partition status:
  - `P` → Primary
  - `*` → Bootable
  - `L` → Logical
  - `D` → Deleted



Step 6: Partition Table Recovery

- Once correct partitions are marked:
  - Confirm with **Write** → press **Enter**, then `y`, then **OK**.
- TestDisk updates the partition table automatically.





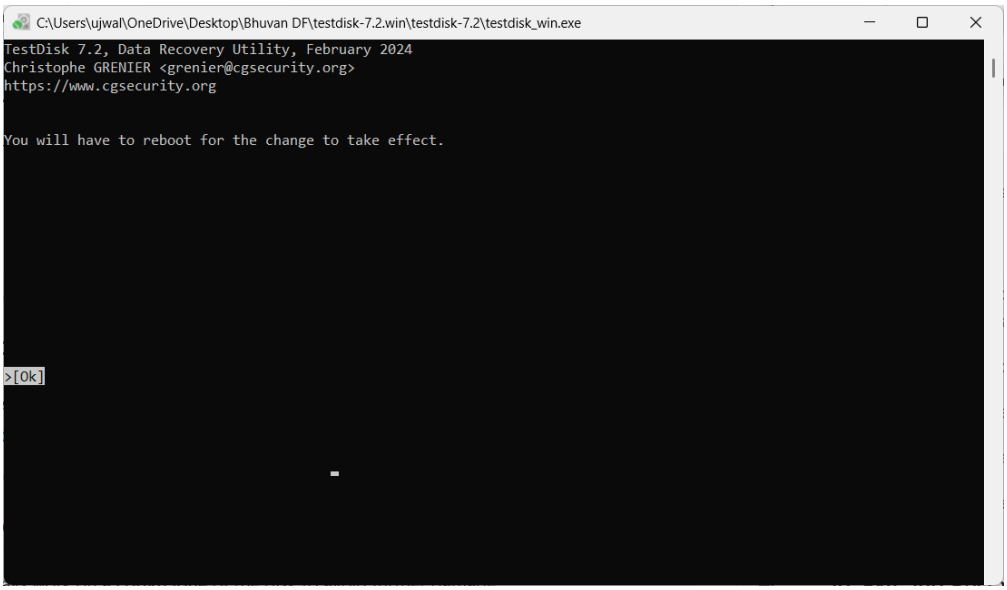
Step 7: NTFS Boot Sector Recovery

- If NTFS boot sector is damaged:
  - Select **Backup BS** to copy the backup boot sector over the bad one.
  - Confirm with **y** → then **OK**.

Now the boot sector and backup are identical, meaning recovery succeeded.

Step 8: Restart System

- After successful recovery, TestDisk prompts you to **reboot the computer**.
- Restart and check if your partitions and files are accessible again.



Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Successfully acquired the **RAM dump (.mem)** and **disk image (.E01)** of the target system using **FTK Imager**.
The **MD5/SHA1 hash values** of the acquired images were verified, confirming that the evidence was collected without alteration and is **forensically sound**.

Bhuvaneshwar-Naidu Update Exp_3_Wire Shark.md d6535de · 1 minute ago

87 lines (63 loc) · 2.66 KB

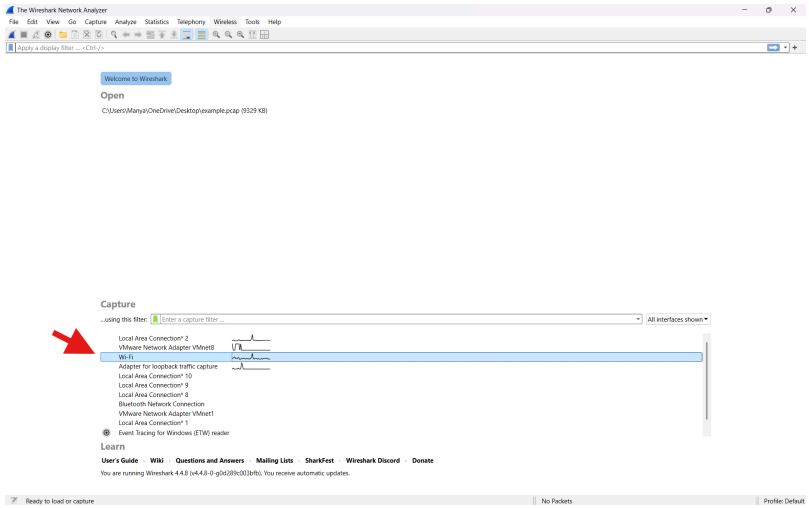
Ex.No.3 Wireshark – Network Packet Capture and Analysis Tool

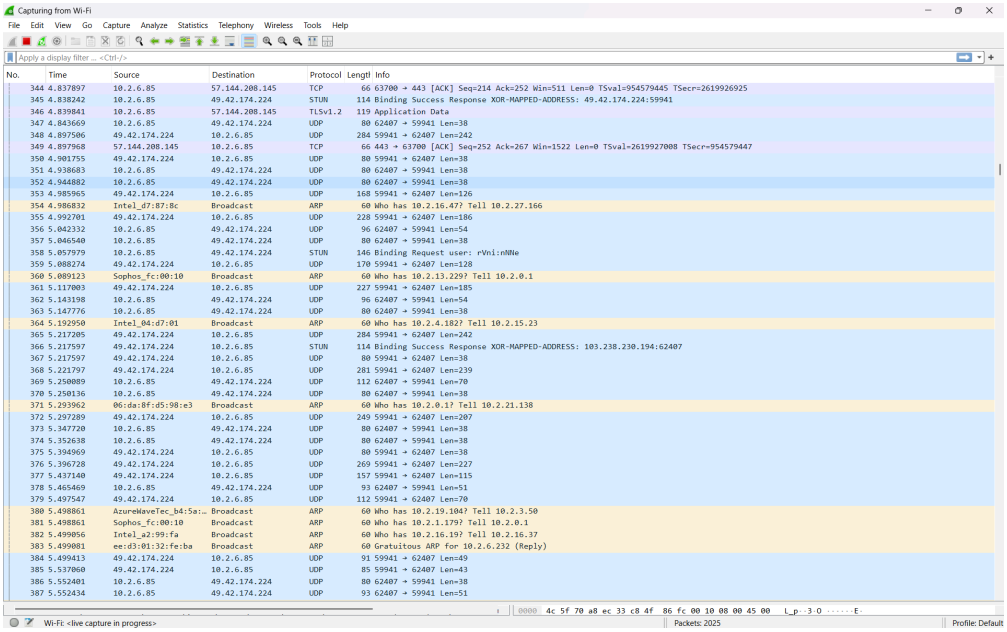
Aim

To capture plaintext **login credentials** transmitted over HTTP using **Wireshark**, and analyze how insecure protocols expose sensitive information.

Step 1: Start Capturing Packets

- Open **Wireshark** in your Windows/Linux machine.
- Select the active network interface (e.g., **Wi-Fi**).
- Click the **blue shark fin** icon to begin capturing packets.



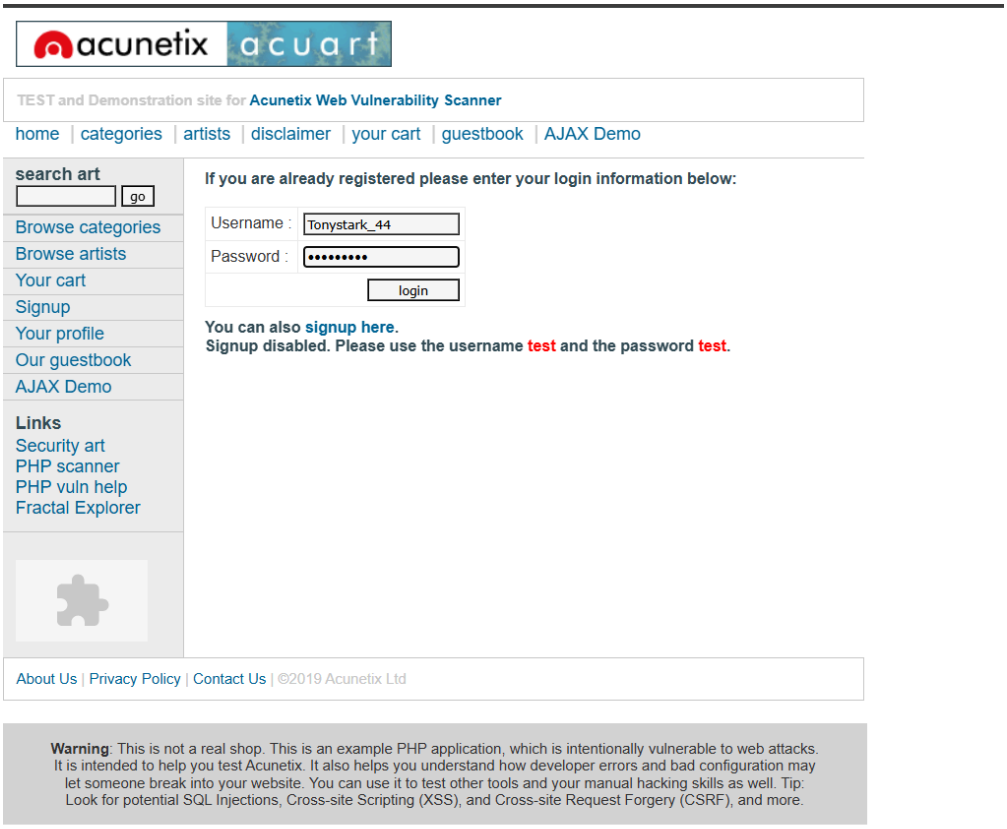


Step 2: Generate Login Traffic

- Open a browser and navigate to a test login page (e.g., `http://testphp.vulnweb.com/login.php`).
- Enter dummy credentials. For this example:

  Username: Tonystark_44

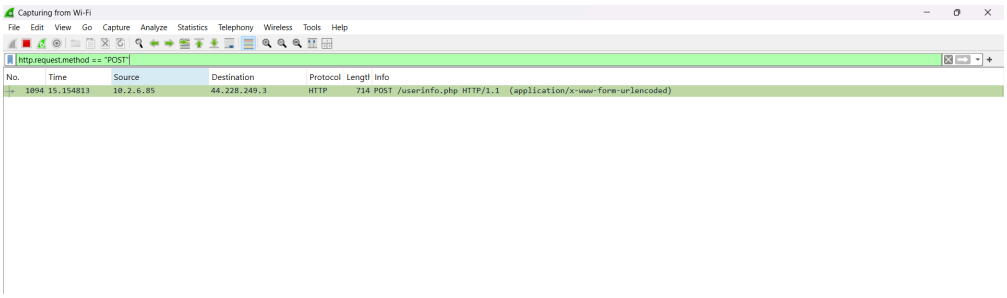
  Password: tony@1234
- Submit the form.
- Even if the login fails, the credentials are **transmitted** in the request.



Step 3: Stop Capture & Filter HTTP Traffic

- Stop the capture (click the **red square** button).
- In the display filter bar, type the following filter and press Enter:

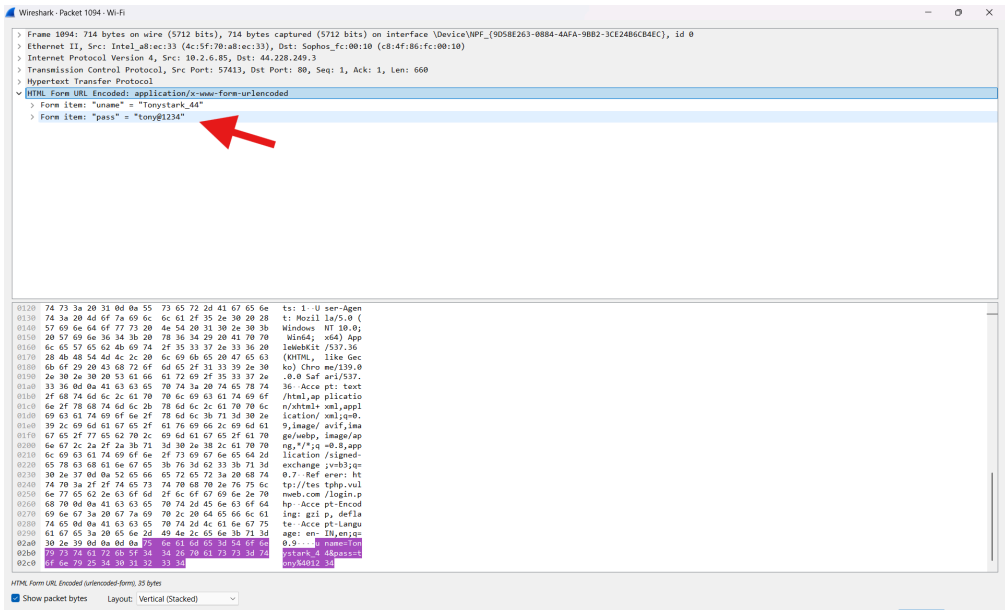
```
http.request.method == "POST"
```



Step 4: Inspect the POST Packet

- From the filtered list, select the POST packet.
- Expand the following sections in the Packet Details Pane:
  - ->Hypertext Transfer Protocol
  - ->HTML Form URL Encoded

You will see the submitted credentials in plaintext: Form item: "uname" = "Tonystark_44" Form item: "pass" = "tony@1234"



Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

The experiment successfully captured **login credentials** transmitted via **HTTP**.
This demonstrates that **HTTP is insecure**, as sensitive information is sent in **plaintext**, making it easy for attackers to intercept.

