



INDIVIDUAL ASSIGNMENT

ASSIGNMENT TITLE	:	Digital Forensics and Cyber Security Tools Final Submission Portfolio
NAME (TP NUMBER)	:	Koo Wai Kit (TP081761)
INTAKE CODE	:	APUMF2406CYS
MODULE TITLE	:	Digital Forensics and Cyber Security Tools
MODULE CODE	:	CT122-0-M-FST
MODULE LECTURER	:	Assoc. Prof. Dr. Jalil bin Md Desa

Table of Contents

I. Introduction	2
II. Portfolio 1: Hex Editor, Wireshark, and Exif Tool	2
Tool 1: Hex Editor.....	2
Task 1	2
Task 2	2
Task 3	4
Task 4.....	7
Tool 2: Wireshark.....	8
Tool 3: Exif Tool.....	13
Image File	13
PDF File	15
III. Portfolio 2: Imaging Tools	16
FTK Imager	16
Guymager	21
Comparison Between FTK Imager and Guymager.....	23
IV. Portfolio 3: Investigation Tools	24
FTK Imager – Windows-based Tool.....	24
Foremost – Linux-based Tool	29
Summary of Case Findings	31
V. Conclusion	32
VI. References	33
VII. Appendix	33

I. Introduction

This portfolio delves into key tools and techniques essential for digital forensics and cybersecurity. It explores practical applications of Hex Editor, Wireshark, Exif Tool, and imaging tools like FTK Imager and Guymager in Kali Linux. We will also look at a case study and use Windows-based and Linux-based investigation tools to analyze an image file in order to find evidence.

Each section provides step-by-step guides accompanied by screenshots, demonstrating how these tools are used for file analysis, network traffic inspection, metadata extraction, and forensic investigations. These tools play a crucial role in safeguarding digital assets and ensuring the integrity of digital evidence, making them indispensable in today's cybersecurity landscape.

II. Portfolio 1: Hex Editor, Wireshark, and Exif Tool

Tool 1: Hex Editor

Task 1

The findings for the different file types and signatures are given in Table 1.

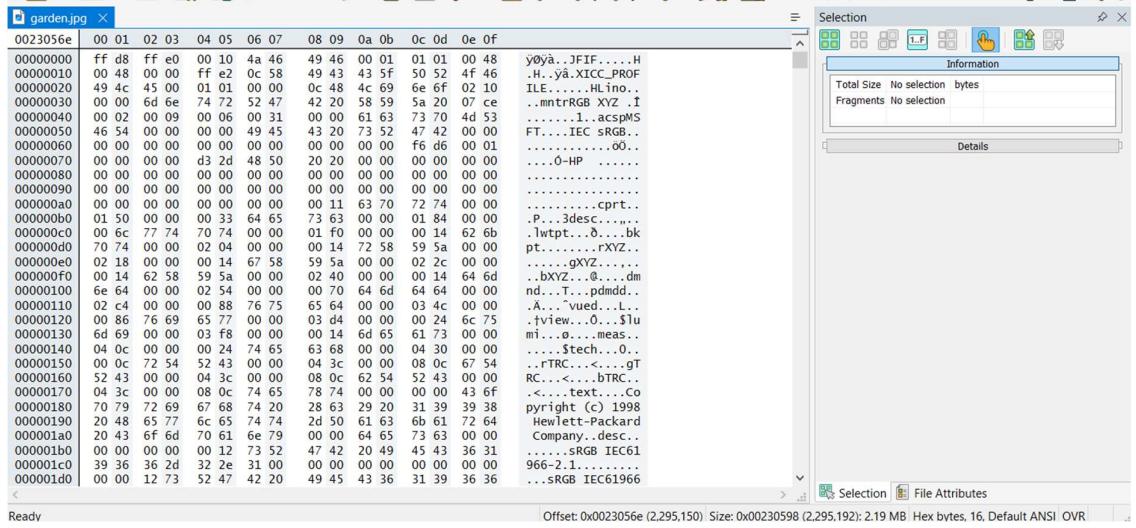
Table 1: File types and signatures

File Type	Hex Signature
.jpg	FF D8
.doc	D0 CF 11 E0 A1 B1 1A E1
.docx	50 4B 03 04 14 00 06 00
.pdf	25 50 44 46
.xls	D0 CF 11 E0 A1 B1 1A E1
.xlsx	50 4B 03 04 14 00 06 00
.ppt	D0 CF 11 E0 A1 B1 1A E1
.pptx	50 4b 03 04 14 00 06 00
mp3	49 44 33

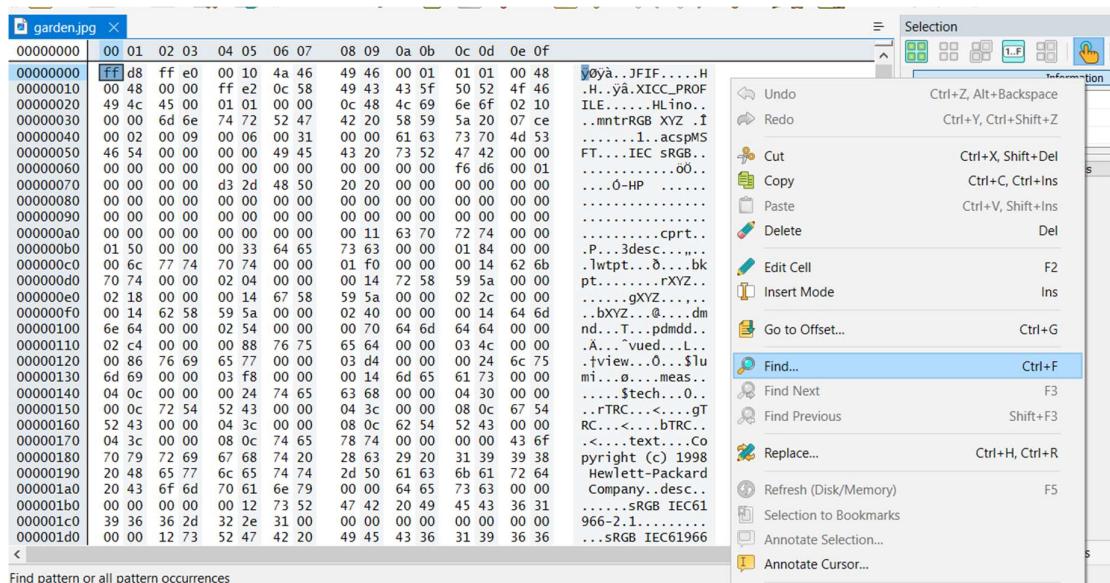
Task 2

Finding flag in garden.jpg:

