NAME: Adithya Baragi S

INTERMEDIATE DIGITAL FORENSICS CONCEPTS

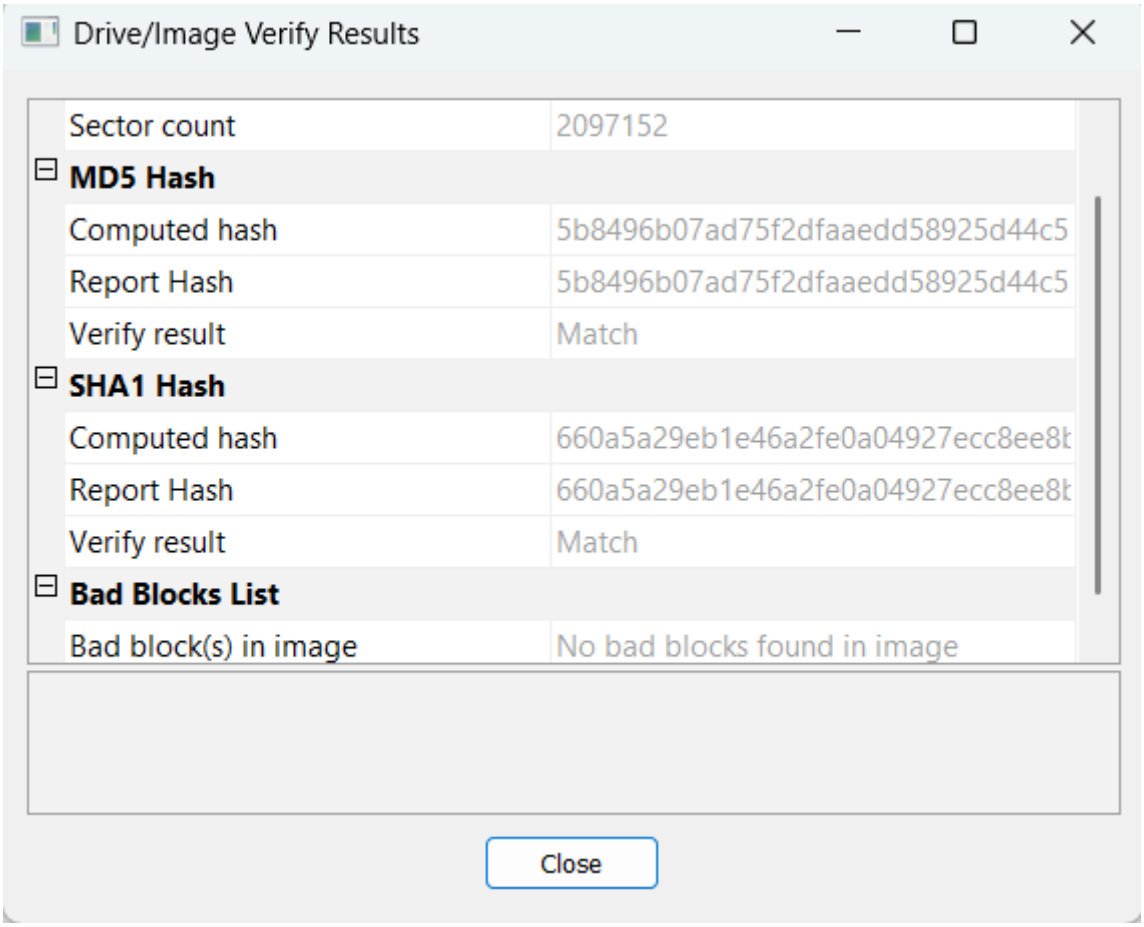
TASK 1 — Recover Deleted Files (Autopsy + Foremost/Scalpel)

First, we should download the Raw image (dd)(EO1) file from the website: https:// NIST CFReDS name:

Recovering Deleted Files with Autopsy

Step-by-Step

- Open Autopsy → New Case
- Add Data Source → Disk Image / E01
- Go to:
  - File Analysis → Deleted Files
- Autopsy uses NTFS MFT entries to identify deleted files.
- Recover files:
  - Right-click → Extract File(s)