Ex. No 4: Analyze Email Headers and Detect Email Spoofing using MHA (Mail Header Analyzer)

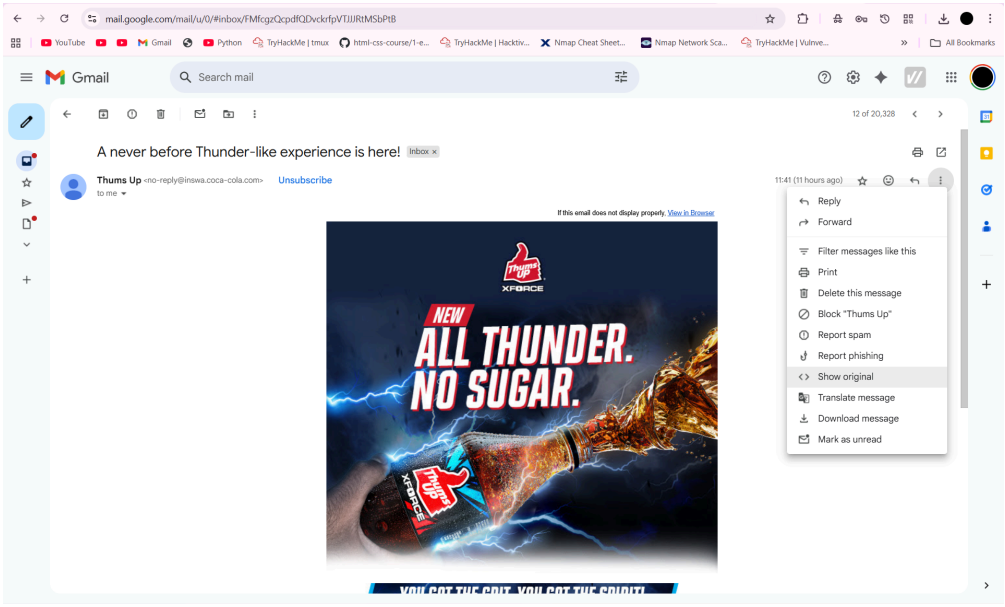
Aim

To analyze an email header and detect possible **email spoofing** using the **Mail Header Analyzer (MHA)** tool.

Steps

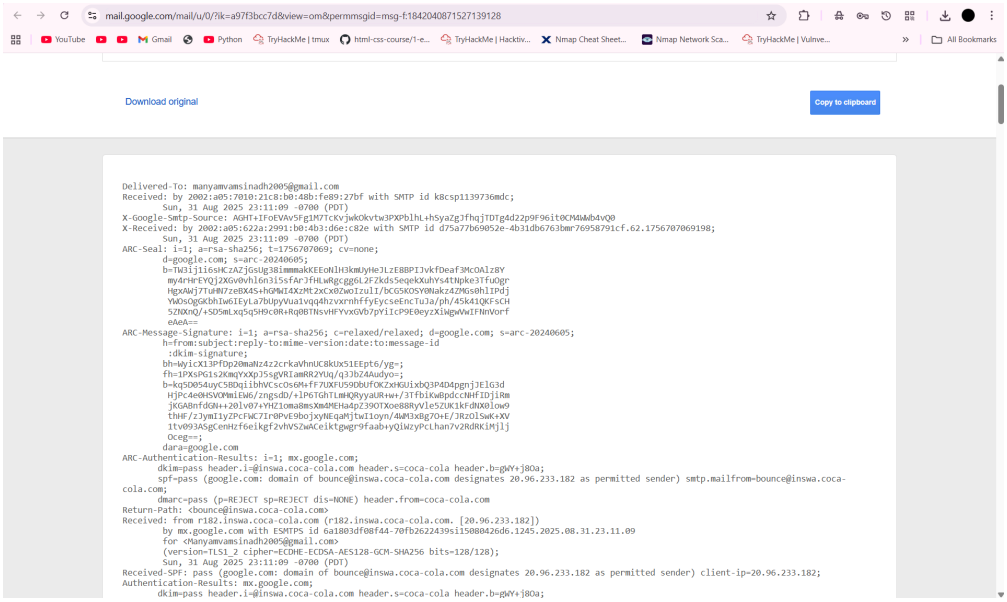
Step 1: Access the Email Header

- **Gmail**: Open the email → Click the three dots (More) → Select **Show Original**
- **Outlook**: Open the email → Click on **File** → **Properties** → Copy from **Internet headers**
- **Yahoo**: Open the email → Click the three dots (More) → Select **View Raw Message**



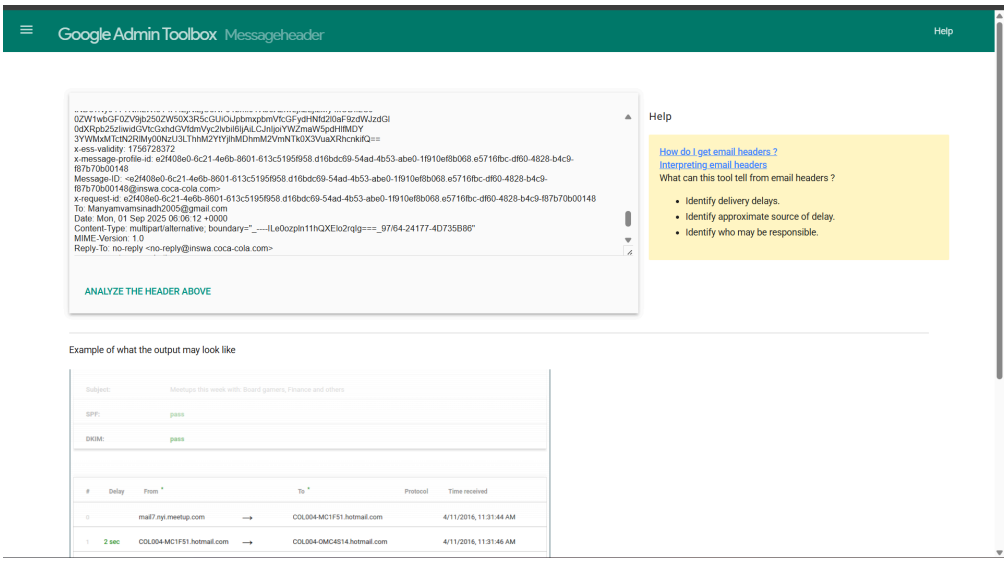
Step 2: Copy the Email Header

- Copy the entire header text (contains details like `Received`, `Message-ID`, `SPF`, `DKIM`, etc.)
- Save it into a text file or keep it ready to paste into an analysis tool.



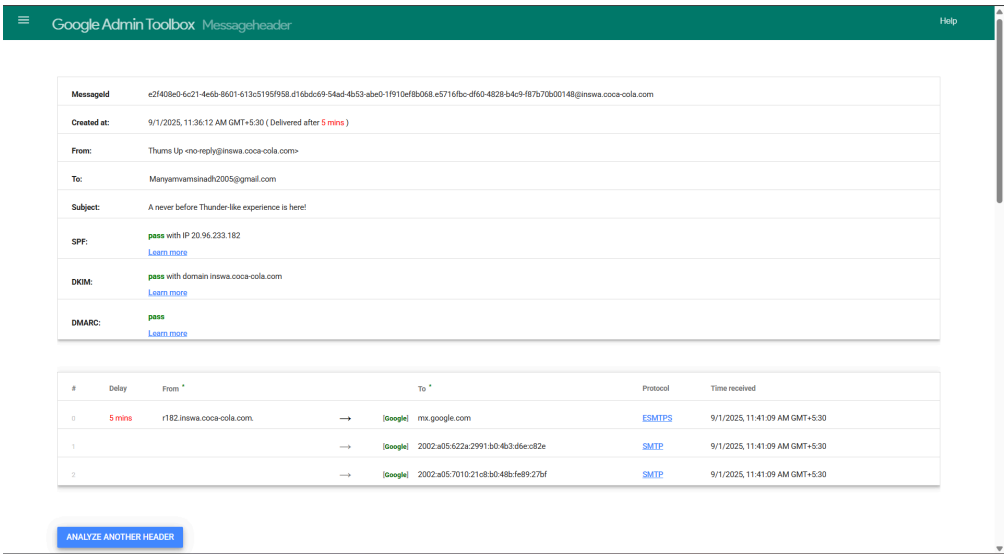
Step 3: Use Mail Header Analyzer (MHA)

1. Open [Google Admin Toolbox MHA]
2. Paste the copied email header into the text box.
3. Click **Analyze the Header**.



Step 4: Interpret the Results

- **Received Chain** → Shows the exact path (servers & timestamps) the email took.
    - The email went through the following servers:
        - **r182.inswa.coca-cola.com → Google Mail Server (mx.google.com)**, taking **5 minutes** for delivery.
        - Further SMTP relays from Google IP addresses.
- **Delay Analysis** → Highlights unusual delays that might indicate suspicious routing.
    - There is a delay of **5 minutes** before the email was received by the final server. This isn't too unusual, but it’s worth considering depending on the context of the email delivery.
- **SPF/DKIM/DMARC** → Shows authentication results.
    - **SPF**: **Pass** → Email was sent from an authorized IP (20.96.233.182).
    - **DKIM**: **Pass** → The email domain (inswa.coca-cola.com) has verified the message's integrity.
    - **DMARC**: **Pass** → The email aligns with the domain's policy.
- **Message-ID** → Should match the sender’s domain; mismatch may indicate forgery.
    - The **Message-ID** appears legitimate and corresponds to the sender's domain (inswa.coca-cola.com), meaning it's unlikely to be forged.



Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result:

Successfully analyzed the email header. All authentication checks (SPF, DKIM, DMARC) **passed**, indicating the email is **genuine**. The **delay** is not unusually long and doesn't suggest suspicious activity.

Ex. No 5: Use Autopsy to Create a Case and Import Evidence

Aim

To perform a forensic investigation using **Autopsy**, by creating a case, importing evidence, analyzing artifacts, and generating a forensic report.

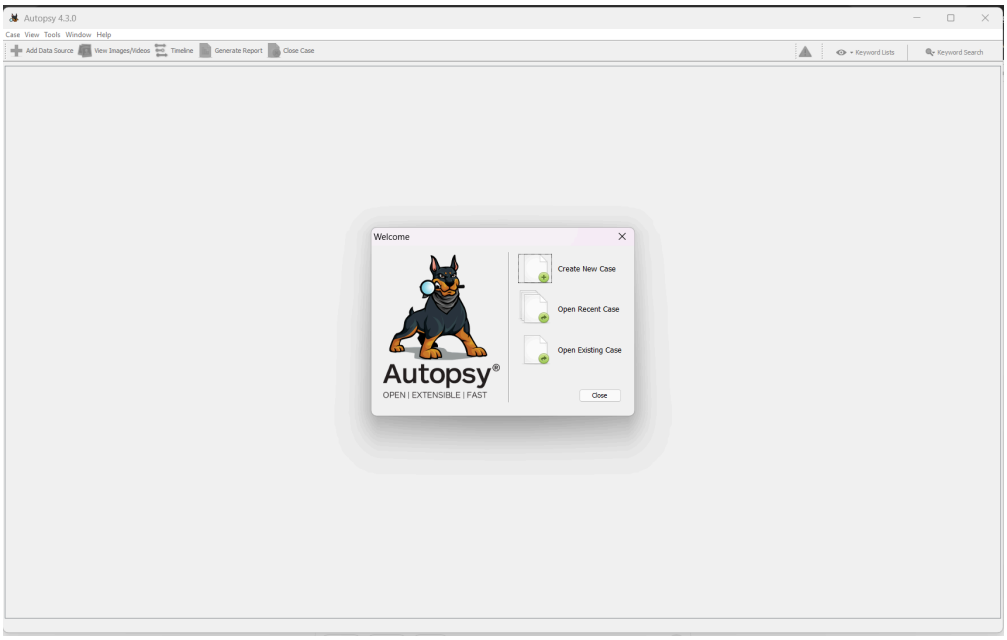
Steps

Step 1: Installation

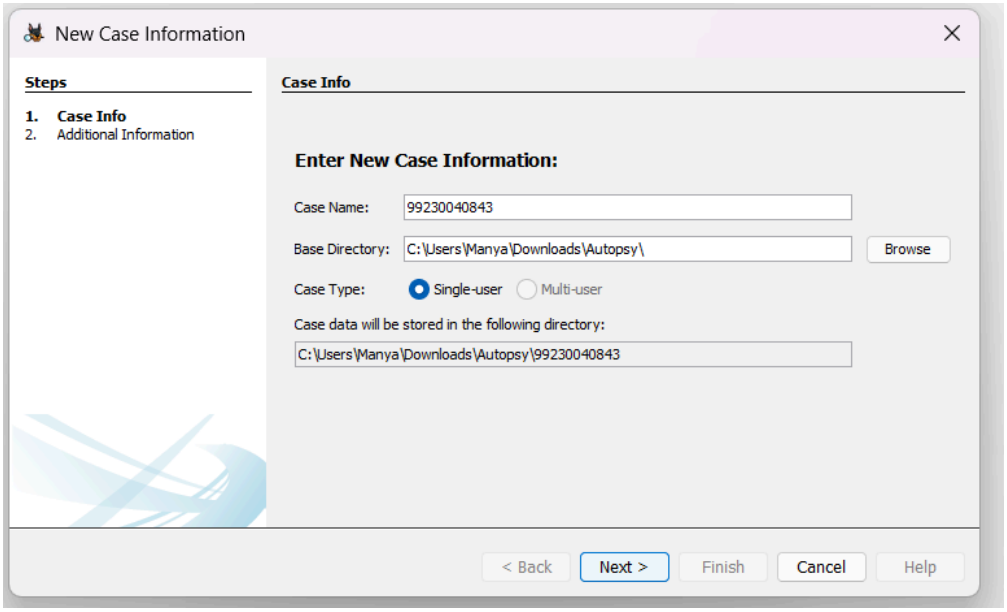
- **Download & Install** Autopsy from the official website.
- Follow installation instructions for your OS (Windows/Linux/macOS).