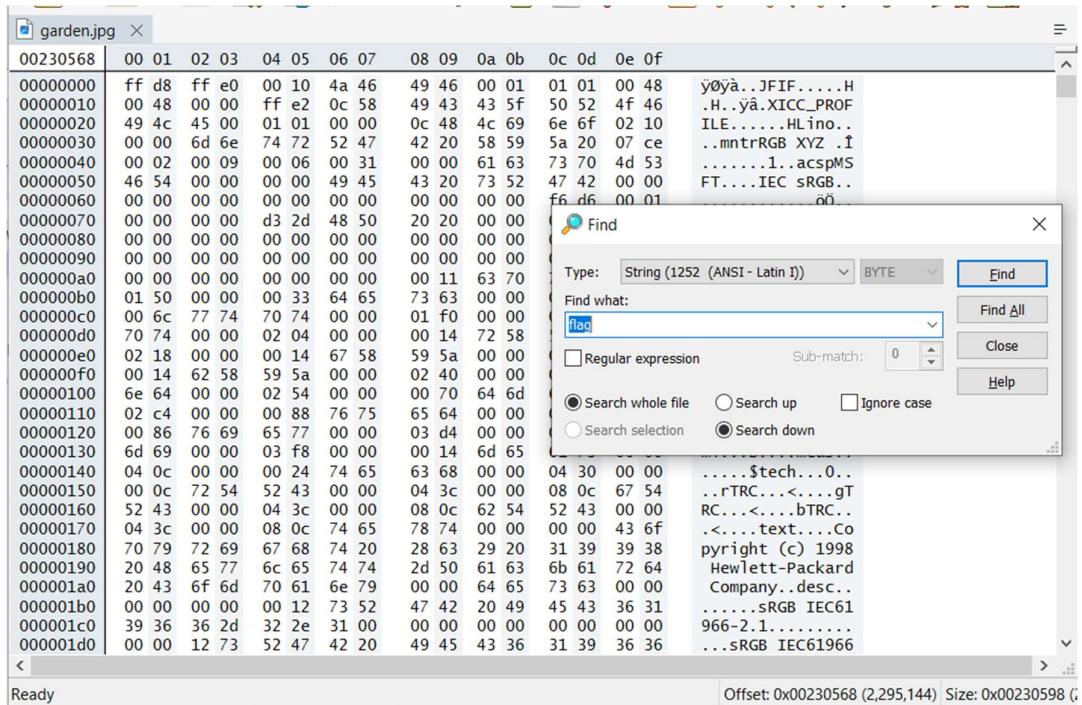
a. To find the flag in the garden.jpg file, first open the image file using the hex editor.



b. Next, we can search for the flag using the Find feature, this can be done by right clicking the interface and selecting Find, or pressing Ctrl + F.



c. We will search for a string that contains the word “flag” in the file.



- d. Once found, we can examine the ASCII view on the right side to determine the ASCII text of the flag.

00230550	a2 bb bd ac	96 87 98 e4	d3 b2 e8 7f	ff d9 48 65	C>`-#`~äó²è.ýÜHe
00230560	72 65 20 69	73 20 61 20	66 6c 61 67	20 22 70 69	re is a flag "pi
00230570	63 6f 43 54	46 7b 6d 6f	72 65 5f 74	68 61 6e 5f	coCTF{more_than_
00230580	6d 33 33 74	73 5f 74 68	65 5f 33 79	33 66 32 30	m33ts_the_3y3f20
00230590	46 35 62 65	39 7d 22 0a	F5be9}".
002305a0					

- e. We can see that the flag is the text enclosed within the double quotation marks.

00230550	a2 bb bd ac	96 87 98 e4	d3 b2 e8 7f	ff d9 48 65	C>`-#`~äó²è.ýÜHe
00230560	72 65 20 69	73 20 61 20	66 6c 61 67	20 22 70 69	re is a flag "pi
00230570	63 6f 43 54	46 7b 6d 6f	72 65 5f 74	68 61 6e 5f	coCTF{more_than_
00230580	6d 33 33 74	73 5f 74 68	65 5f 33 79	33 66 32 30	m33ts_the_3y3f20
00230590	46 35 62 65	39 7d 22 0a	F5be9}".
002305a0					

- f. The flag obtained from garden.jpg is 'picoCTF{more_than_m33ts_the_3y3f20F5be9}'.

Task 3

- a. To find the flag within flag.txt, we have to open the file in a hex editor.

flag.txt

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	89	50	4e	47	0d	0a	1a	0a	00	00	00	0d	49	48	44	52
00000010	00	00	06	a1	00	00	02	60	08	02	00	00	00	85	ad	5e
00000020	0a	00	00	00	01	73	52	47	42	00	ae	ce	1c	e9	00	00
00000030	00	04	67	41	4d	41	00	00	b1	8f	0b	fc	61	05	00	00
00000040	00	09	70	48	59	73	00	00	16	25	00	00	16	25	01	49
00000050	52	24	f0	00	00	26	95	49	44	41	54	78	5e	ed	dd	6b
00000060	42	1b	39	b7	05	d0	3b	2e	06	94	f1	30	9a	4c	26	83
00000070	f9	ae	5f	80	4e	3d	25	bb	4c	b3	f1	5a	bf	ba	a1	4a
00000080	75	74	24	13	79	27	c0	ff	fd	0f	00	00	00	48	26	utS.yAyy..H&
00000090	e3	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03
000000a0	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	
000000b0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	00	
000000c0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	00	
000000d0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	00	
000000e0	80	6c	32	3e	00	00	00	00	c8	26	e3	03	00	00	00	80
000000f0	6c	32	3e	00	00	00	c8	26	e3	03	00	00	00	80	6c	
00000100	32	3e	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	
00000110	3e	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	
00000120	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	
00000130	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	
00000140	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	00	
00000150	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	00	
00000160	c8	26	e3	03	00	00	80	6c	32	3e	00	00	00	c8	26	
00000170	26	e3	03	00	00	80	6c	32	3e	00	00	00	c8	26	ë&	
00000180	e3	03	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	
00000190	03	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001a0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001b0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001c0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001d0	80	6c	32	3e	00	00	00	c8	26	e3	03	00	00	80		

Ready

Offset: 0x00000000 (0) Size: 0x00002700 (9.984) 9.75 KB Hex bytes, 16, Default ANSI OVR

b. First, we examine the file signature and determine if this file is using the correct file extension. After checking, we realized that the correct file type is actually a PNG file, but this file was saved as a .txt file.

flag.txt

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	89	50	4e	47	0d	0a	1a	0a	00	00	00	0d	49	48	44	52
00000010	00	00	06	a1	00	00	02	60	08	02	00	00	00	85	ad	5e
00000020	0a	00	00	00	01	73	52	47	42	00	ae	ce	1c	e9	00	00
00000030	00	04	67	41	4d	41	00	00	b1	8f	0b	fc	61	05	00	00
00000040	00	09	70	48	59	73	00	00	16	25	00	00	16	25	01	49
00000050	52	24	f0	00	00	26	95	49	44	41	54	78	5e	ed	dd	6b
00000060	42	1b	39	b7	05	d0	3b	2e	06	94	f1	30	9a	4c	26	83
00000070	f9	ae	5f	80	4e	3d	25	bb	4c	b3	f1	5a	bf	ba	a1	4a
00000080	75	74	24	13	79	27	c0	ff	fd	0f	00	00	00	48	26	utS.yAyy..H&
00000090	e3	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03
000000a0	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	
000000b0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000000c0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000000d0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
00000100	32	3e	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	
00000110	3e	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	
00000120	00	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	
00000130	00	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	
00000140	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	00	
00000150	00	00	c8	26	e3	03	00	00	00	80	6c	32	3e	00	00	
00000160	c8	26	e3	03	00	00	00	80	6c	32	3e	00	00	00	c8	
00000170	26	e3	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	
00000180	e3	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	
00000190	03	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03	
000001a0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001b0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001c0	00	00	00	80	6c	32	3e	00	00	00	c8	26	e3	03		
000001d0	80	6c	32	3e	00	00	00	c8	26	e3	03	00	00	80		

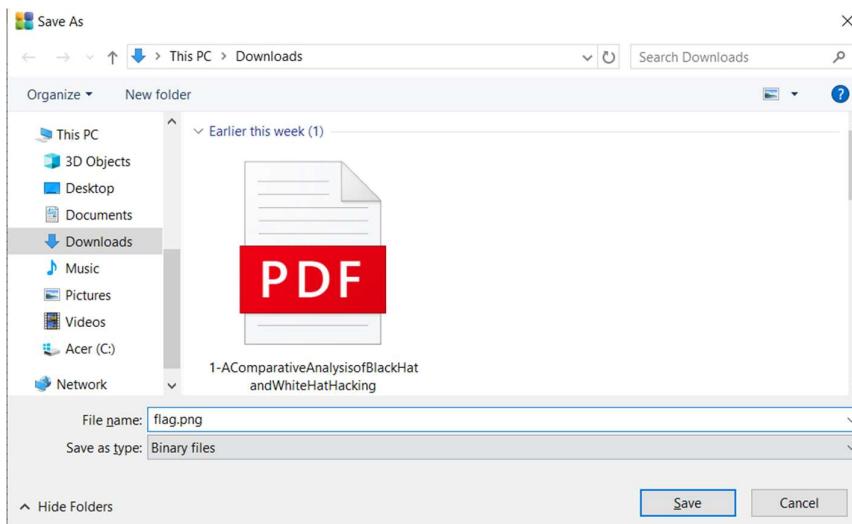
Ready

Offset: 0x00000015 (27) Size: 0x00002700 (9.984) 9.75 KB Hex bytes, 16, Default ANSI OVR