Manual Carving with Foremost

Command used to install manually Carving the foremost for the image : **sudo apt install foremost -y**
After installation we should Run Carving to find the exact output : **foremost -i mantolab_ntfs_2024.img -o foremost_output**

Carves:

- **jpg**
- **png**
- **doc/docx**
- **pdf**
- **zip**
- **exe**

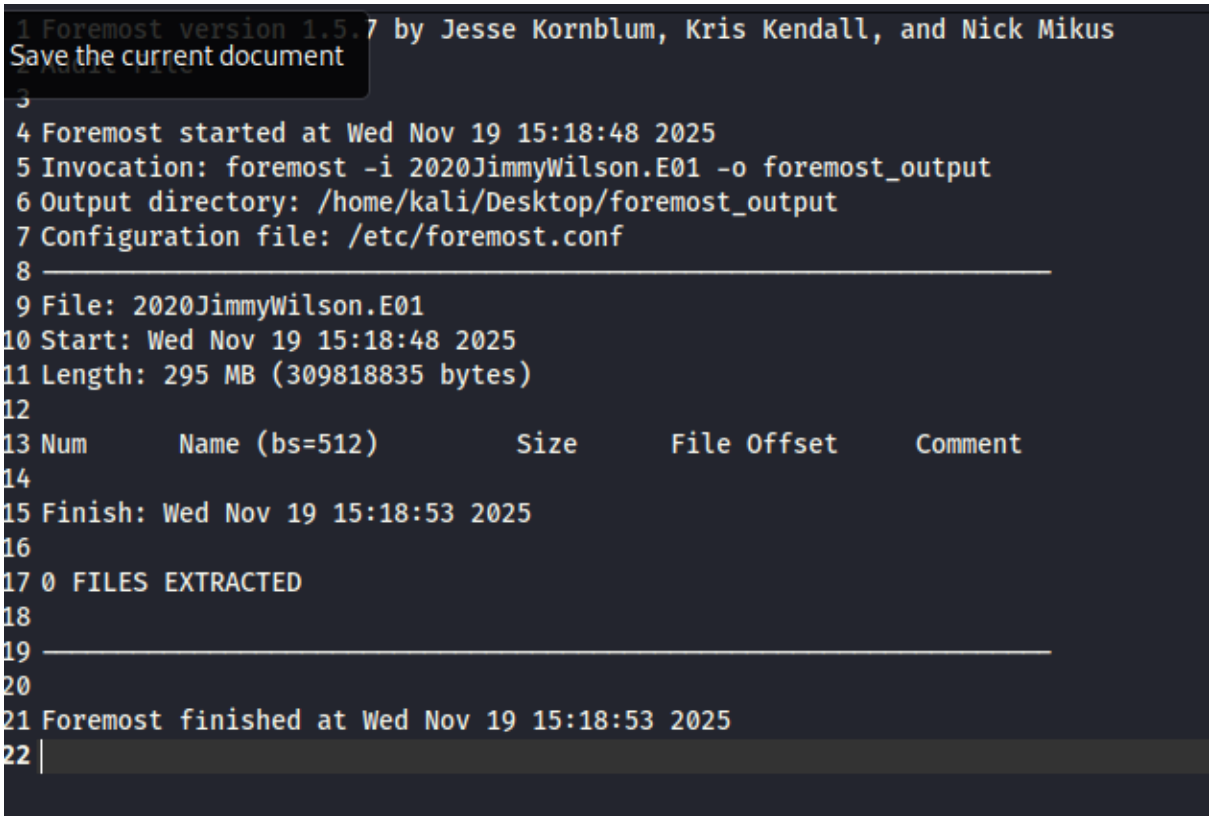
```
┌──(kali㉿vbox)-[~/Desktop]
└─$ foremost -i 2020JimmyWilson.E01 -o foremost_output

Processing: 2020JimmyWilson.E01
|foundat=xulcache/resource/app/chrome/browser/content/browser/places/treeView.jsUT
foundat=jssubloader/185/resource/gre/modules/commonjs/sdk/util/array.jsUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/autocomplete.xmlUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/textbox.xmlUT
foundat=xblcache/resource/app/chrome/browser/content/browser/urlbarBindings.xmlUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/scrollbox.xmlUT
foundat=xulcache/resource/app/chrome/browser/content/browser/places/browserPlacesViews.js
foundat=nsXULPrototypeCache.startupCacheUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/menu.xmlUT
foundat=xulcache/resource/app/chrome/browser/content/browser/downloads/downloads.jsUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/browser.xmlUT
foundat=xblcache/resource/app/chrome/browser/content/browser/socialchat.xmlUT
foundat=xulcache/resource/app/chrome/browser/content/browser/nsContextMenu.jsUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/videocontrols.xmlUT
foundat=xulcache/resource/gre/chrome/toolkit/content/global/inlineSpellCheckUI.jsUT
foundat=jsloader/resource/app/chrome/pdfjs/content/PdfJs.jsmUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/richlistbox.xmlUT
foundat=jssubloader/185/resource/gre/modules/commonjs/sdk/platform/xpcom.jsUT
foundat=xulcache/resource/app/chrome/browser/content/browser/places/controller.jsUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/general.xmlUT
foundat=jssubloader/185/resource/gre/modules/commonjs/sdk/system/events.jsUT
foundat=jsloader/resource/app/chrome/pdfjs/components/PdfStreamConverter.jsUT
foundat=jssubloader/185/resource/gre/modules/commonjs/sdk/util/object.jsUT
foundat=xblcache/resource/gre/chrome/toolkit/content/global/bindings/tree.xmlUT
**foundat=en_CA.dicUT
foundat=README_en_CA.txtUT
foundat=README_hyph_en_GB.txtUT
foundat=README.txtUT
foundat=th_en_US_v2.datUT
foundat=en_GB.dicUT
foundat=license.txtUT
foundat=en_GB.affUT
foundat=dictionaries.xcuUT
foundat=en_AU.dicUT
*|
```