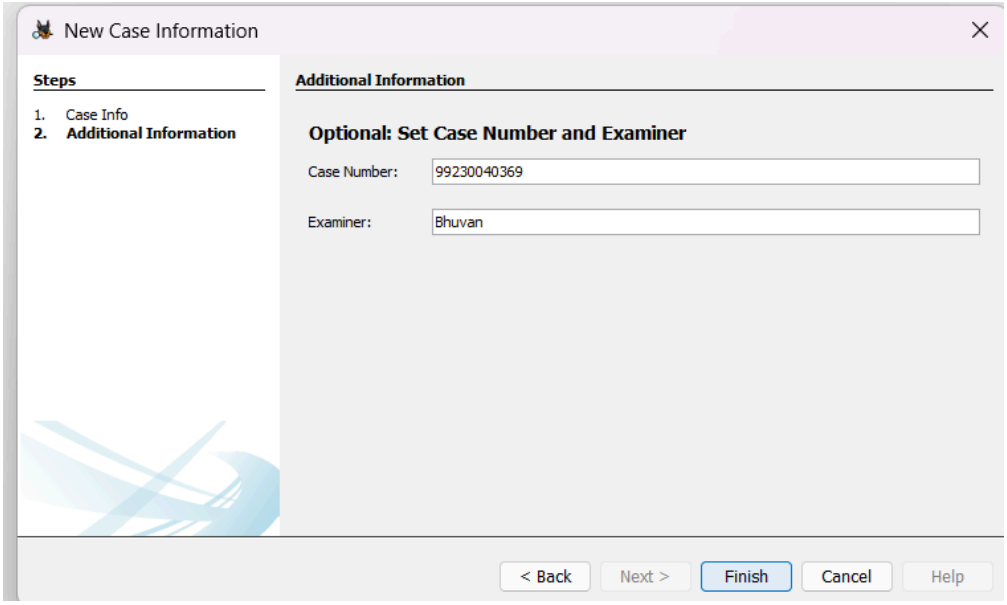
Step 2: Starting a New Case

- Open **Autopsy**.
- Click **New Case**.



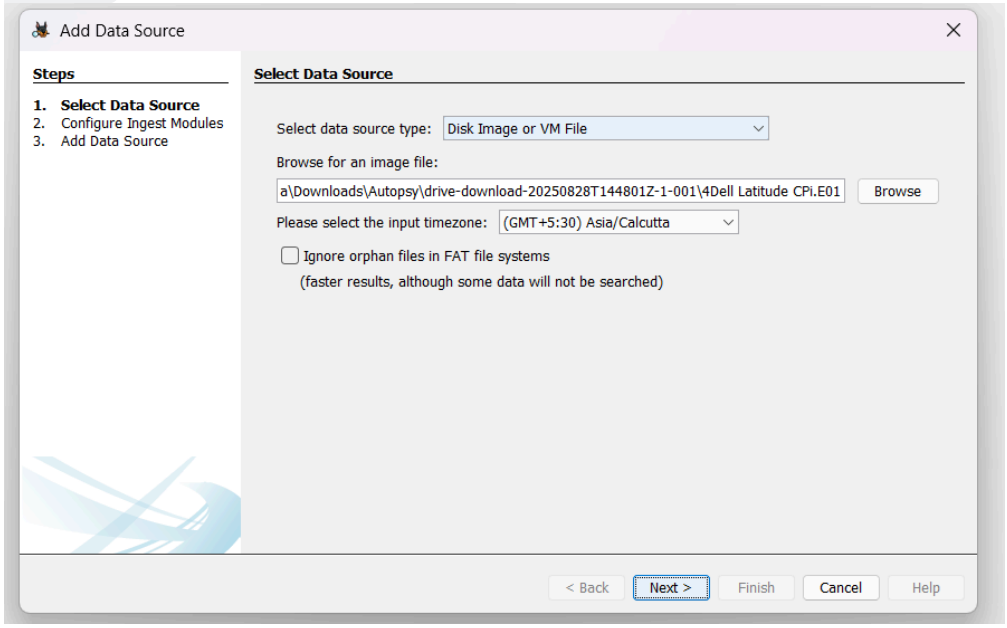
- Enter:
  - Case name
  - Case location (storage path)
  - Case number, examiner's name, etc.
- Click **Next** to proceed.





Step 3: Adding a Data Source

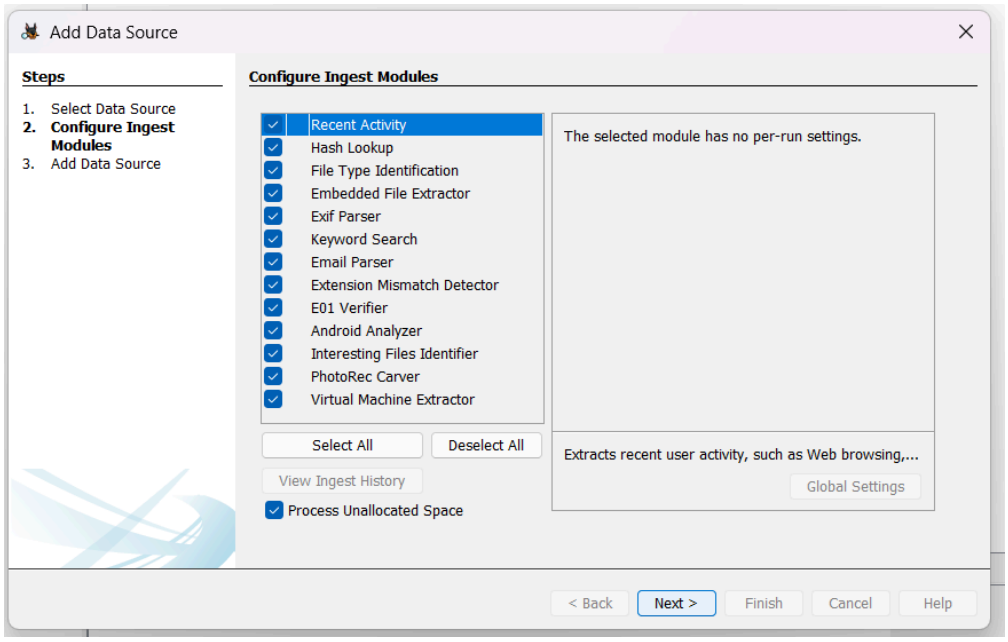
- After creating a case, Autopsy will prompt you to add a **data source**.
- Choose the source type:
  - Disk images (`.E01` , `.dd` , `.raw`)
  - Directories
  - Logical files
  - Local disks
- Browse and select the image file (e.g., `4Dell Latitude CPi.E01` , `4Dell Latitude CPi.E02`).



- **Configure Ingest Modules**:
  - File Type Identification
  - Keyword Search
  - Hash Lookup

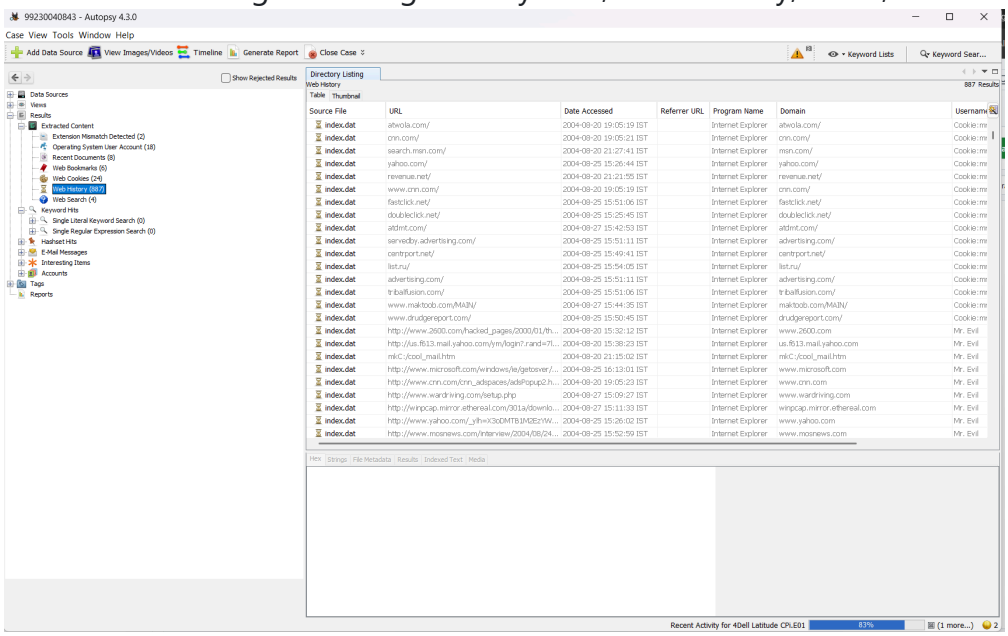
- (Enable/disable as per requirement)

- Click **Next** to start the analysis.



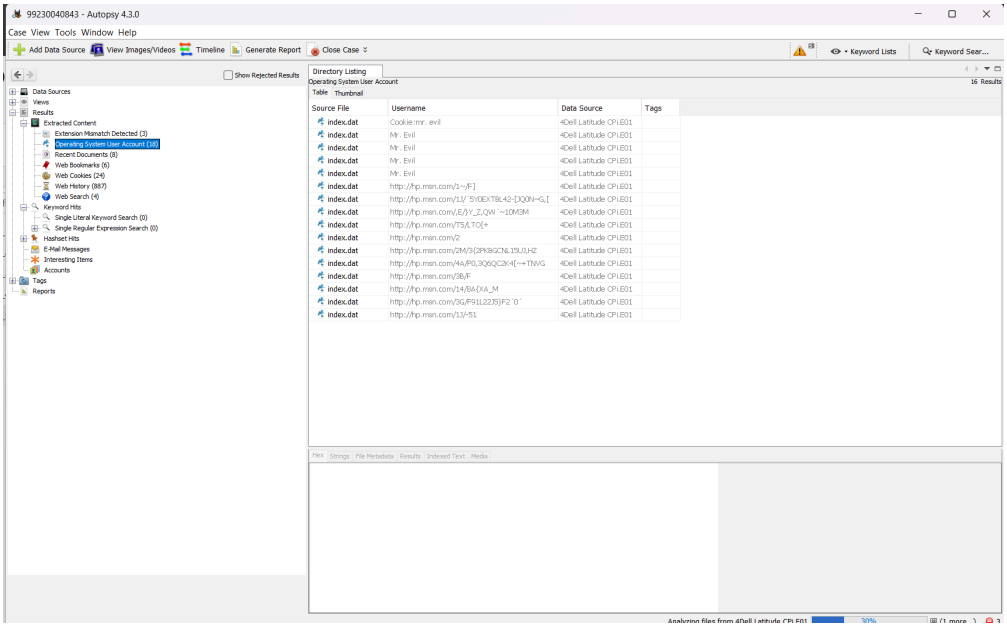
Step 4: Initial Analysis & Overview

- **Ingest Progress**: Progress is shown in the lower-left corner.
- **Artifacts Categorization**:
  - File system metadata
  - Web artifacts
  - Communication records
- **Tree Viewer**: Navigate through File System, Web History, Email, etc.



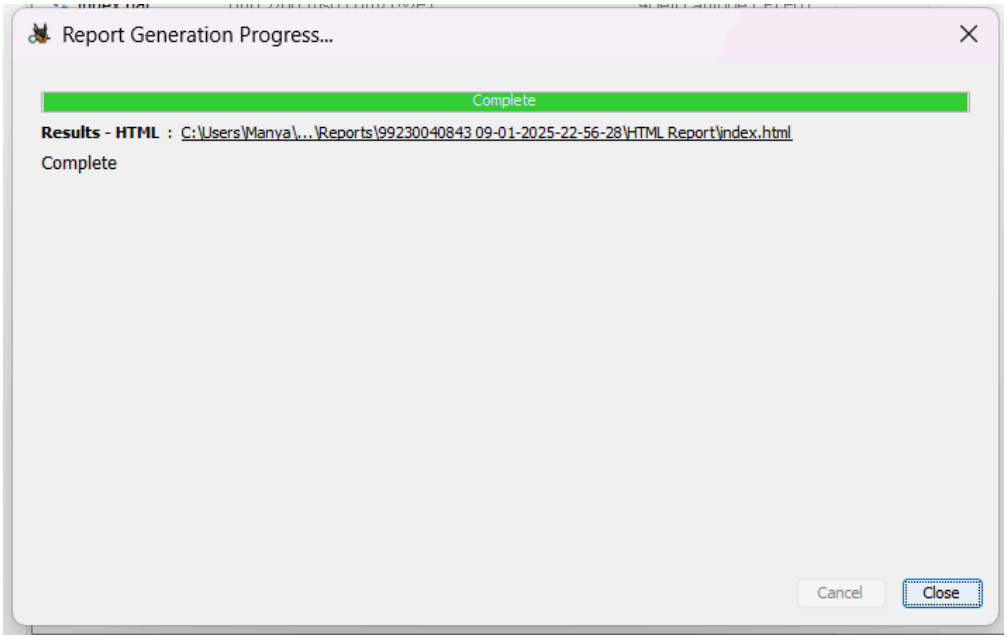
Step 5: Detailed Analysis

- **Keyword Search**: Use pre-configured or custom keyword lists.
- **File Analysis**: Open, preview, or export files.
- **Timeline Analysis**: Visualize user activity across time.
- **Hash Analysis**: Compare with known hash databases (good/bad files).