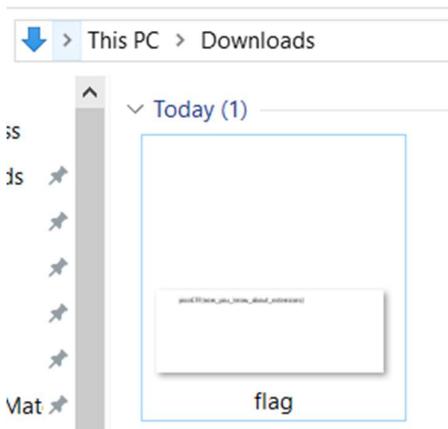
c. To obtain the flag, we have to save the file using the correct file extension. Use the ‘Save As’ function to save the file.



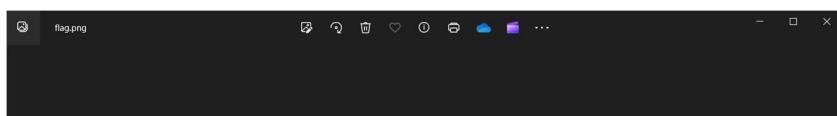
d. The file can be saved as a PNG file by adding “.png” after the file name.



e. After saving the file, the file can now be viewed as an image file.

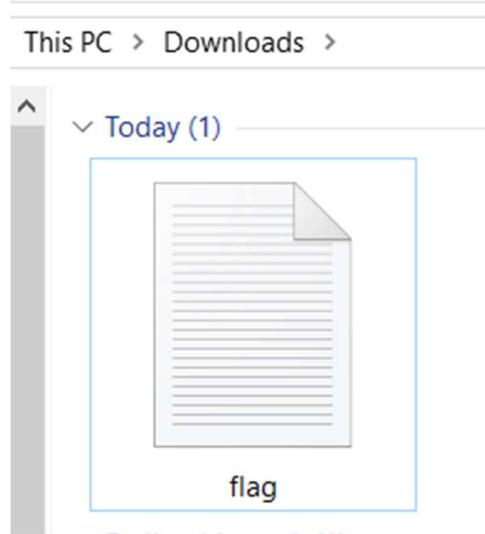


f. By opening the file, we can now obtain the flag.



Task 4

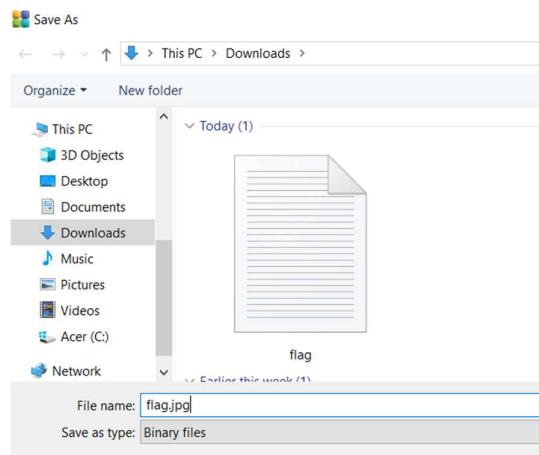
- a. This is a .txt file containing a flag.



- b. First, we open the file and examine if it is saved in the correct file extension,

00000123	00 01 02 03	04 05 06 07	08 09 0a 0b	0c 0d 0e 0f	00000123	00 01 02 03	04 05 06 07	08 09 0a 0b	0c 0d 0e 0f	00000123	00 01 02 03	04 05 06 07	08 09 0a 0b	0c 0d 0e 0f
00000000	ff d8 ff e0	00 10 4a 46	49 46 00 01	01 00 00 01	00000000	ff d8 ff e0	00 10 4a 46	49 46 00 01	01 00 00 01	00000000	ff d8 ff e0	00 10 4a 46	49 46 00 01	01 00 00 01
00000010	00 01 00 00	ff db 00 84	00 09 06 07	13 13 12 15	00000010	00 01 00 00	ff db 00 84	00 09 06 07	13 13 12 15	00000010	00 01 00 00	ff db 00 84	00 09 06 07	13 13 12 15
00000020	13 13 13 15	16 15 17 19	18 17 18 18	17 17 18 18	00000020	13 13 13 15	16 15 17 19	18 17 18 18	17 17 18 18	00000020	13 13 13 15	16 15 17 19	18 17 18 18	17 17 18 18
00000030	19 1a 1e 1a	18 16 1b 1d	1a 1f 18 1e	1d 28 20 1a	00000030	19 1a 1e 1a	18 16 1b 1d	1a 1f 18 1e	1d 28 20 1a	00000030	19 1a 1e 1a	18 16 1b 1d	1a 1f 18 1e	1d 28 20 1a
00000040	1b 25 1f 1d	18 22 31 21	25 29 2b 2e	2e 2e 19 20	00000040	1b 25 1f 1d	18 22 31 21	25 29 2b 2e	2e 2e 19 20	00000040	1b 25 1f 1d	18 22 31 21	25 29 2b 2e	2e 2e 19 20
00000050	33 38 33 2d	37 28 2d 2e	2d 01 0a 0a	0a 0e 0d 0e	00000050	33 38 33 2d	37 28 2d 2e	2d 01 0a 0a	0a 0e 0d 0e	00000050	33 38 33 2d	37 28 2d 2e	2d 01 0a 0a	0a 0e 0d 0e
00000060	1b 10 10 1b	2d 25 20 25	2d 2d 2f 2d	2d 2d 2d 2d	00000060	1b 10 10 1b	2d 25 20 25	2d 2d 2f 2d	2d 2d 2d 2d	00000060	1b 10 10 1b	2d 25 20 25	2d 2d 2f 2d	2d 2d 2d 2d
00000070	2d 2d 2d 2d	00000070	2d 2d 2d 2d	00000070	2d 2d 2d 2d									
00000080	2d 2d 2d 2d	2d 2d 2d 35	2d 2d 2d 2d	2d 2d 2d 2d	00000080	2d 2d 2d 2d	2d 2d 2d 35	2d 2d 2d 2d	2d 2d 2d 2d	00000080	2d 2d 2d 2d	2d 2d 2d 35	2d 2d 2d 2d	2d 2d 2d 2d

- c. From the file signature, we know that this file is supposed to be a JPG file. Next, we have to save this file using the .jpg extension.