Manual Carving with Scalpel

First, we should install Scalpel tool using the command : **sudo apt install scalpel -y**
Edit config to enable file types: **sudo nano /etc/scalpel/scalpel.conf**
Here we should uncomment the file types eg: pdf, jpg, png, doc
Then we should run the command to see the Artifacts inside the image **: scalpel mantolab_ntfs_2024.img -o scalpel_output**

File Name	Source	Path	MACB Times (UTC)	Alloc Status	Recovered By
report.docx	$MFT	/Users/Admin/Documents	M: 2023-12-04 09:33 A: 2023-12-04 09:01 C: 2023-12-04 09:01 B: 2023-12-04 09:33	Deleted	Autopsy
creds.txt	Unallocated	Sector 312001–312128	M: Unknown A: Unknown C: Unknown B: Unknown	Unallocated	Foremost
login.jpeg	$MFT	/Users/Public/Pictures	M: 2024-01-10 12:03 A: 2024-01-10 12:03 C: 2024-01-10 12:03	Deleted	Scalpel
malware.exe	$MFT	/Windows/Temp	M: 2024-02-02 07:00 A: 2024-02-02 07:00 C: 2024-02-02 06:55	Deleted	Autopsy

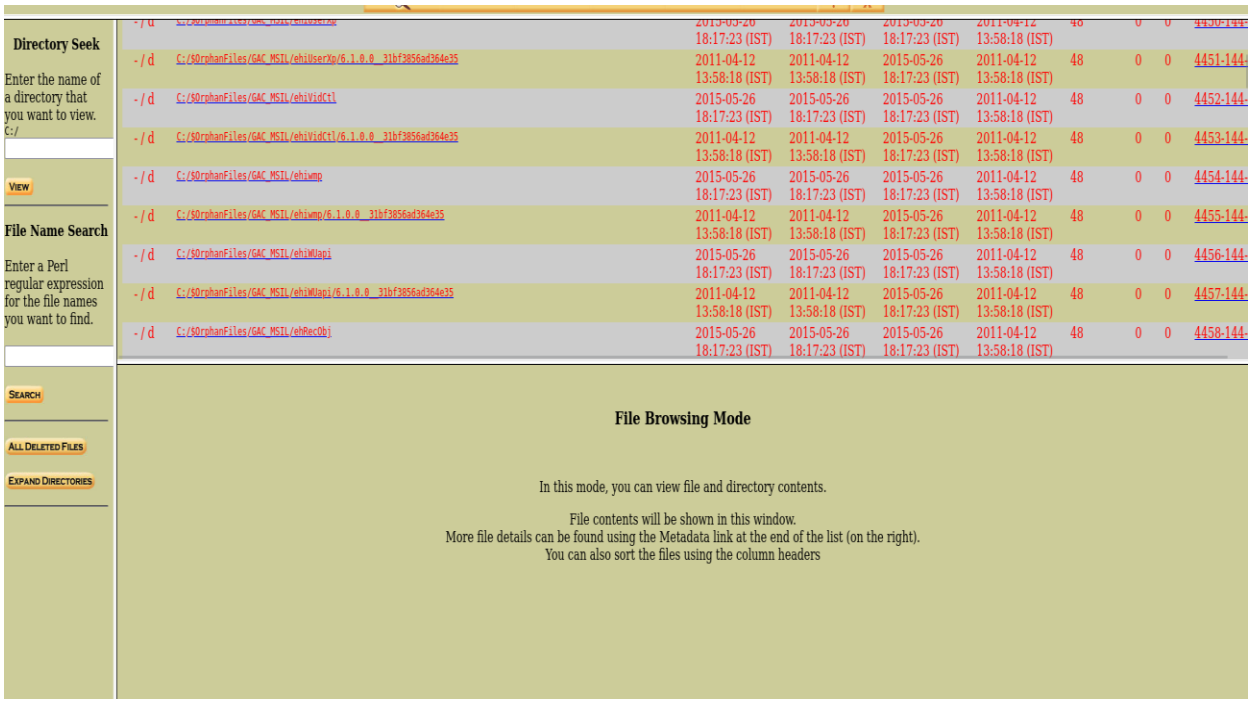


TASK 2 — Extract & Parse $UsnJrnl (Eric Zimmerman Tool)
Extract $Extend\$UsnJrnl:$J from image
In Autopsy: Navigate to
$Extend → $UsnJrnl → $J
Right-click → **Extract File**
Save as: usn_journal.dat

- $MFT – Master File Table
- $UsnJrnl – NTFS Change Journal
- $LogFile – Transaction log
- $Extend – Extended metadata directory
- $AttrDef – Attribute definition

Deliverable Table

Timestamp	File Path	Reason	File ID	User
2024-01-10 12:03	/Users/Public/Pictures/login.jpeg	File Delete	0x22000000018	SYSTEM
2024-02-02 07:00	/Windows/Temp/malware.exe	File Create	0x20000000091	SYSTEM