Step 6: Reporting

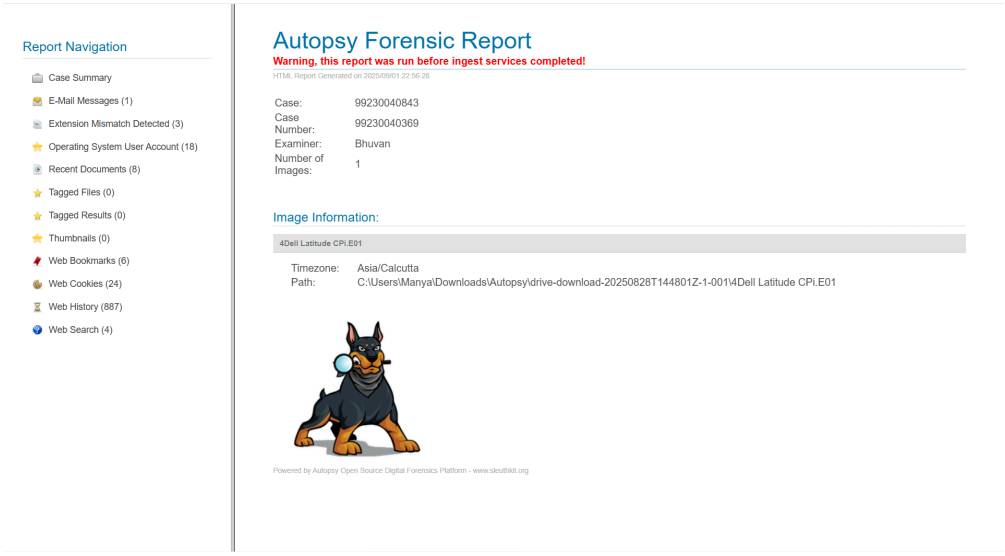
- Click **Generate Report** from the toolbar.
- Choose report type: HTML, CSV, Excel, etc.
- Select which parts of analysis to include.



- Export findings (files or artifacts). - Review and finalize the report.

Step 7: Case Closure

- Close the case in Autopsy.
- Archive data & reports as per organizational policy.



Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Successfully created a case in Autopsy, imported a forensic disk image, analyzed artifacts, and generated a forensic report.

Ex. No 6: Use Sleuth Kit to Analyze Digital Evidence

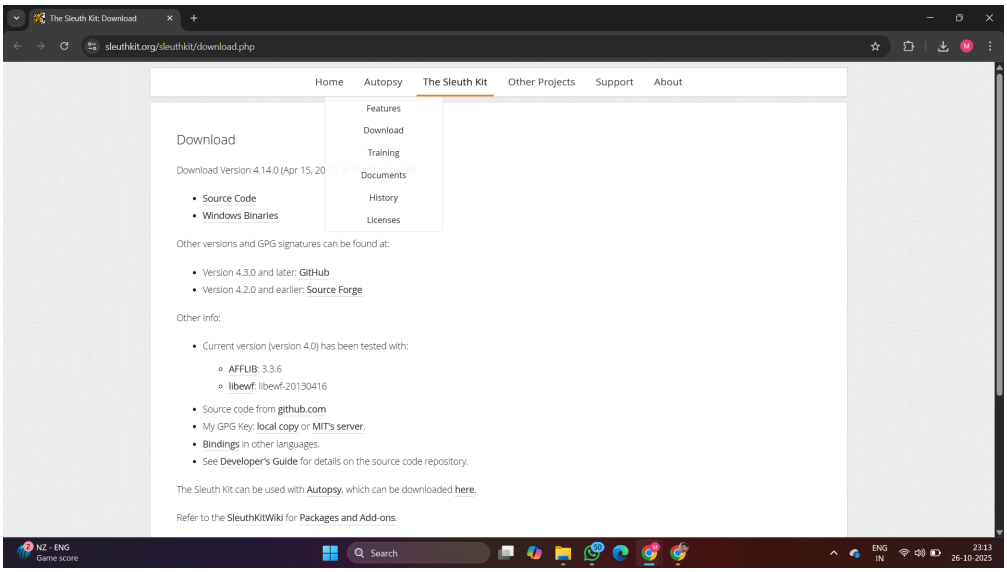
Aim

To analyze a disk image and recover digital evidence using **The Sleuth Kit (TSK)** command-line tools.

Steps

Step 1: Installation

- Download and install **The Sleuth Kit (TSK)** for Windows from its official website.
- Follow on-screen installation instructions.



Step 2: Acquire the Disk Image

- Create or obtain a forensic disk image using tools like **FTK Imager** or **dd**.
- Supported formats: `.dd` , `.raw` , `.img` , `.E01`
- Example evidence files:
  - `4Dell Latitude CPi.E01`
  - `4Dell Latitude CPi.E02`

Step 3: (Optional) Mount the Disk Image

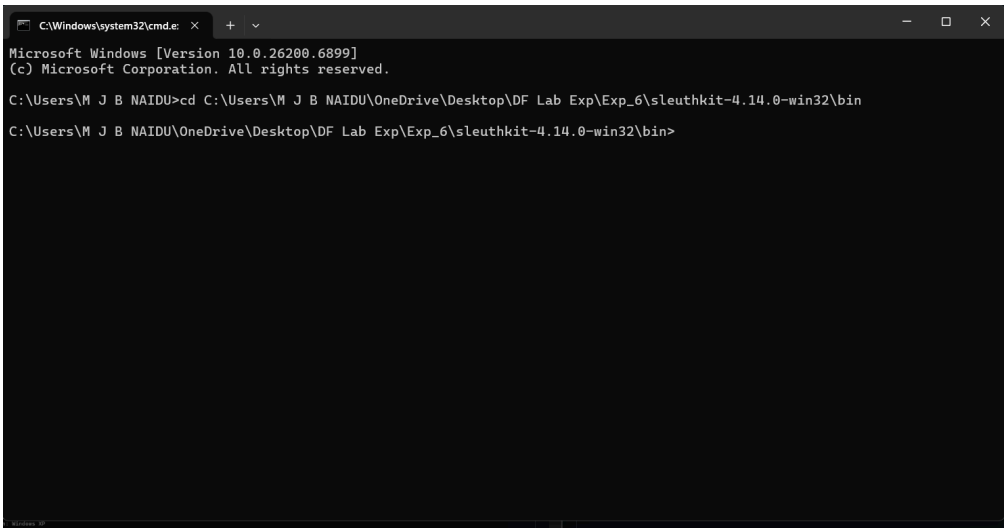
- Use **OSFMount** to mount the `.E01` or `.dd` image as a virtual drive.
- This helps in manually browsing the contents if needed.

Step 4: Analyze the File System

Navigate to Sleuth Kit:

Run:

```
cd C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp\Exp_6\sleuthkit-4.14.0-
```

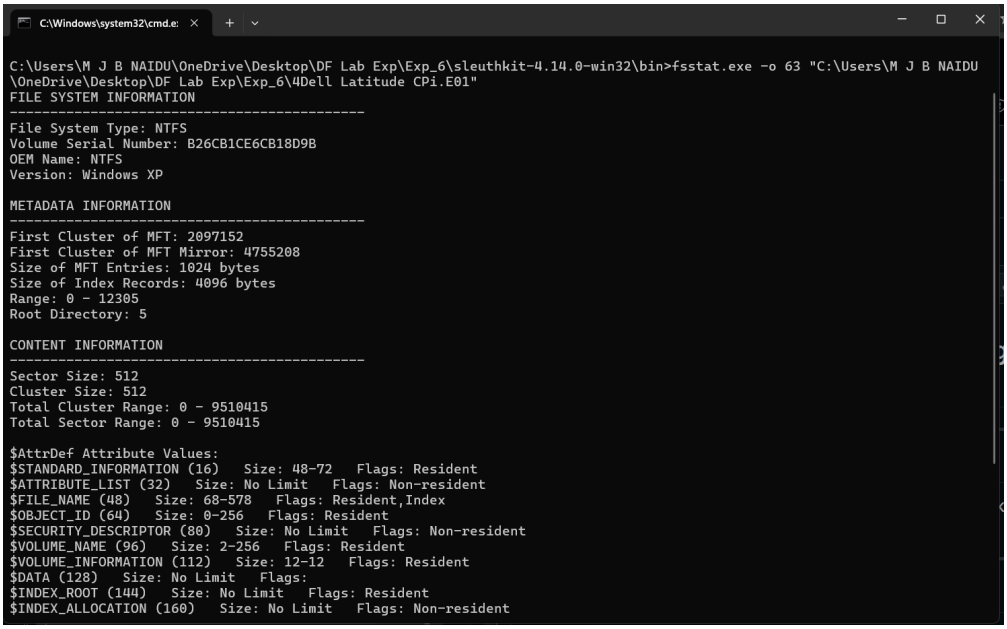


Identify File System Type:

Run:

```
fsstat.exe -o 63 "C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp\Exp_6\4D
```

This command gives details about the file system type, layout, and structure.

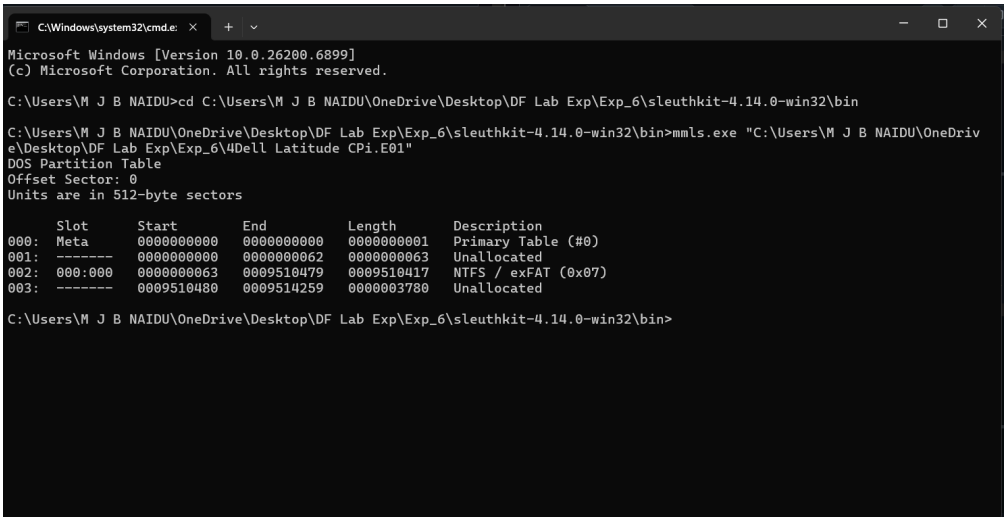


List Partitions:

Run:

```
mmls.exe "C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp\Exp_6\4Dell Lati
```

This lists all partitions present in the disk image.

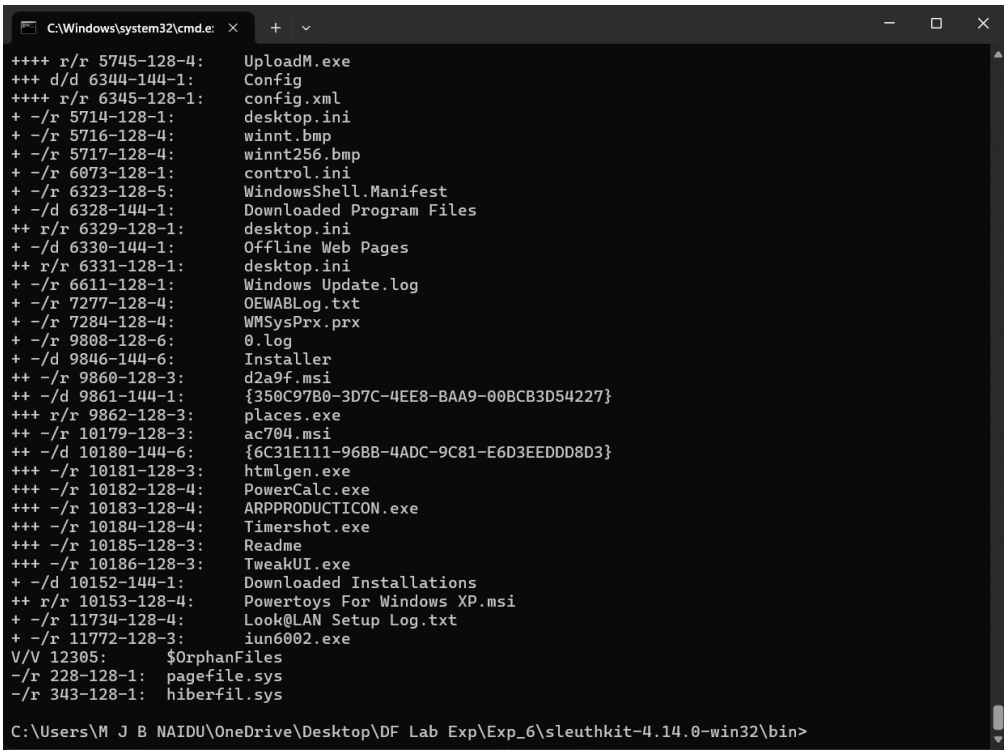


List Files and Directories

Run:

```
fls.exe -r -o 63 "C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp\Exp_6\4D
```

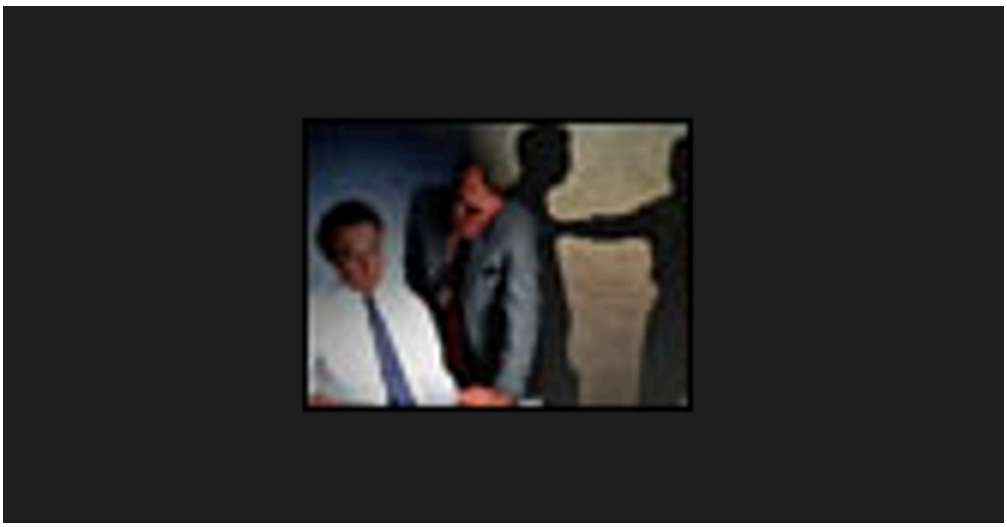
This recursively lists all files and directories, including deleted ones.