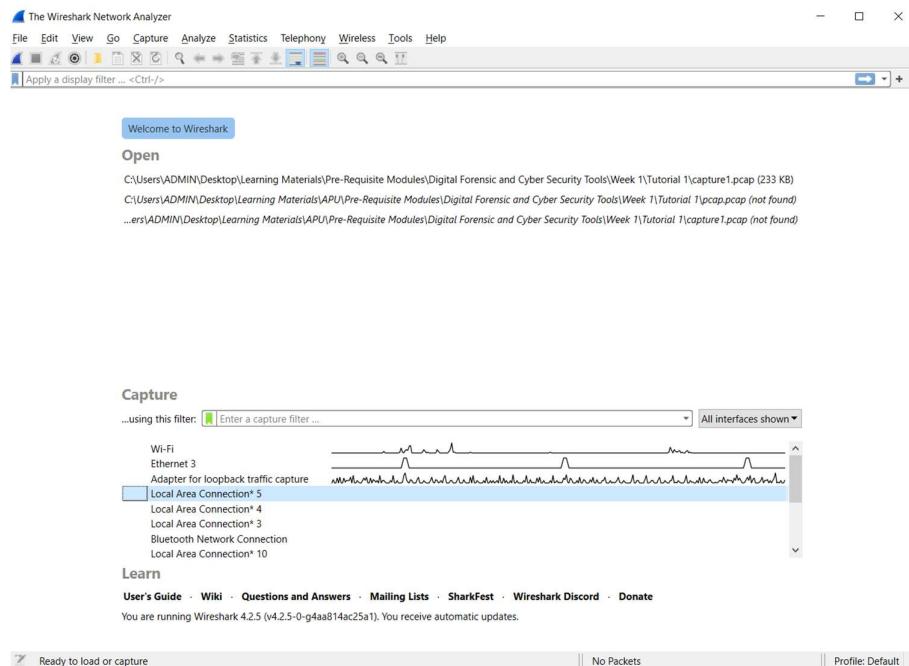
- d. Once saved, the file now can be opened as an image file, in which the flag can be obtained.



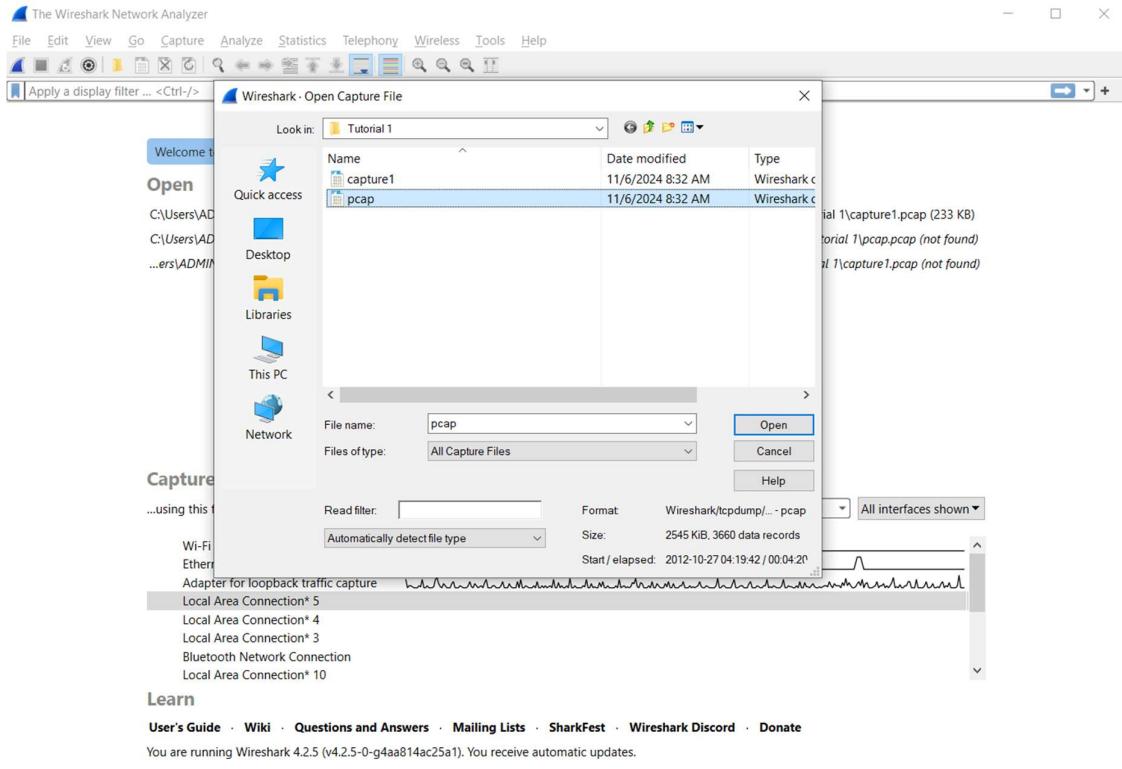
- e. The flag contained within the file is '16 9 3 15 3 20 6 { 20 8 5 14 21 13 2 5 18 19 13 1 19 15 14 }'.

Tool 2: Wireshark

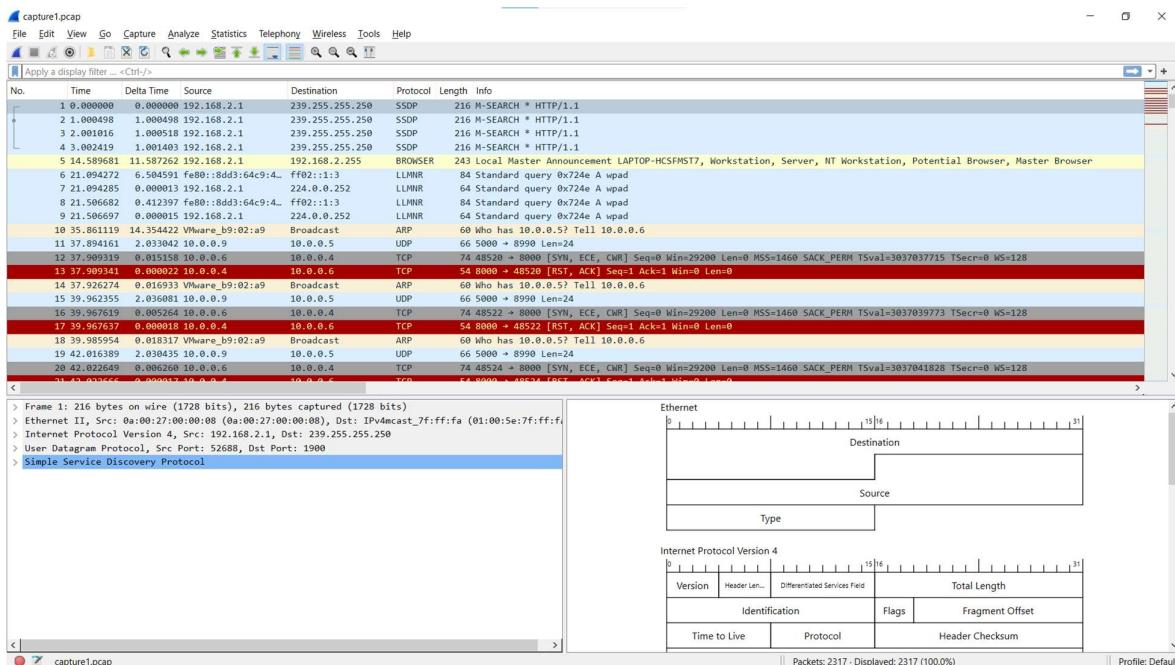
- a. To analyze the given pcap file, first we have to open the Wireshark Network Analyzer.