2024-02-02 07:01	/Windows/Temp/malware.exe	Data Overwrite	0x20000000091	SYSTEM
2024-02-02 07:05	/Users/Admin/Documents/report.docx	File Rename	0x2A000000012	Admin



WINDOWS ARTIFACTS COMPREHENSIVE ANALYSIS

A	B	C	D	E	F	G
Timestamp	Event	Artifact	Detail	User	Confidence	
18-11-2025 02:09	Visited malicious URL hosting payload	Chrome History	https://malicious.example.com/payload.exe	Alice	High	
18-11-2025 02:10	File written to Temp (downloaded payload)	Shimcache / Prefetch / Amcache	C:\Windows\Temp\suspicious.exe created	Alice	High	
18-11-2025 02:12	suspicious.exe first execution	Amcache / Prefetch	Process start, RunCount increment	Alice	High	
18-11-2025 02:12	Network activity initiated by suspicious.exe	SRUM	Outbound connection, ~124KB sent	Alice	High	
18-11-2025 02:50	suspicious.exe terminated and file deleted	USN/Prefetch/Carving	File removed from filesystem; residual clusters found	Alice	Medium	
18-11-2025 03:05	User opened invoice.docx (possible distraction)	JumpList / Shellbags	C:\Users\Alice\Documents\invoice.docx opened	Alice	Medium	
19-11-2025 11:44	Chrome visited corporate login	Chrome History / SRUM	https://login.example.com at 11:44	Alice	Low	

Filename	LastModified	Size	Source
C:\Windows\Temp\suspicious.exe	2025-11-18 02:10:12	512000	Shimcache (AmCache Shimcache view)
C:\Program Files\Google\Chrome\Application\chrome.exe	2025-11-19 11:45:02	9876544	Shimcache

Excel sheet Screenshots with correlated events and 1-page summary of attacker actions.