Recover Deleted Files

Run:

```
icat.exe -o 63 "C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp\Exp_6\4Del
```



Step 5: Analyze Metadata

Use the `istat` command to extract metadata of a specific file:

```
istat.exe -o 63  "C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp\Exp_6\4D
```

This provides information such as:

- File creation, modification, and access timestamps
- File size and allocation status

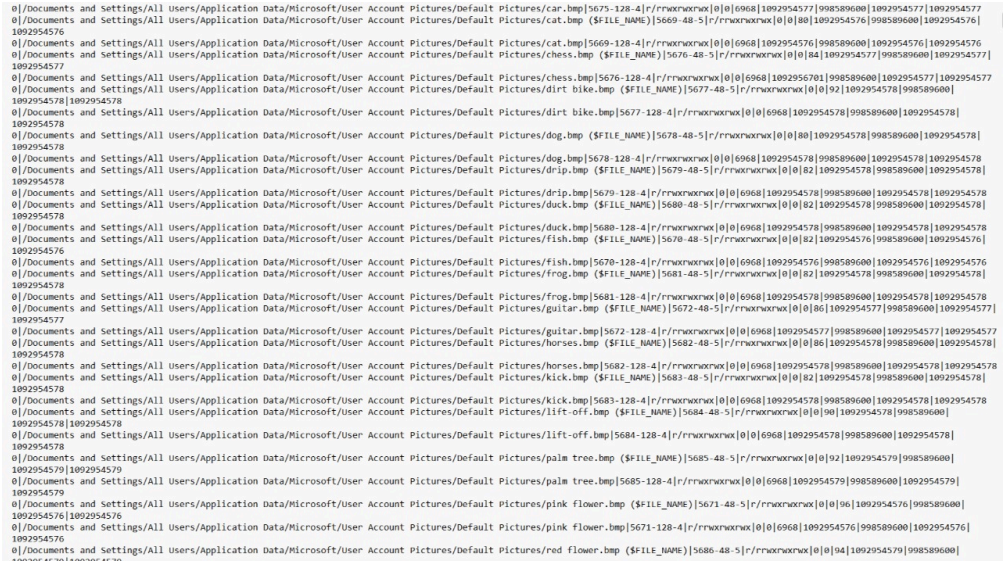
```
MFT Entry Header Values:
Entry: 11366        Sequence: 1
$LogFile Sequence Number: 31327279
Allocated File
Links: 2

$STANDARD_INFORMATION Attribute Values:
Flags: Archive
Owner ID: 0
Security ID: 295  (S-1-5-21-2000478354-688789844-1708537768-1003)
Created:        2004-08-21 00:35:21.487568000 (India Standard Time)
File Modified:  2004-08-21 00:35:21.487568000 (India Standard Time)
MFT Modified:   2004-08-21 00:35:21.487568000 (India Standard Time)
Accessed:       2004-08-21 00:35:21.487568000 (India Standard Time)

$FILE_NAME Attribute Values:
Flags: Archive
Name: TZ_CAR~1.JPG
Parent MFT Entry: 8375  Sequence: 3
Allocated Size: 0       Actual Size: 0
Created:        2004-08-21 00:35:21.487568000 (India Standard Time)
File Modified:  2004-08-21 00:35:21.487568000 (India Standard Time)
MFT Modified:   2004-08-21 00:35:21.487568000 (India Standard Time)
Accessed:       2004-08-21 00:35:21.487568000 (India Standard Time)

$FILE_NAME Attribute Values:
Flags: Archive
Name: tz_career_boss[1].jpg
Parent MFT Entry: 8375  Sequence: 3
Allocated Size: 0       Actual Size: 0
Created:        2004-08-21 00:35:21.487568000 (India Standard Time)
File Modified:  2004-08-21 00:35:21.487568000 (India Standard Time)
MFT Modified:   2004-08-21 00:35:21.487568000 (India Standard Time)
Accessed:       2004-08-21 00:35:21.487568000 (India Standard Time)

Attributes:
Type: $STANDARD_INFORMATION (16-0)   Name: N/A   Resident   size: 72
Type: $FILE_NAME (48-3)   Name: N/A   Resident   size: 90
Type: $FILE_NAME (48-2)   Name: N/A   Resident   size: 108
Type: $DATA (128-4)   Name: N/A   Non-Resident   size: 1644  init_size: 1644
3514379 3514380 3514381 3514382
```

Step 6: Timeline Analysis (Optional)

Generate a chronological timeline of file system activity.

1. Create a **body file**:

```
fls.exe -m / -r -o 63 "C:\Users\M J B NAIDU\OneDrive\Desktop\DF Lab Exp
```



2. Generate the **timeline**:

```
mactime -b body.txt > timeline.txt
```

This file lists the **MAC (Modified, Accessed, Changed)** timestamps of all files.

Step 7: Reporting

1. Collect all analysis outputs:
    - `filesystem_info.txt`
    - `partitions.txt`
    - `file_list.txt`
    - `metadata_info.txt`
    - `timeline.txt` (if created)
2. Review and summarize findings such as:
    - File system type and structure
    - Deleted or recovered files
    - File metadata insights
    - Activity timeline (if available)

Step 8: Finalize and Store Evidence

- Archive all reports and analysis files securely.
- Maintain the **chain of custody** to ensure evidence integrity.
- Store archives in a **secure and access-controlled** environment.

Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Successfully analyzed the given disk image using **The Sleuth Kit (TSK)**.
Extracted file system information, recovered deleted files, analyzed metadata, and generated a forensic timeline report.

Bhuvaneshwar-Naidu Update Exp_7_AFLogical_OSE.md 4a12dba · 1 minute ago

208 lines (158 loc) · 5.35 KB

Ex.No.7: Use AFLogical OSE to Extract Data from an Android Device

Aim

To extract logical data such as contacts, messages, call logs, and other user information from an Android device using **AFLogical OSE (Open Source Edition)** as part of digital forensic analysis.

STEP 1 — Initial Setup & File Extraction

Required Files (Pre-requisites)

- **Android Platform Tools (ADB):** For device communication
- **AFLogical OSE ZIP (Source/APK):** Core forensic extraction tool
- **Google USB Driver (Windows):** For PC-device connectivity

Instructions

1. Create the main lab directory:

```
C:\DF
```

2. Extract all downloaded ZIP archives into this folder:

```
C:\DF\platform-tools\
C:\DF\aflogical-ose\
C:\DF\usb-driver\
```

3. **Note:**
   If `AFLogical-OSE.apk` is missing, use **Santoku Linux** or another forensic OS to build or extract it from the source.

STEP 2 — Configure System Environment (PATH)

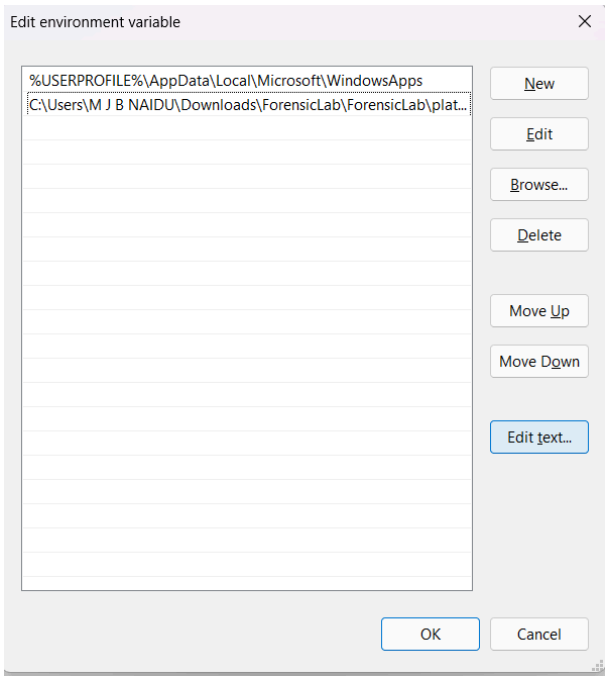
Purpose

To make `adb` commands accessible from any terminal or command prompt without specifying the full path.

Steps

1. Open:
   Control Panel → System → Advanced system settings → Environment Variables
2. Under *User Variables*, select **Path → Edit → New**.
3. Add:

```
C:\DF\platform-tools
```

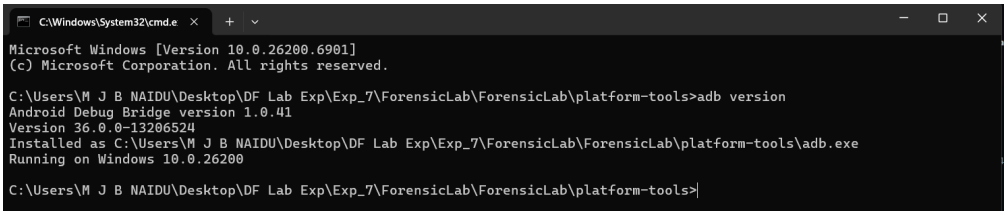


4. Click **OK** to apply.

Verification

Run:

```
adb version
```



Expected Output:
Displays the installed **Android Debug Bridge version**.

STEP 3 — Install Google USB Driver (Windows Specific)

Purpose

To ensure the Windows system can identify and communicate with the connected Android device.

Steps

1. Connect the Android phone via USB.
2. Open **Device Manager** → Locate your phone.
3. Right-click → **Update Driver** → **Browse my computer for drivers**.
4. Specify path:

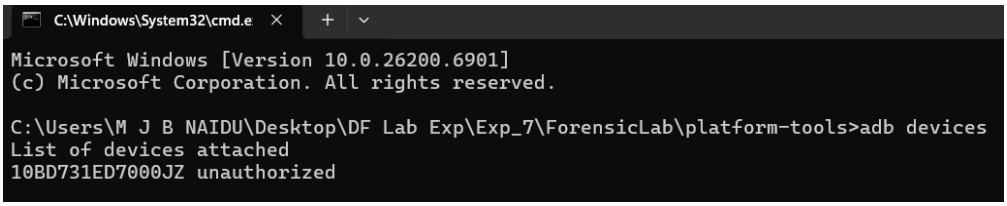
   ```
   C:\DF\usb-driver
   ```

5. Click **Next** to complete installation.

Verification

Run:

```
adb devices
```



Device should be listed as "**device**", not *offline* or *unauthorized*.

STEP 4 — Prepare the Android Device (Developer Options)

Steps

1. Go to **Settings → About Phone → Tap "Build Number" 7 times**.
2. Return to **Settings → Developer Options**.
3. Enable:
   - USB Debugging
   - Install via USB (if available)

STEP 5 — Establish and Verify ADB Connection

Purpose

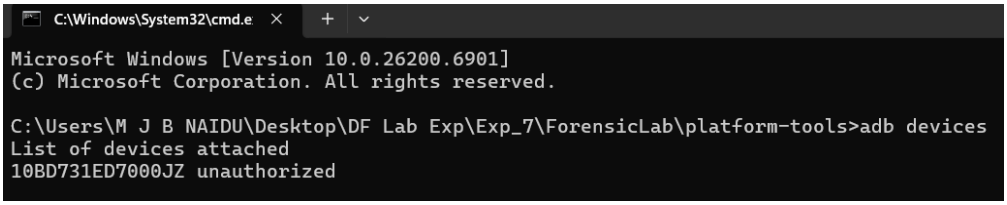
To confirm a stable and authorized link between your computer and the Android device.

Steps

1. Connect your phone via USB.
2. Run:

```
adb devices
```

3. Tap **Allow** on the phone if prompted for debugging authorization.



Troubleshooting

If the device shows as *unauthorized*, replug it and reauthorize USB debugging.

STEP 6 — Deploy AFLogical OSE to the Device

Purpose

To install the AFLogical forensic application on the Android device.

Steps

1. Confirm APK location:

```
C:\DF\aflogical-ose\AFLogical-OSE.apk
```

2. Install using ADB:

```
adb install --bypass-low-target-sdk-block "C:\Users\Manya\Downloads\DF\
```

STEP 7 — Execute Logical Data Extraction

Purpose

To perform the actual data extraction from the Android device using AFLogical.

Steps

1. Open **AFLogical** on the Android device.
2. Grant all necessary permissions (Contacts, SMS, Call Logs, Storage).
3. Select the data types to extract:
   - Contacts
   - SMS
   - Call Logs
   - MMS
   - Calendar
4. Tap **Start Extraction**.
5. Wait for extraction to finish.

Default Save Location

```
/sdcard/aflogical/
```

or

```
/storage/emulated/0/aflogical/
```

STEP 8 — Collect Extracted Data (Pull to PC)

Purpose

To transfer the extracted `.csv` data from the Android device to your computer.