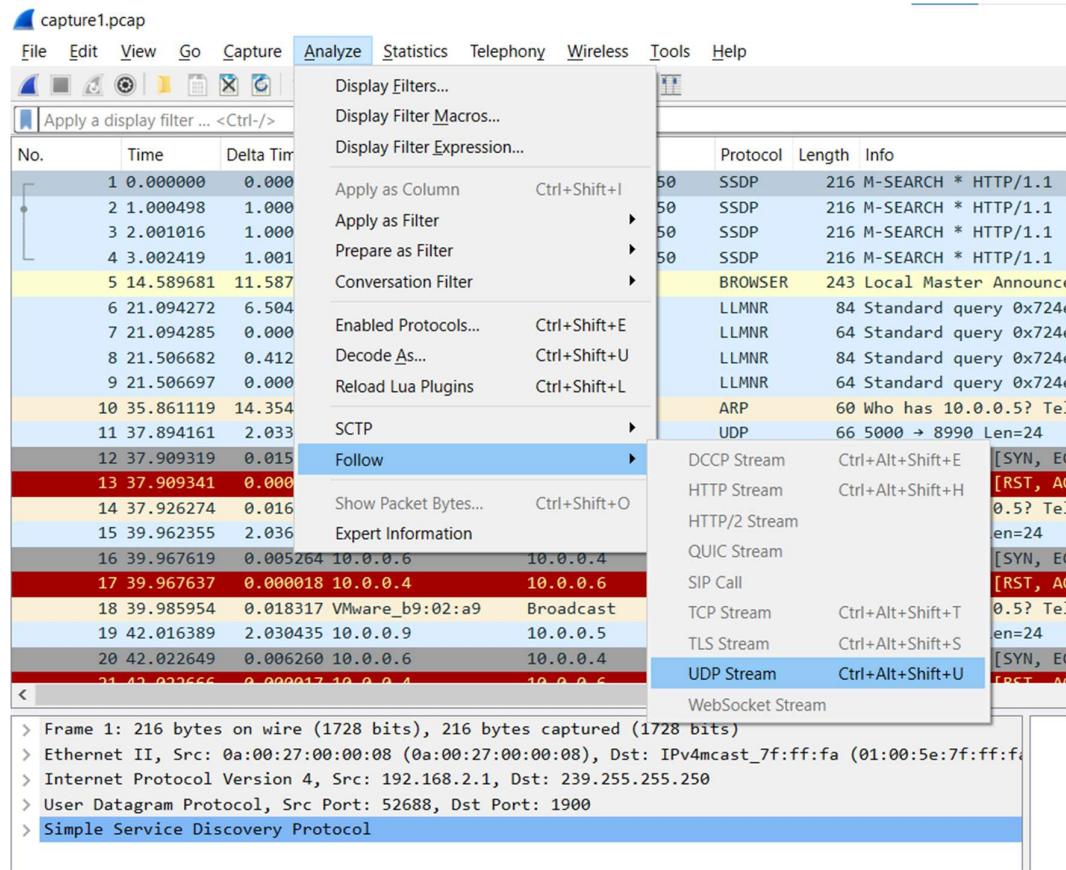
- b. Then, look for the pcap file named 'capture1.pcap'.



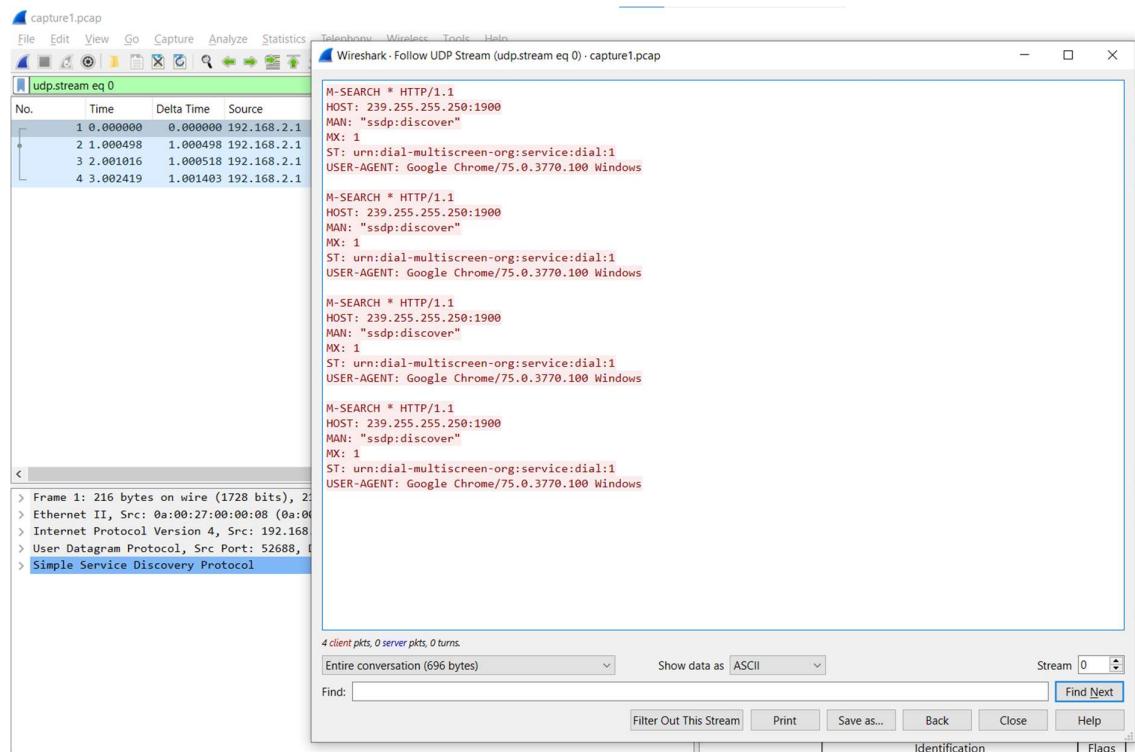
c. Once the file is opened, it will show this interface:



d. To find the flag, we have to analyze the file and follow the UDP stream.



e. This window will pop up, which represents UDP stream 0.



f. We follow to the next streams until we find the flag.

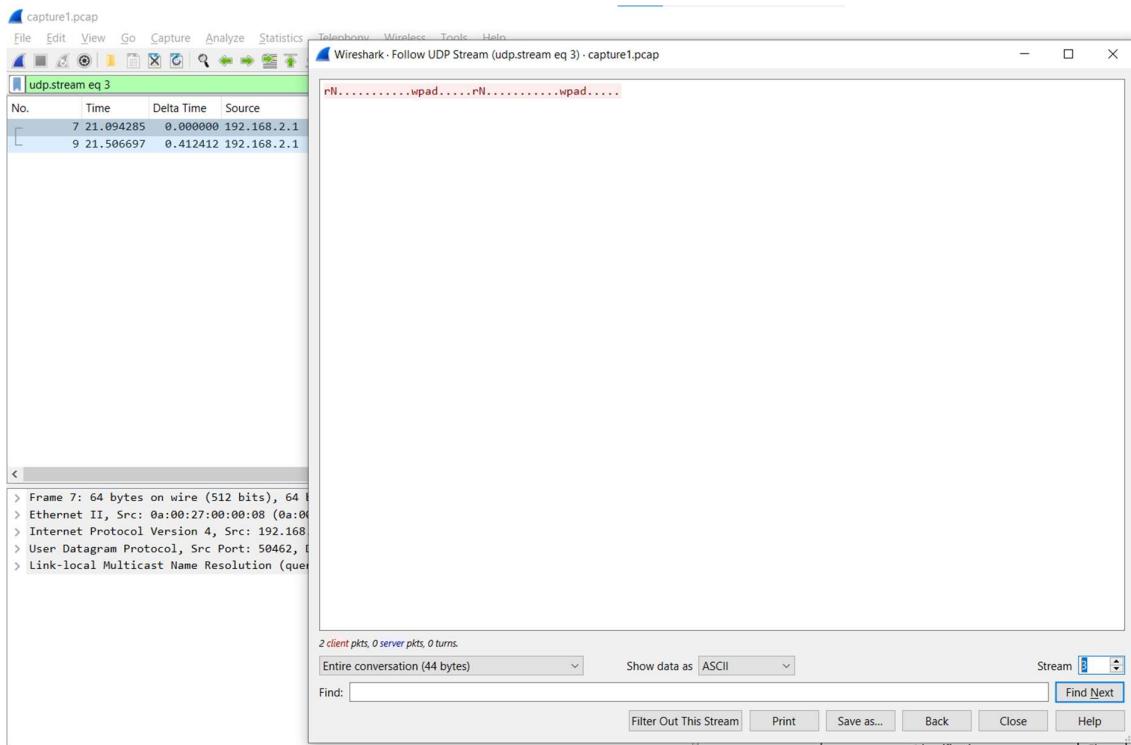
The image consists of two side-by-side screenshots of the Wireshark application interface. Both screenshots show a main window for 'capture1.pcap' and a detailed view of a selected stream.