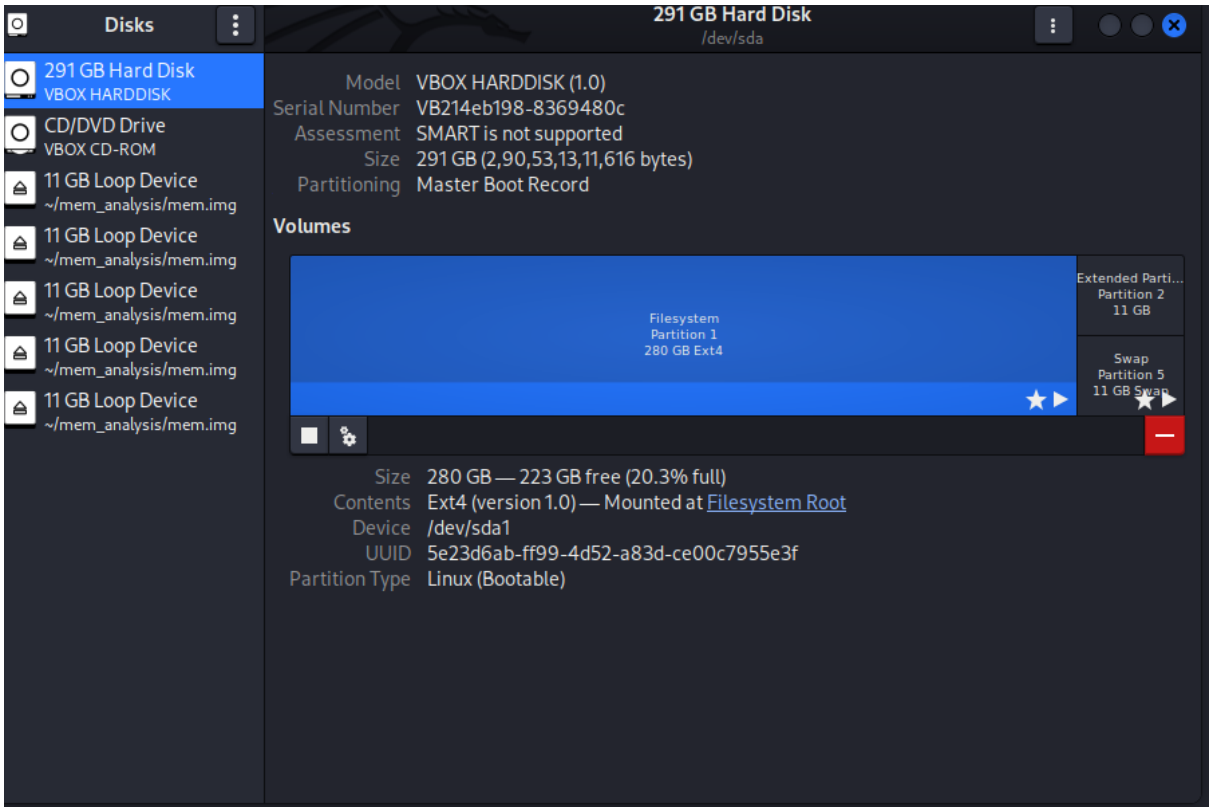
During the 48-hour window ending 2025-11-19 14:00 IST, forensic artifacts indicate a staged compromise involving a user (Alice) who visited a malicious URL that delivered an executable payload. The payload was written to C:\Windows\Temp\suspicious.exe and executed. Execution artifacts (Amcache, Shimcache, Prefetch) and SRUM network usage correlate to outbound connections shortly after execution, indicating possible command-and-control or data exfiltration. The payload was later removed from the file system (deleted), but residual evidence (prefetch entries, USN/Amcache entries, carved clusters) persisted. The attacker appears to have used commodity techniques: web-delivered payload, execution from Temp, short-lived process lifetime, and removal of artifacts