Command

```
adb pull /sdcard/aflogical C:\Users\Manya\Downloads
```

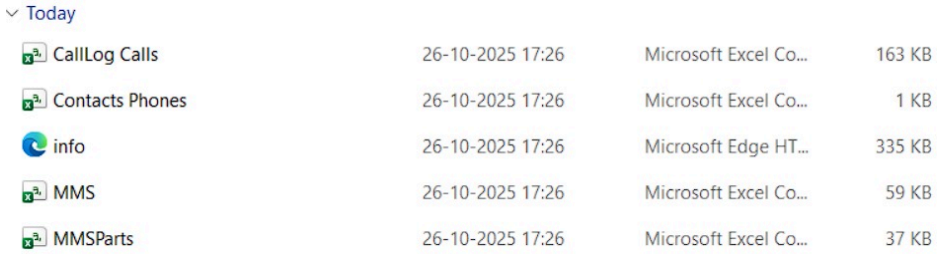
Verification

Extracted files will be saved in:

```
C:\Users\Manya\Downloads
```

The folder should contain files like:

- `contacts.csv`
- `sms.csv`
- `calllogs.csv`
- `calendar.csv`



Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Successfully extracted logical data (Contacts, SMS, Call Logs, etc.) from an Android device using **AFLogical OSE**, transferred it to the computer using ADB, and analyzed the extracted `.csv` files for forensic investigation.

Ex.No.8: Use StegExpose to Detect Hidden Data in Images

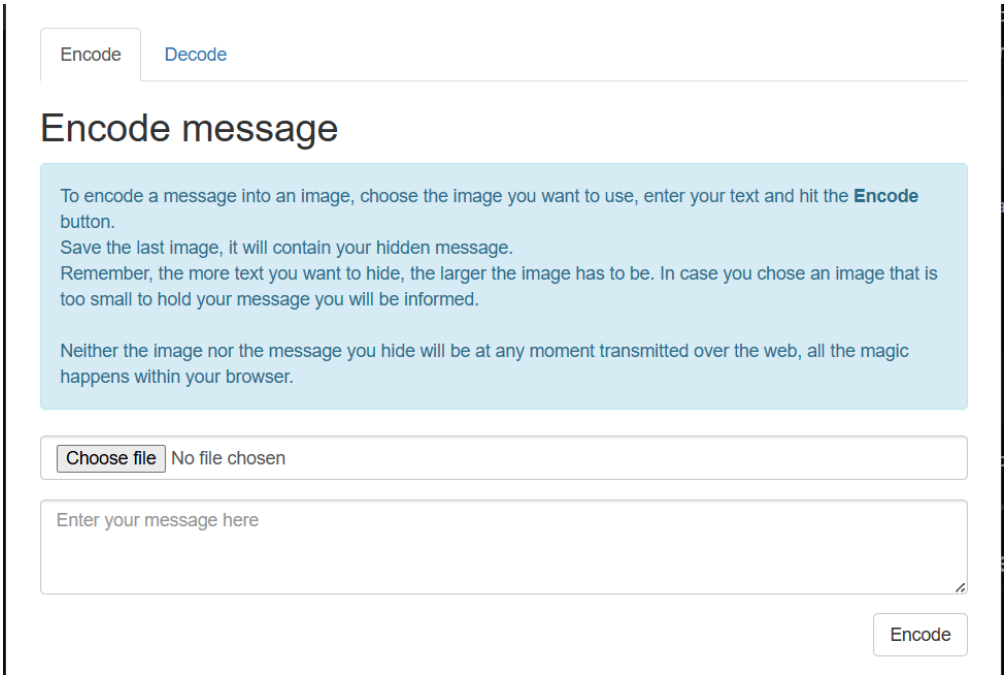
Aim

To detect the presence of hidden data within digital images using **StegExpose**, a steganalysis tool that evaluates image statistics to identify steganographic content.

Step-by-Step Procedure

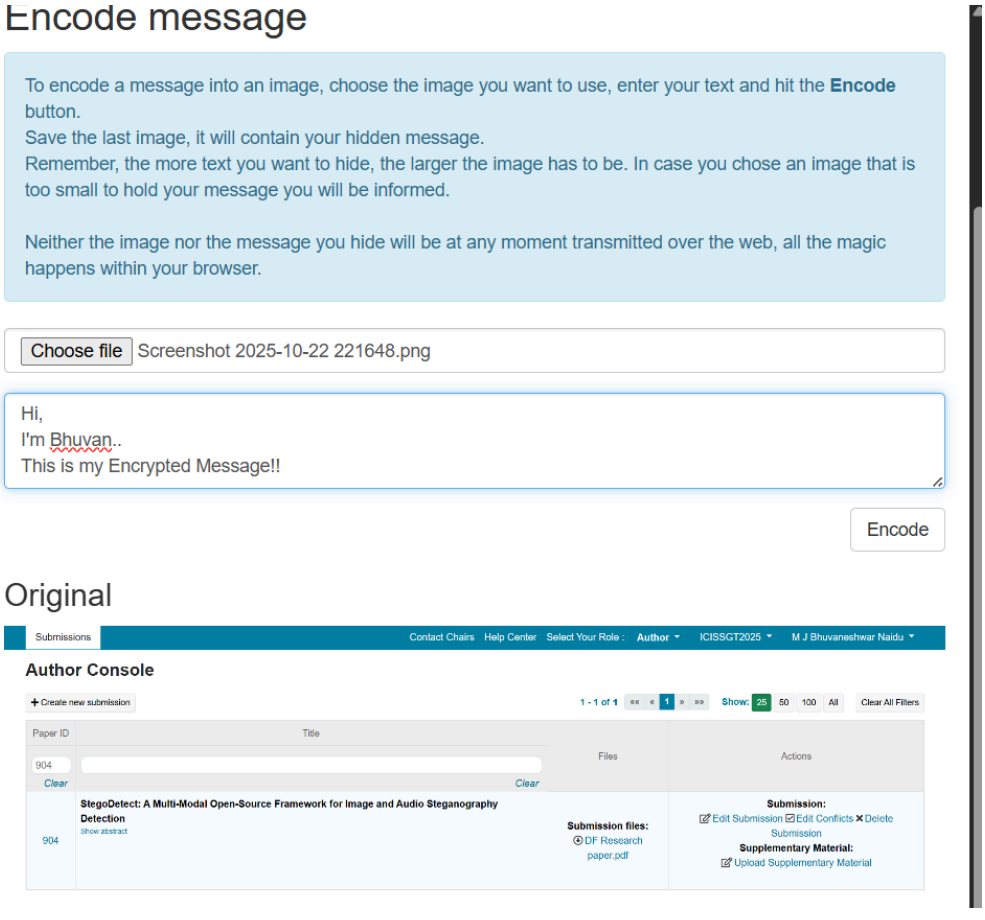
Step1: Download and Set Up for Steganography

1. Visit the **StegExpose GitHub page** and download the `.jar` file.
2. Ensure **Java** is installed on your system. If not, download it from Oracle's official website.
3. Place the `StegExpose.jar` file in your working directory (e.g., `C:\DF\StegExpose\`).
4. Or you can run it online, using any Steganography tools available online.



Step2: Select Images for Encrytion

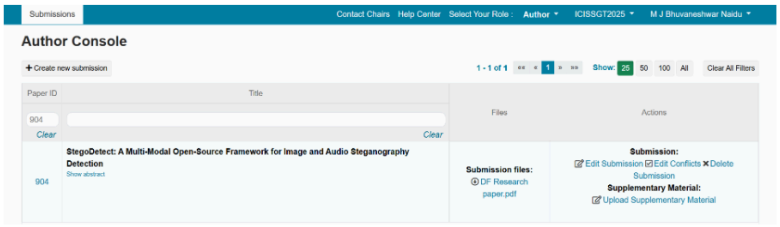
- Select the file you want to hide data.
- Give the message you want to encrypt in the file.
- Supported formats include `.png` , `.jpg` , `.bmp` and `.pdf` .



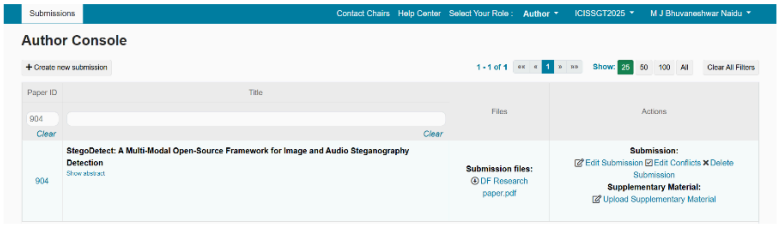
Step3: Compare and Save the Stegno Image

1. Compare the difference between the normal and the stegnoimage.
2. Right click the image and click on save image.
3. Name the file as **Stegno image** and click on save.

Normalized

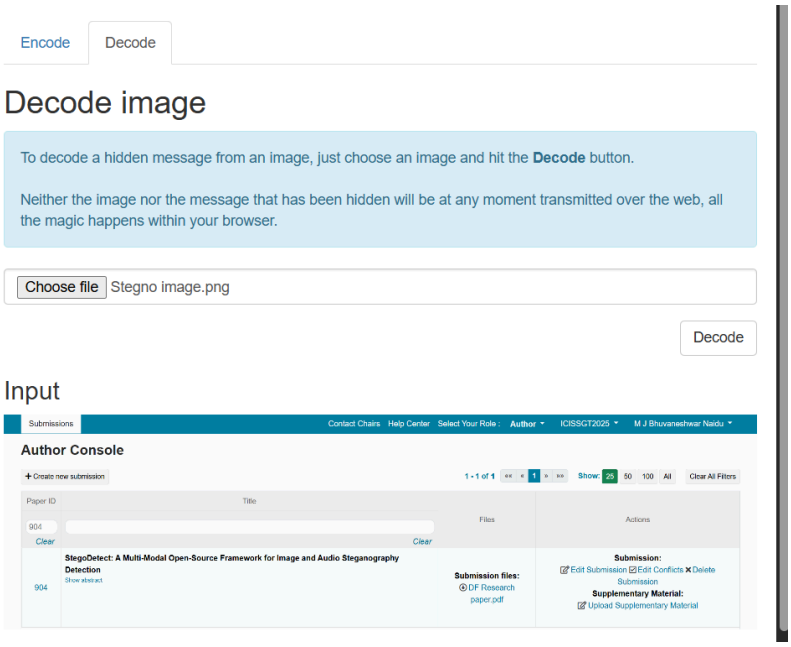


Message hidden in image (right click ➡ save as)

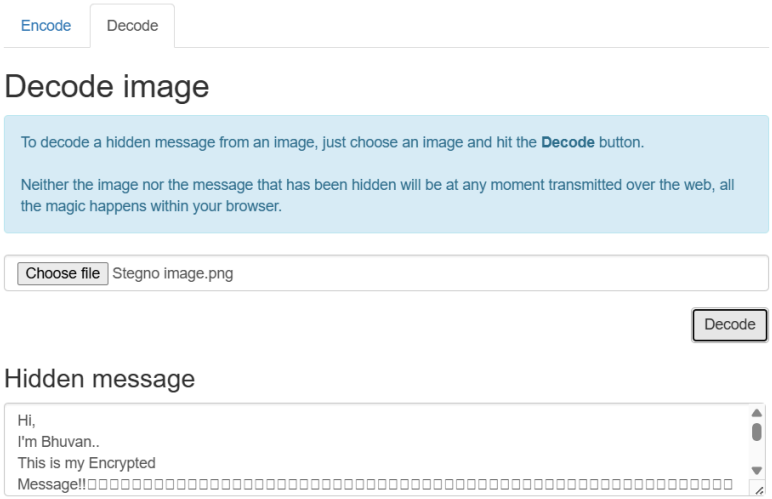


Step4: Run Decode on a Stegno Image

1. Select the image you want decode.
2. Click on decode.



Step5: Analyze the Output



Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Successfully used **StegExpose** to perform steganalysis on images, identified potential hidden data through suspect scores, and interpreted the likelihood of steganography based on the tool's statistical output.

Bhuvaneshwar-Naidu Update Exp_9_Process_Explorer.md 9ddb462 · now

149 lines (109 loc) · 4.97 KB

Ex.No.9: Use Process Explorer to Identify Suspicious Processes

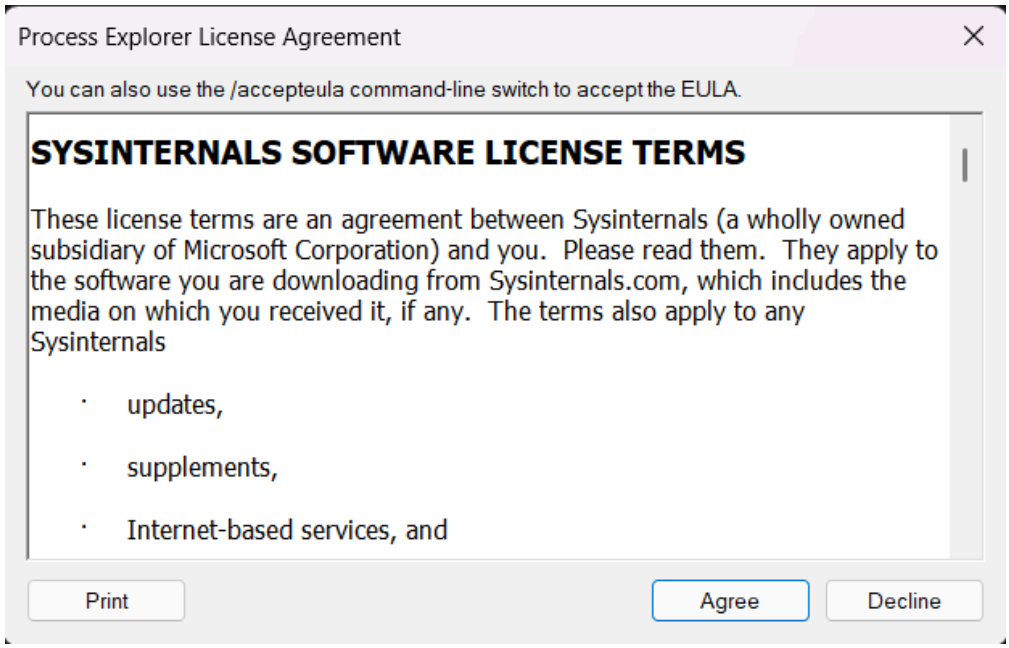
Aim

To identify and analyze suspicious or potentially malicious processes running on a Windows system using **Process Explorer**, a tool from Microsoft Sysinternals Suite.