Screenshot 1 (Left):

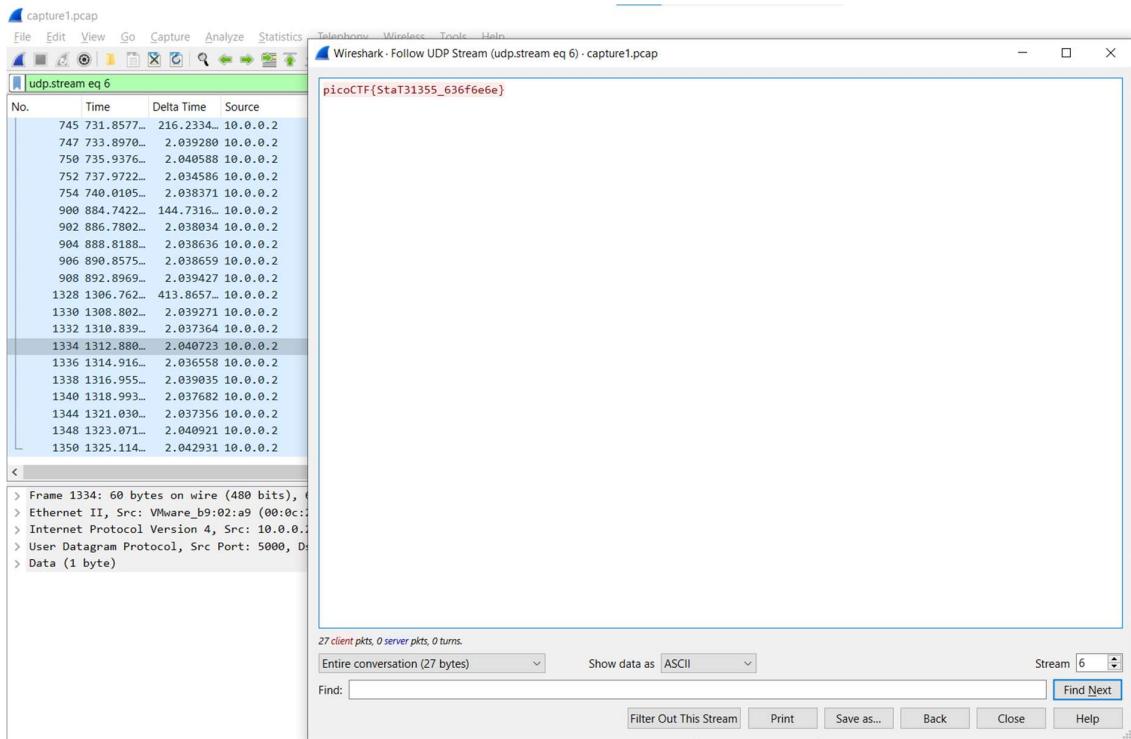
- Title Bar:** capture1.pcap
- Menu Bar:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help
- Toolbar:** Standard icons for opening files, capturing, analyzing, and saving.
- Table View:** Shows a list of captured frames. Frame 5 is highlighted in yellow. The table columns are No., Time, Delta Time, and Source.
- Followed Stream View:** A large pane titled 'Wireshark - Follow UDP Stream (udp.stream eq 1) · capture1.pcap'. It displays the raw hex and ASCII data for the selected stream. The ASCII dump shows repeated SMB traffic between 'LAPTOP-HCSFMST7' and '2.\MAILSLOT\BROWSE....'.
- Frame Details:** A scrollable list at the bottom left provides details for Frame 5, including its size (243 bytes), protocol stack (Ethernet II, User Datagram Protocol, SMB), and source/destination information.
- Bottom Buttons:** Find, Filter Out This Stream, Print, Save as..., Back, Close, Help.

Screenshot 2 (Right):

- Title Bar:** capture1.pcap
- Menu Bar:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help
- Toolbar:** Standard icons for opening files, capturing, analyzing, and saving.
- Table View:** Shows a list of captured frames. Frame 6 is highlighted in yellow. The table columns are No., Time, Delta Time, and Source.
- Followed Stream View:** A large pane titled 'Wireshark - Follow UDP Stream (udp.stream eq 2) · capture1.pcap'. It displays the raw hex and ASCII data for the selected stream. The ASCII dump shows the string 'rN.....wpad.....rN.....wpad....'.
- Frame Details:** A scrollable list at the bottom left provides details for Frame 6, including its size (84 bytes), protocol stack (Ethernet II, User Datagram Protocol, Link-local Multicast Name Resolution), and source/destination information.
- Bottom Buttons:** Find, Filter Out This Stream, Print, Save as..., Back, Close, Help.



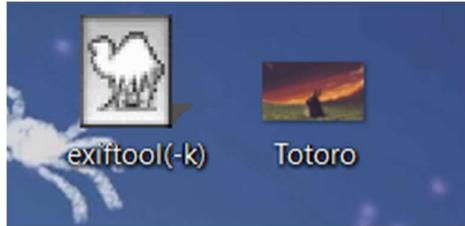
g. On stream number 6, we found the flag, which is ‘picoCTF{StaT31355_636f6e6e}’.



Tool 3: Exif Tool

Image File

- To obtain the metadata from an image file, first I saved the ExifTool executable and a JPG image file in the same directory, which is on my Desktop directory.



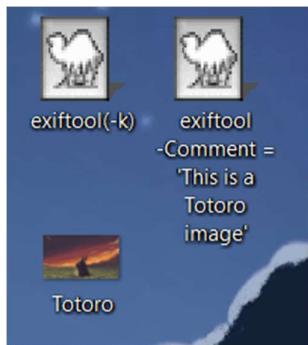
- Next, I can drag the image file to overlap with the exiftool icon. This gives me the option to open the image file with exiftool.



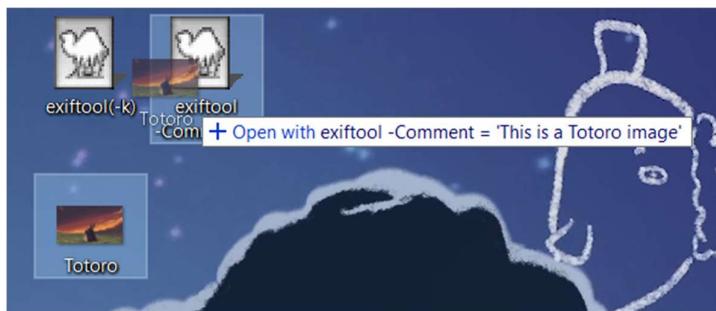
- Then, the exiftool executable will open automatically and display the metadata of the image file.

```
C:\Users\ADMIN\Desktop\exiftool(-k).exe
ExifTool Version Number : 19.86
File Name   : Totoro.jpg
Directory  : C:/Users/ADMIN/Desktop
File Size   : 1774 kB
File Identifier : Exists
File Modification Date/Time : 2020:10:10 23:12:41+08:00
File Access Date/Time : 2020:06:28 15:00:03+08:00
File Creation Date/Time : 2020:06:28 14:52:03+08:00
File Permissions : rwxrwxr-
File Type    : JPEG
File Type Extension : jpg
MIME Type   : image/jpeg
JFIF Version : 1.0
Current IPTC Digest : e471fc9b771b9f3e47046e891059c20d
Code Character Set : UTF8
Application Record Version : 0
IPTC Digest : e471fc9b771b9f3e47046e891059c20d
X Resolution : 300
Displayed Units X : inches
Resolution : 300
Displayed Units Y : inches
Print Style : Centered
Print Position : 0 0
Print Scale : 1
Global Angle : 120
Global Altitude : 30
File List : fallwall
Slices Group Name : 1
Num Slices : 1
Pixel Aspect Ratio : 1
Photoshop Thumbnail : (Binary data 7032 bytes, use -b option to extract)
Has Real Merged Data : Yes
Profile Name : Adobe Photoshop
Reader Name : Adobe Photoshop CS6
Photoshop Quality : 12
Photoshop Format : Progressive
Progressive Scans : 3 Scans
Exif Byte Order : Big-endian (Motorola, MM)
Print Coloration : Unspecified (0)
Resolution Unit : inches
Software : Adobe Photoshop CS6 (Macintosh)
Modify Date : 2013:10:09 15:38:16
Color Space : sRGB
Exif Image Width : 2560
Exif Image Height : 1440
Compression : JPEG (old-style)
Thumbnail Offset : 9808
Thumbnail Length : 7032
Profile CMYK Type : Linetronic
Profile Version : 2.1.0
Profile Class : Display Device Profile
Color Space Data : RGB
```