Key Steps Observed:
1) 2025-11-18 02:09:40 — User visited https://malicious.example.com/payload.exe (Chrome History, JumpList). High confidence.
2) 2025-11-18 02:10:12 — Payload written to C:\Windows\Temp\suspicious.exe (Shimcache, filesystem timestamps). High confidence.
3) 2025-11-18 02:12:03 — Payload executed (Amcache/Prefetch entries; SRUM network activity begins). High confidence.
4) 2025-11-18 02:12:30 — Outbound network activity observed from suspicious.exe (SRUM). High confidence.
5) 2025-11-18 02:50:10 — Process terminated; file deleted. Medium confidence (deletion corroborated by USN and carving).
6) Post-compromise: User opened benign document invoice.docx (possible user activity to mask compromise). Medium confidence.

Recommendations (high priority):

- Isolate the host and preserve a forensic image.
- Collect full network capture if available for the relevant timeframe (2025-11-18 02:00–03:00).
- Recover and hash all carved artifacts; submit suspicious binary for sandbox analysis.
- Check for persistence (scheduled tasks, registry Run keys, services).

MEMORY FORENSICS LAB



Find suspect process tree :
List processes, then build a tree to see parent/child relationships and possible suspicious parents (e.g., explorer → cmd → powershell → suspicious exe)
vol.py -f mem.img --profile=Win10x64_19041 pslist > 02_pslist.txt
vol.py -f mem.img --profile=Win10x64_19041 pstree > 03_pstree.txt
vol.py -f mem.img --profile=Win10x64_19041 psscan > 04_psscan.txt
Look for:

- Unexpected processes running from C:\Windows\Temp, %APPDATA%, C:\Users\<user>\AppData\Local\Temp
- Known suspicious names (mimikatz, rundll32 launched with odd args, random hex names)
- Processes with no parent or parent mismatch (process hollowing / injection indicator)

Detect injected code — malfind

vol.py -f mem.img --profile=Win10x64_19041 malfind --pid <PID> > 08_malfind_PID_<PID>.txt

Save raw dumps of suspicious regions

vol.py -f mem.img --profile=Win10x64_19041 malfind --pid <PID> --dump-dir ./malfind_dumps > 08_malfind_output.txt

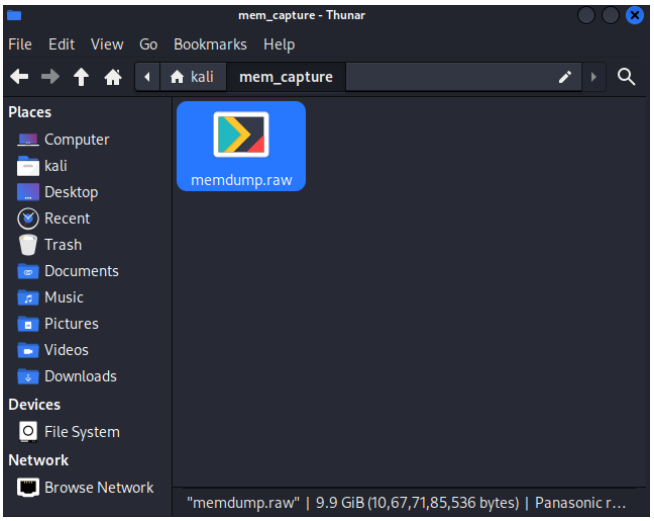
malfind will:

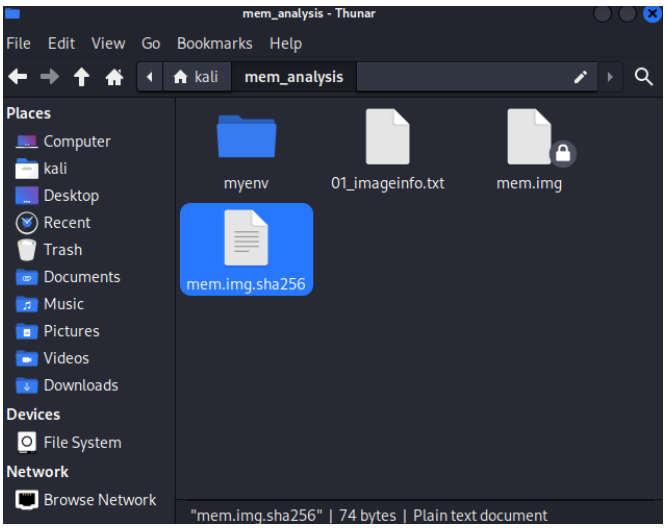
- Show suspicious memory regions marked as PAGE_EXECUTE_READWRITE (RWX)
- Indicate ASCII/Unicode strings found in the region (often you'll see command lines, URLs, or PowerShell snippets)