STEP 1 — Download and Set Up Process Explorer

Instructions

1. **Download Process Explorer**
   - Visit the official Microsoft Sysinternals website:
     Download Process Explorer
2. **Extract the Program**
   - Extract the downloaded ZIP file to a preferred location on your computer.
3. **Run Process Explorer**
   - Open the folder and launch the appropriate version:
     - `procexp64.exe` for 64-bit systems
     - `procexp.exe` for 32-bit systems
   - Right-click and select **Run as Administrator** to ensure full privileges.



STEP 2 — Familiarize Yourself with the Interface

Key Components

- **Process Tree:** Displays hierarchical structure of running processes.
- **Color Codes:**
  - Pink → Suspended processes
  - Light Blue → Processes under the current user
  - Dark Blue → System or service processes
  - Green → Newly created processes
  - Red → Recently exited processes

Columns Overview

- **PID:** Process ID number
- **CPU Usage:** Real-time processor consumption
- **Memory Usage:** RAM utilization
- **Description & Company Name:** Metadata for legitimacy verification

---

STEP 3 — Identify Suspicious Processes

1 Look for Unfamiliar Processes

- Review all running processes and identify unknown or oddly named ones.

- Malware often disguises itself using similar names to legitimate processes.

2 Verify Digital Signatures

- Right-click a process → **Properties** → **Image Tab** → **Verify**.
- Check for a valid **Digital Signature**.
  - Valid Signature → Legitimate software
  - No/Invalid Signature → Potentially malicious

3 Check Process Path

- In the **Properties** → **Image Tab**, review the file path.
  - Legitimate processes reside in:

```
C:\Windows\System32
```

  - Suspicious if running from:
    - Temporary folders
    - User download folders
    - Unknown directories

4 Monitor Resource Usage

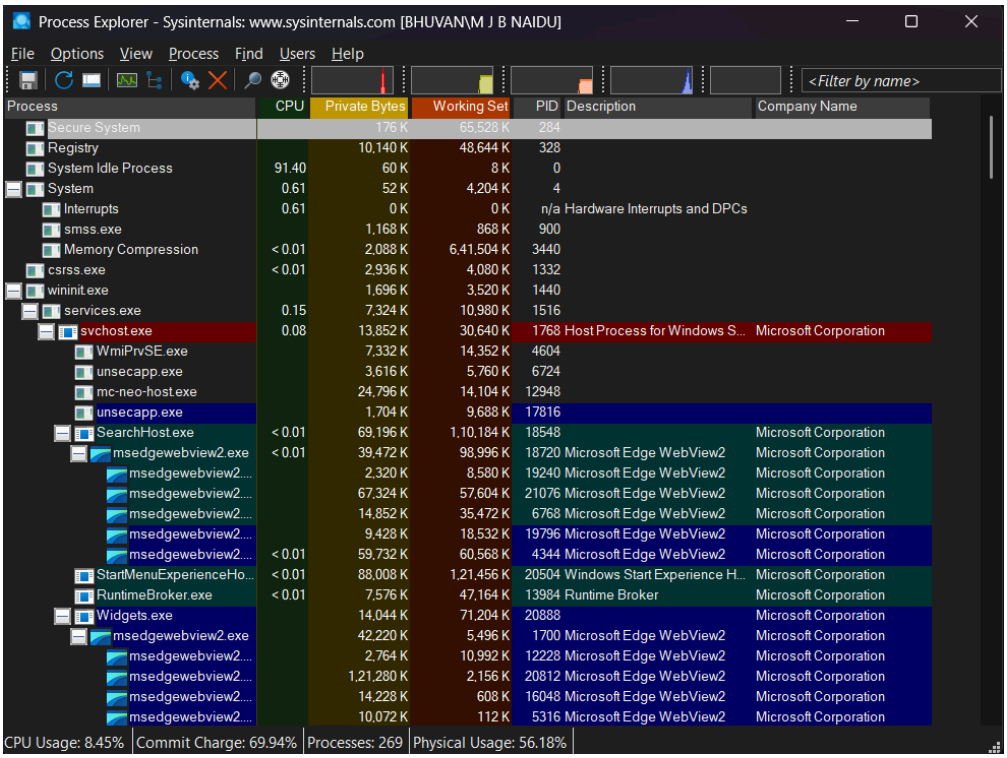
- Observe CPU, Memory, and Disk columns.
- Abnormally high or fluctuating usage could signal malware.

5 Review Description & Company Name

- Missing or misleading information may indicate fake or rogue software.

6 Check Network Activity

- Right-click process → **Properties** → **TCP/IP Tab**.
- Monitor for unexpected network connections to unknown IPs.



STEP 4 — Perform Online Verification

Actions

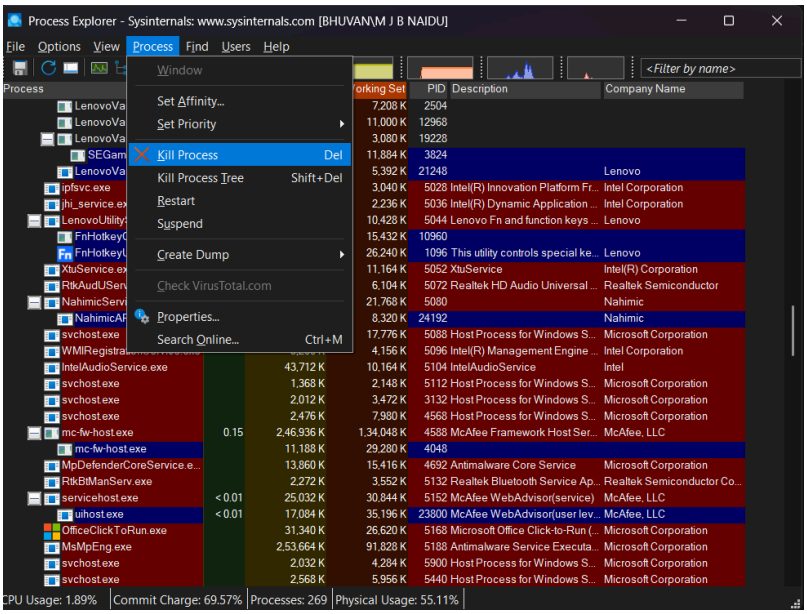
- Perform a quick Google search using the suspicious process name (e.g., `randomname.exe`).
- Cross-check the process in malware databases like:
  - VirusTotal
  - ProcessLibrary

STEP 5 — Take Action on Suspicious Processes

Options

- **Kill Process:**
  Terminate the process immediately:
  `Right-click → Kill Process`
- **Suspend Process:**
  Temporarily pause activity for further analysis:
  `Right-click → Suspend`
- **Delete Source File:**
  Locate the file via **Path** and delete it if confirmed malicious.

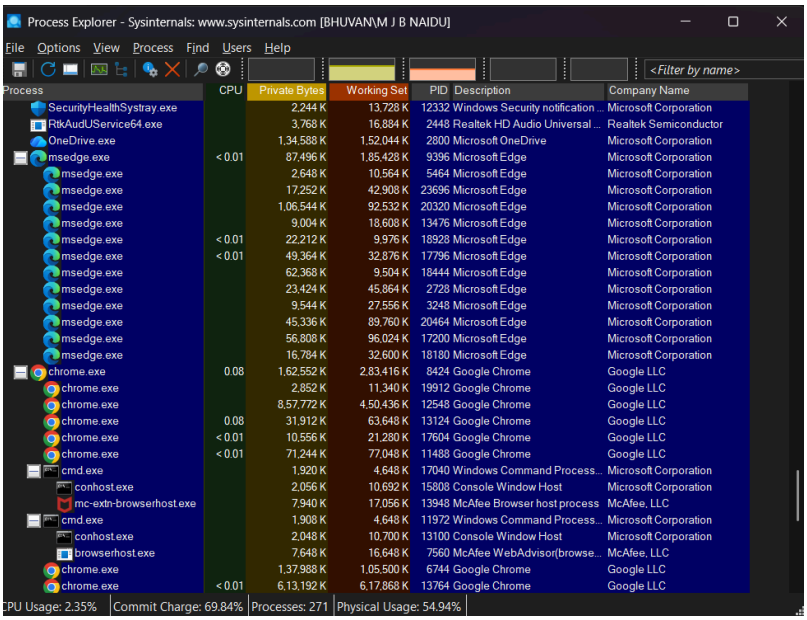
> *Note:* Some malware prevents termination. In such cases, reboot into **Safe Mode** or use antivirus tools.



STEP 6 — System Cleanup and Scan

Recommendations

- Perform a full antivirus scan using tools like:
  - Windows Defender
  - Malwarebytes Anti-Malware
- Remove quarantined threats and restart the system.



Example — Identifying a Malicious Process

Action Taken

- Suspended → Killed the process
- Removed file from directory
- Performed full malware scan

Result: Confirmed as malicious software and successfully removed.

Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Successfully utilized **Process Explorer** to monitor and analyze system processes, identify suspicious activities, and mitigate potential malware threats.

Ex No-10: Use Ghidra to Disassemble and Analyze Malware Code

Aim

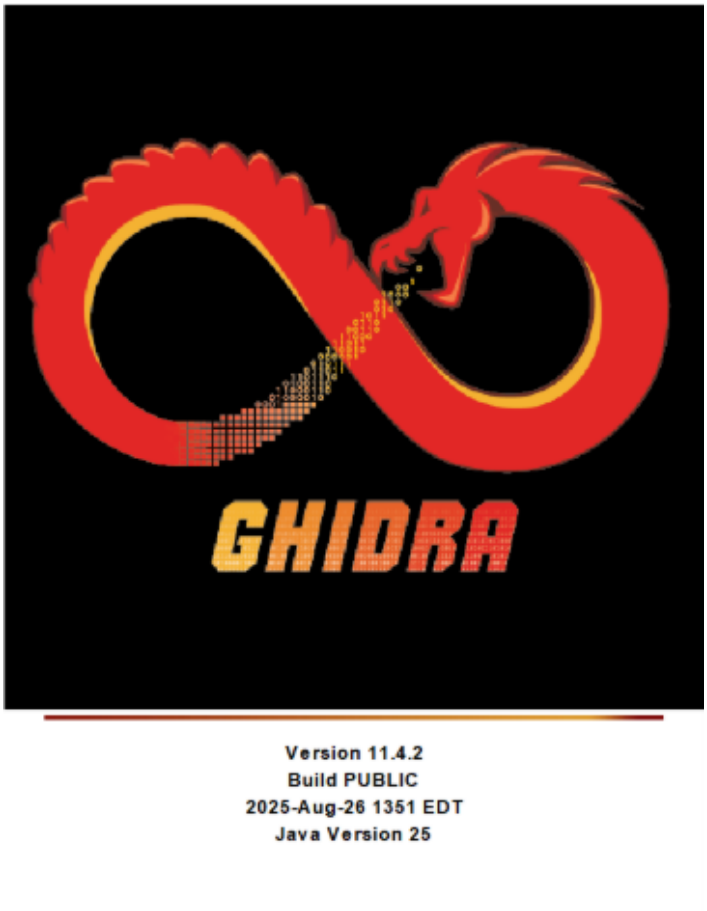
To disassemble and analyze a binary using **Ghidra** to identify potential malicious behavior and understand code functionality.

Description

Ghidra is a free reverse-engineering framework that helps analysts disassemble and decompile binaries to inspect assembly and pseudocode. This experiment covers loading a binary, running auto-analysis, locating key functions and strings, and deriving behavioral indicators.

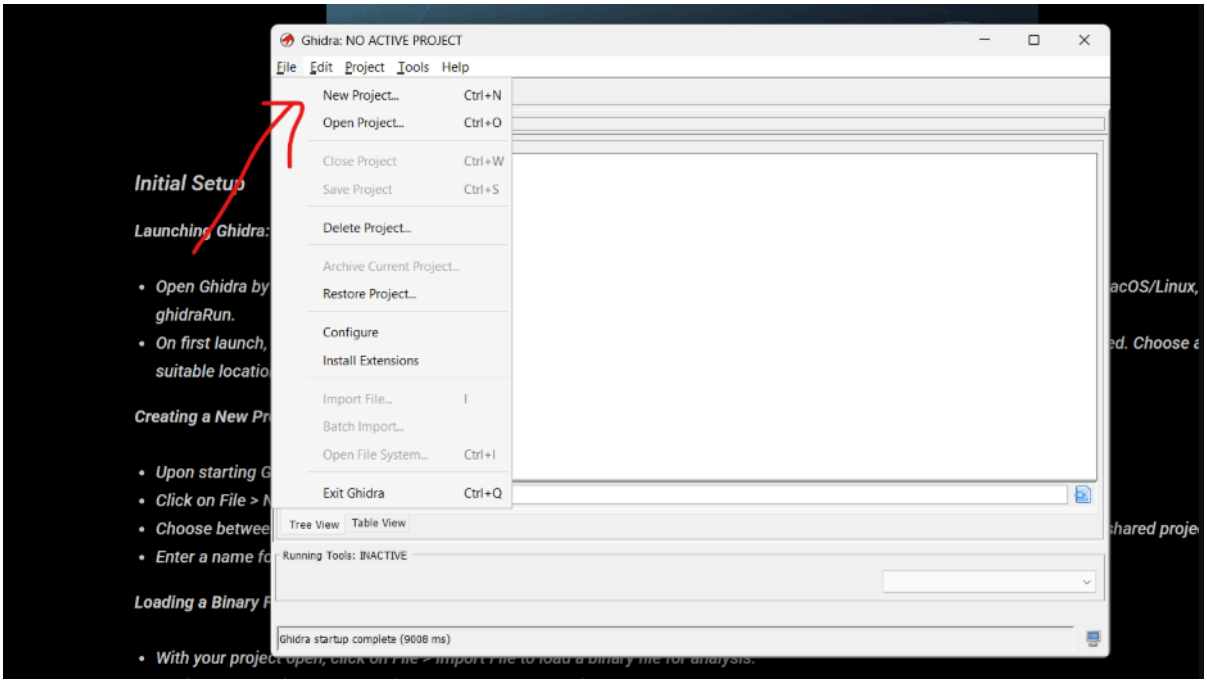
Steps

1. **Prepare Environment**
   - Use an isolated VM (VirtualBox/VMware) with a snapshot.
   - Install Ghidra and required dependencies.