- d. To add a comment to the image file, we can create a copy of the exiftool executable and change the argument tag -Comment to add a comment.



- e. Next, drag and drop the image file on top of the executable with the comment argument.



- f. A window will pop up and close automatically when the update is performed, we can then check if the comment is added by dragging and dropping the image file on the first exiftool executable without the argument tag.

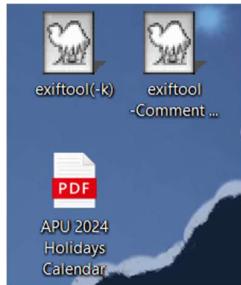


- g. When inspecting the metadata, we can now see the newly added comment to the file.

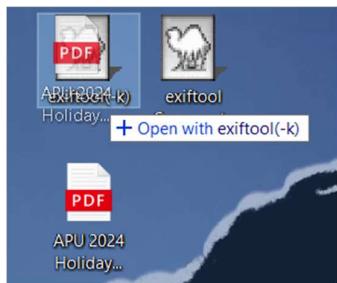
```
Select C:\Users\ADMIN\Desktop\exiftool(-k).exe
Document Ancestors           : 260B1F34F4D693EA3C6DF7CA
5B2BB0294673C57C2DD19FBD6, xmp.did:7561008333206811822AB9D
Comment                      : This is a Totoro image
```

PDF File

- a. The same steps apply for a PDF file. First, the PDF file and exiftool executable are saved within the same directory.



- b. Next, drag and drop the PDF file onto the exiftool executable.



- c. A window will pop up automatically and we can now view the metadata of the PDF file.

```
C:\Users\ADMIN\Desktop\exiftool(-k).exe
ExifTool Version Number      : 12.86
File Name                   : APU 2024 Holidays Calendar.pdf
Directory                   : C:/Users/ADMIN/Desktop
File Size                    : 358 kB
Zone Identifier              : Exists
File Modification Date/Time : 2024:05:21 22:22:09+08:00
File Access Date/Time       : 2024:06:28 15:13:30+08:00
File Creation Date/Time    : 2024:06:28 15:13:29+08:00
File Permissions             : -rw-rw-rw-
File Type                   : PDF
File Type Extension         : pdf
MIME Type                   : application/pdf
PDF Version                 : 1.5
Linearized                  : Yes
XMP Toolkit                 : Adobe XMP Core 9.0-c001 79.14ecb42, 2022/12/02-19:12:44
Format                      : application/pdf
Title                       : 202311-03 PH-Schedule Leave 2024(Student)-D1
Metadata Date               : 2023:12:21 14:47:35+08:00
Modify Date                 : 2023:12:21 14:47:35+08:00
Create Date                 : 2023:12:21 14:47:35+08:00
Creator Tool                : Adobe Illustrator 27.4 (Macintosh)
Thumbnail Width             : 256
Thumbnail Height            : 188
Thumbnail Format            : JPEG
Thumbnail Image              : (Binary data 18803 bytes, use -b option to extract)
Instance ID                 : uuid:6faf2842-df14-2e49-8ada-0237db35c937
Document ID                 : xmp.did:c7fb31e8-c946-4da0-85d4-457a0edc3f9b
Original Document ID       : uid:5D20892493BFDB11914A8590D31508C8
Rendition Class             : proof:pdf
Derived From Instance ID    : uid:5d3e37cb-1d16-6843-aabe-d480fa2f0279
Derived From Document ID   : xmp.did:a170ed09-1a77-43d7-b045-978aab46b8e5
Derived From Original Document ID: uid:5D20892493BFDB11914A8590D31508C8
Derived From Rendition Class: default
History Action               : saved, saved
```

- d. To add a comment to the PDF file, I will use the exiftool executable copy that I created earlier, but the argument tag is changed to -keywords. The keyword will act as the comment of the file.



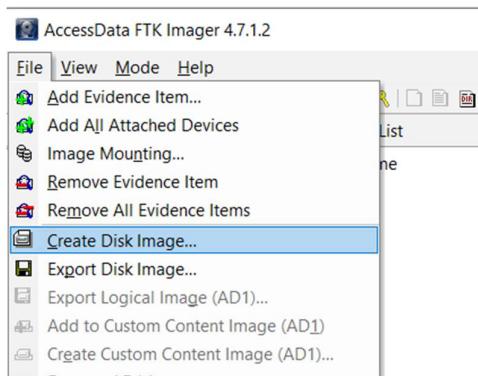
- h. Drag and drop the PDF file on top of the executable with the 'keywords' argument.



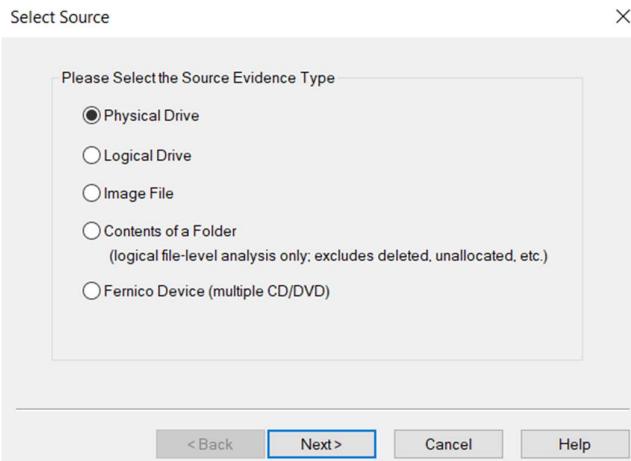
III. Portfolio 2: Imaging Tools

FTK Imager

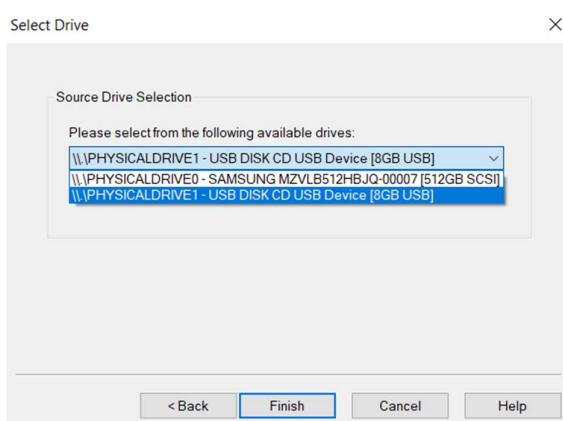
- a. To create a disk image of a USB drive, we can use the 'Create Disk Image' function of FTK Imager.