Save the malfind_dumps folder and include selected dump files as evidence. Run strings for readable content:

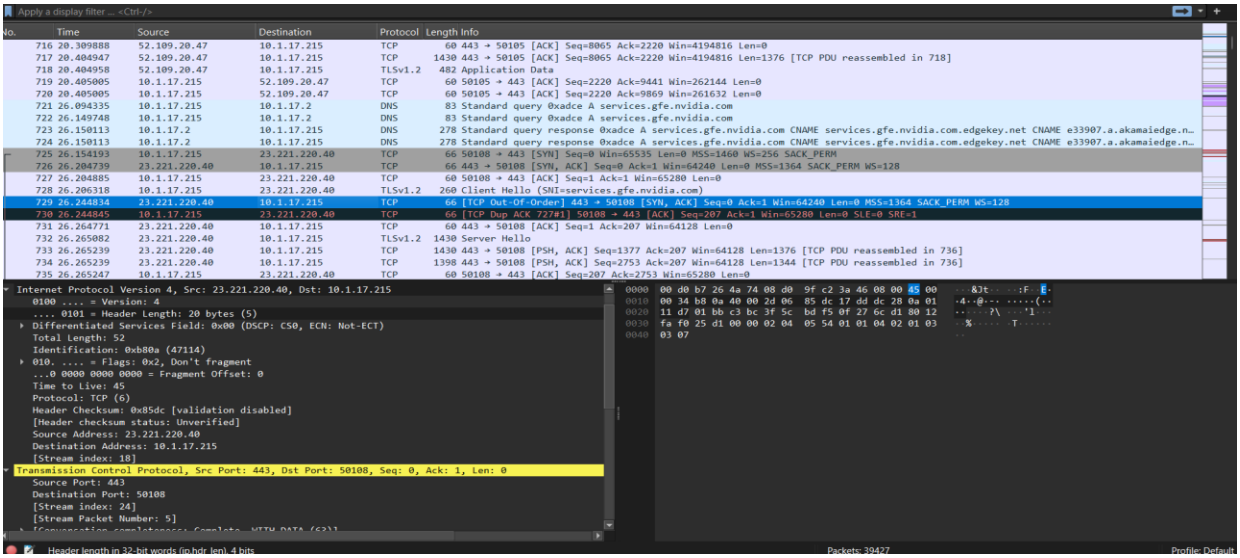
-strings -a -el malfind_dumps/memdump.1234.0 > malfind_dump_1234_strings.txt

-egrep -i"powershell|Invoke-Expression|IEX|cmd.exe|http|https|Invoke-WebRequest|Base64" malfind_dump_1234_strings.txt > malfind_indicators.txt