---

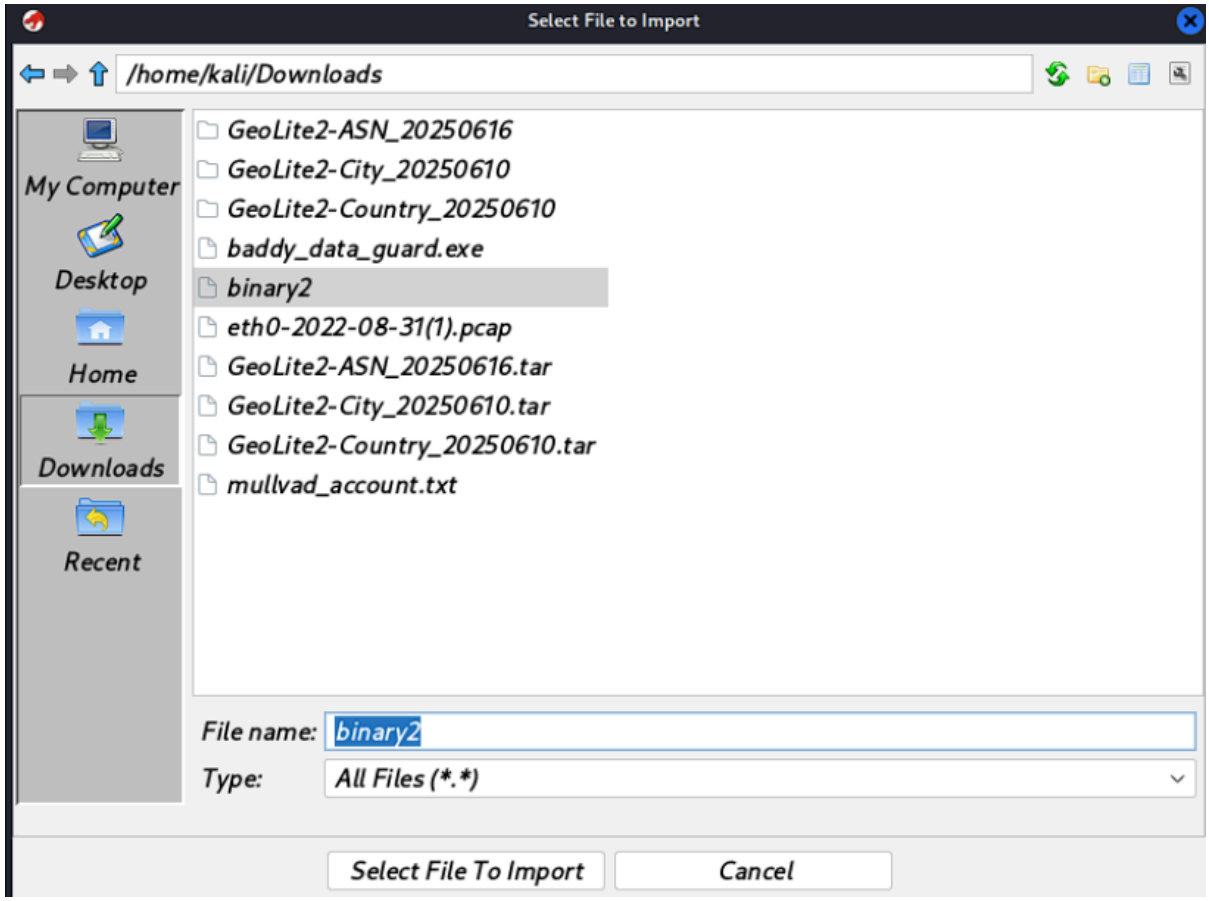
2. **Load Binary**
    - Open Ghidra and create a new project.
    - Import the target binary into the project.



---

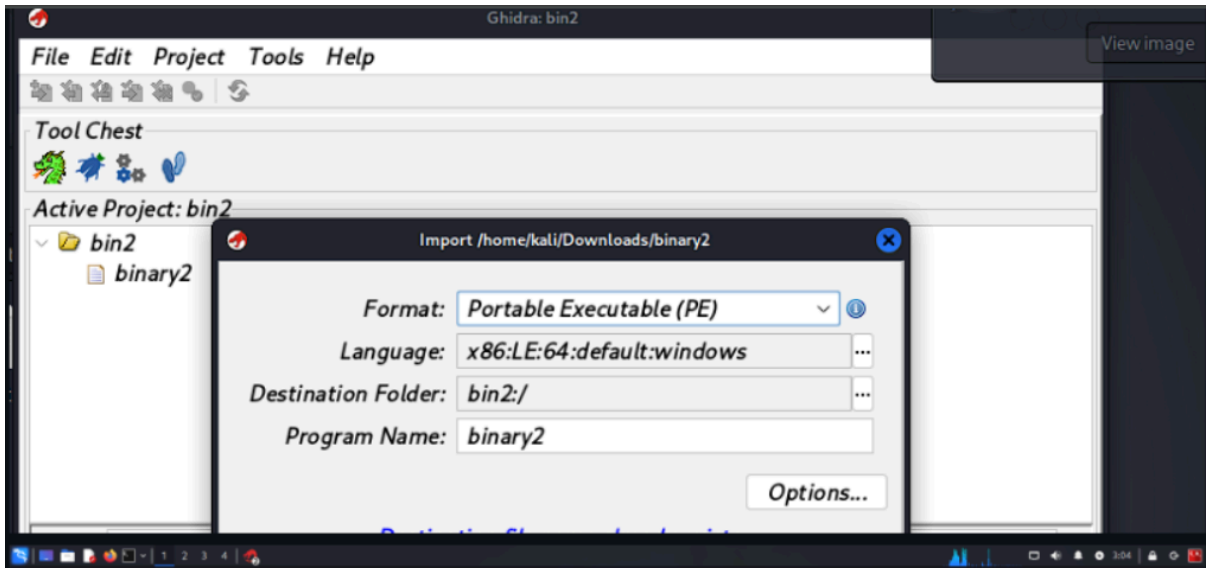
3. **Run Auto-Analysis**
    - Allow Ghidra to perform auto-analysis with default options.

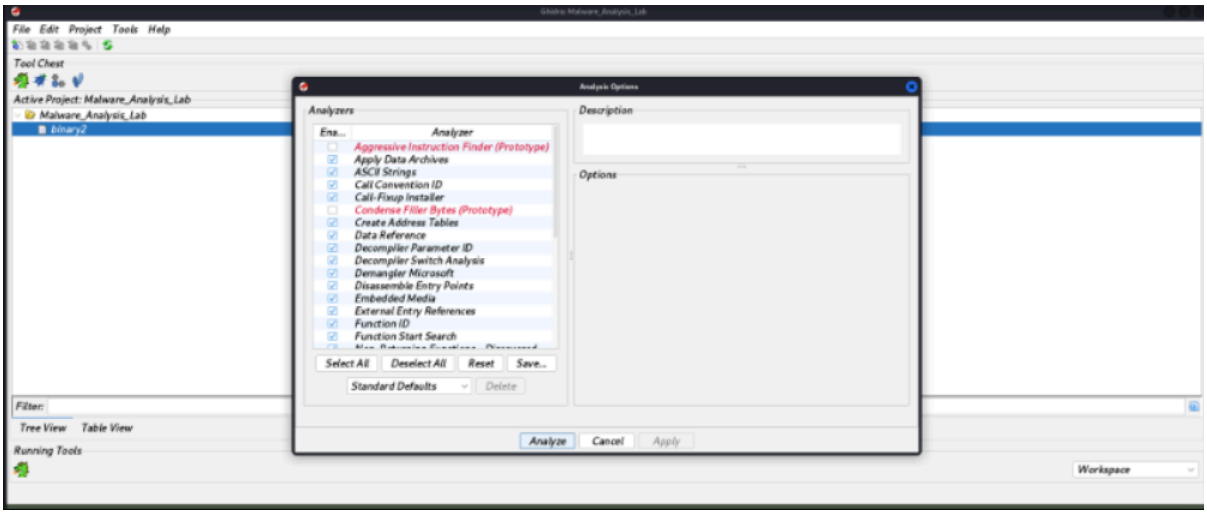
- Review the analysis log for any warnings.



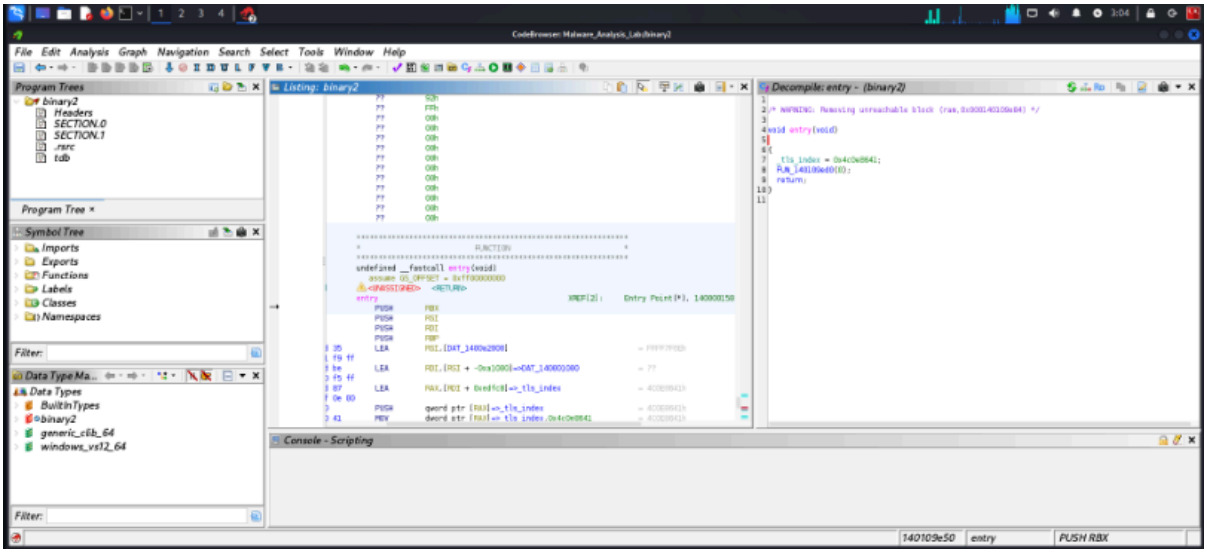
---

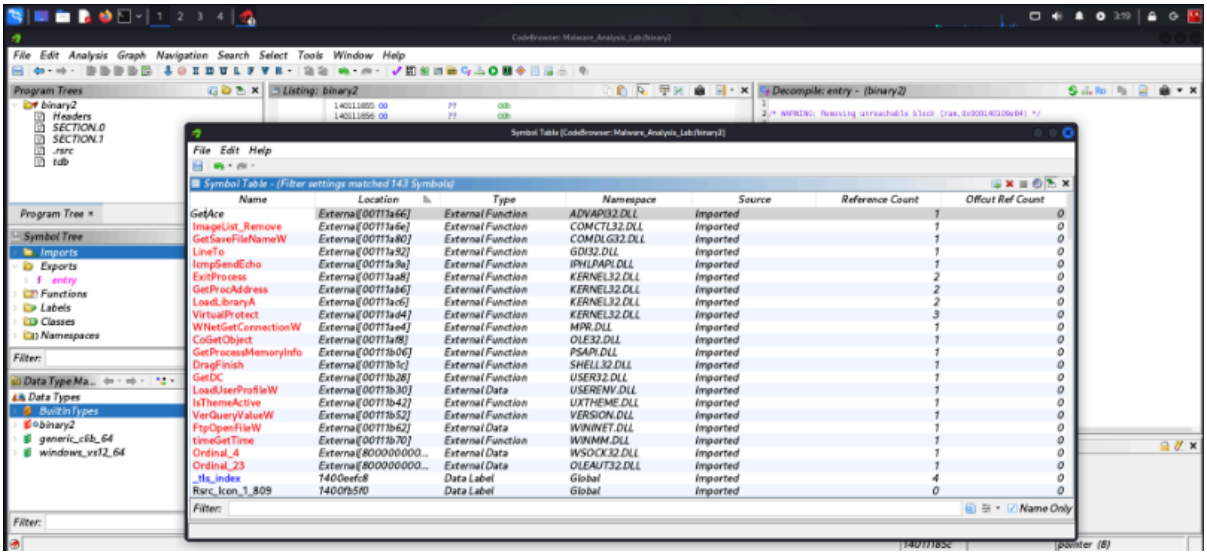
4. **String and Import Inspection**
   - Search for suspicious strings (URLs, file paths, commands).
   - Check imported functions for networking, file, or process APIs.





5. **Function Analysis**
    - Identify interesting functions via cross-references (xrefs).
    - Use the Decompiler to read C-like pseudocode and annotate logic.
    - Rename functions and variables for clarity.





Rubrics

Criteria	Mark Allotted	Mark Awarded
1. GitHub Activity & Submission Regularity	3	
2. Application of Forensic Tools & Practical Execution	3	
3. Documentation & Reporting	2	
4. Engagement, Problem-Solving & Team Collaboration	2	
Total	10	

Result

Loaded the binary into Ghidra, performed static analysis to identify key functions and strings, and produced a summarized report of observed behaviors and indicators of compromise (IoCs).