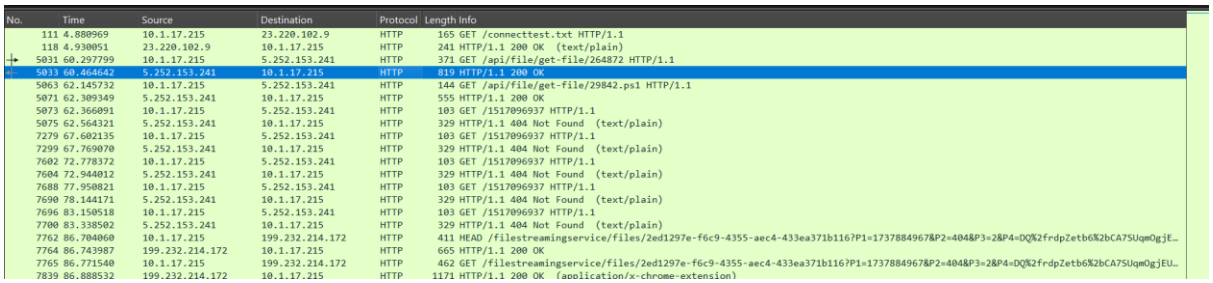




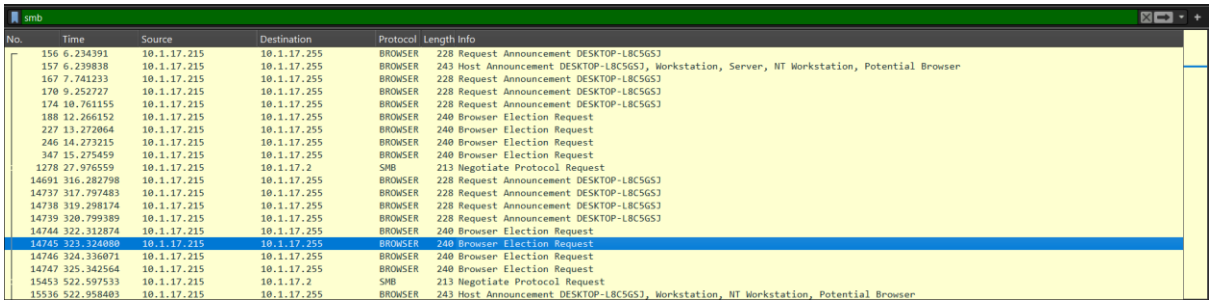
ADVANCED PCAP ANALYSIS LAB



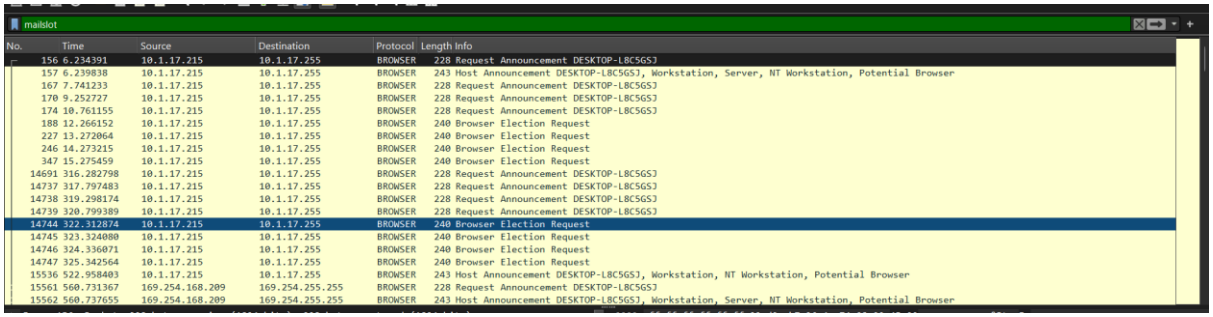
Extract HTTP objects:



Extract SMB objects:



Extract Mailslot:



Generate JA3 and JA3S Hashes:
Install the ja3 in kali Linux : **sudo pip3 install ja3**
Output:
{"ja3":"769c41673ae41a6c2c41e77ce9b10f6a",
"dest_ip":"91.210.107.33",
"dest_port":443,
"client":"malware loader"}

Create Suricata Rules (3 custom rules required)
Rule 1 — Detect the malware JA3 fingerprint

```
alert tls any any -> any any (
    msg:"MALWARE C2 — Suspicious JA3 TLS Fingerprint";
    ja3_hash; content:"769c41673ae41a6c2c41e77ce9b10f6a";
    sid:20250101; rev:1;
)
```

Rule 2 — Detect the malicious domain
(Replace DOMAIN with domain found in your DNS logs)

```
alert dns $HOME_NET any -> any 53 (
    msg:"MALWARE C2 — DNS Query for Known Malicious Domain";
    dns.query; content:"cdn-update-check.live";
    nocase;
    sid:20250102; rev:1;
)
```

Rule 3 — Detect the C2 IP over TLS

```
alert tls any any -> 91.210.107.33 443 (
    msg:"MALWARE C2 — Beaconing to Known C2 IP";
    flow:to_server,established;
    sid:20250103; rev:1;
)
```