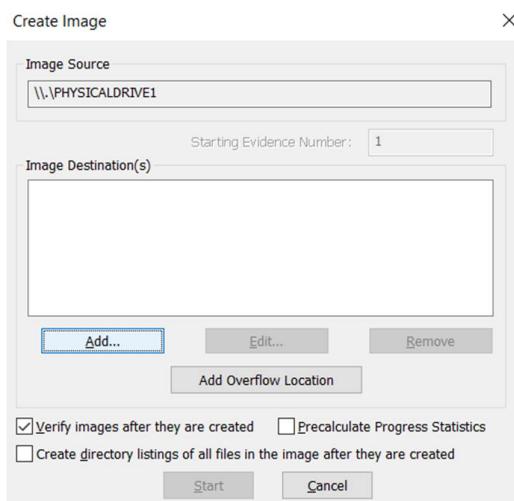
- b. Then, choose the physical drive as the source type of the image.



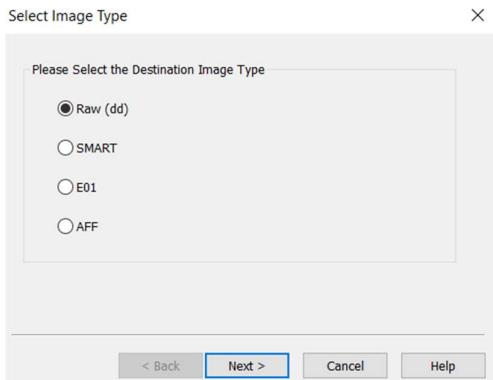
c. The USB drive is then chosen from the list of available drives.



d. The details of the process to create the image will be displayed. We will add an image destination to store the image files.



e. The destination image will be saved as type Raw (dd).



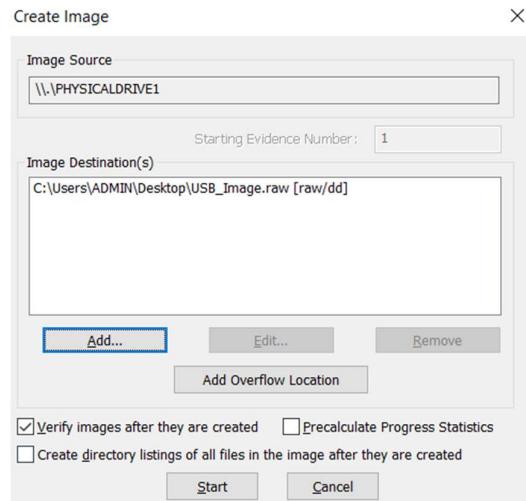
f. Next, the case number, evidence number, and examiner name are provided.

The screenshot shows a dialog box titled "Evidence Item Information". It contains five input fields: "Case Number" with value "001", "Evidence Number" with value "001", "Unique Description" (empty), "Examiner" with value "Wai Kit" (which is highlighted with a blue border), and "Notes" (empty). At the bottom are four buttons: "< Back", "Next >" (which is highlighted with a blue border), "Cancel", and "Help".

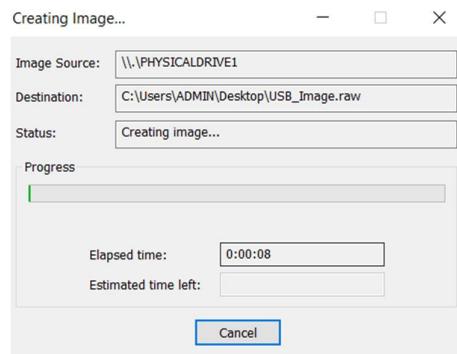
g. Then, we choose the destination folder and the filename to save the image. The image fragment size will be 1500 MB.

The screenshot shows a dialog box titled "Select Image Destination". It has several input fields: "Image Destination Folder" with value "C:\Users\ADMIN\Desktop" and a "Browse" button; "Image Filename (Excluding Extension)" with value "USB_Image.raw"; "Image Fragment Size (MB)" with value "1500" and a note "For Raw, E01, and AFF formats: 0 = do not fragment"; "Compression (0=None, 1=Fastest, ..., 9=Smallest)" with value "0"; and a "Use AD Encryption" checkbox which is unchecked. At the bottom are four buttons: "< Back", "Finish" (which is highlighted with a blue border), "Cancel", and "Help".

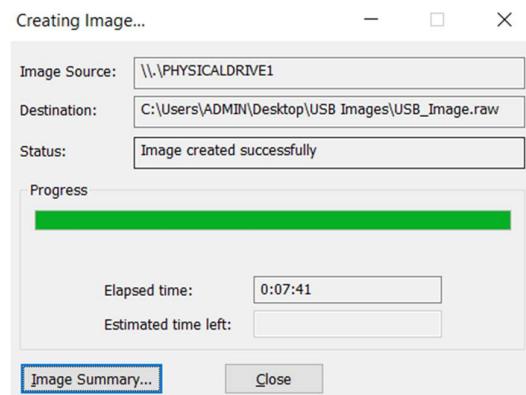
h. Once the destination is provided, we can now start creating the image.



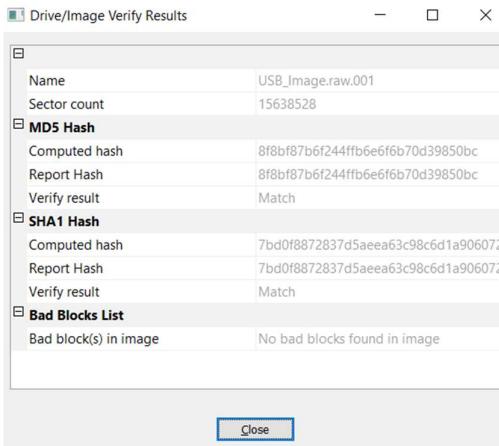
- i. This window will show the progress and time elapsed of the imaging process.



- j. The imaging of this USB drive took 7 minutes and 41 seconds to complete.



- k. The image results are verified and there are no imaging errors.



1. The image summary is as follows:

Image Summary

Created By AccessData® FTK® Imager 4.7.1.2

Case Information:
Acquired using: AD4.7.1.2
Case Number: 001
Evidence Number: 001
Unique Description:
Examiner: Wai Kit
Notes:

Information for C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw:

Physical Evidentiary Item (Source) Information:

- [Device Info]
- Source Type: Physical
- [Drive Geometry]
- Cylinders: 973
- Tracks per Cylinder: 255
- Sectors per Track: 63
- Bytes per Sector: 512
- Sector Count: 15,638,528
- [Physical Drive Information]
- Drive Model: USB DISK CD USB Device
- Drive Serial Number: 07082894401C7401

Image Summary

Drive Serial Number: 07082894401C7401
Drive Interface Type: USB
Removable drive: True
Source data size: 7636 MB
Sector count: 15638528
[Computed Hashes]
MD5 checksum: 8f8bf87b6f244ffb6e6f6b70d39850bc
SHA1 checksum: 7bd0f8872837d5aeeaa63c98c6d1a9060722a3db9

Image Information:
Acquisition started: Fri Jun 28 16:28:09 2024
Acquisition finished: Fri Jun 28 16:35:50 2024
Segment list:
C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw.001
C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw.002
C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw.003
C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw.004
C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw.005
C:\Users\ADMIN\Desktop\USB Images\USB_Image.raw.006

Image Verification Results:
Verification started: Fri Jun 28 16:35:52 2024
Verification finished: Fri Jun 28 16:36:19 2024
MD5 checksum: 8f8bf87b6f244ffb6e6f6b70d39850bc : verified
SHA1 checksum: 7bd0f8872837d5aeeaa63c98c6d1a9060722a3db9 : verified

OK

m. There are a total of 6 image files created and saved within the directory, with the addition of one information file about the imaging.

```
C:\Users\ADMIN>cd C:\Users\ADMIN\Desktop\USB Images
C:\Users\ADMIN\Desktop\USB Images>dir
Volume in drive C is Acer
Volume Serial Number is 083C-B766

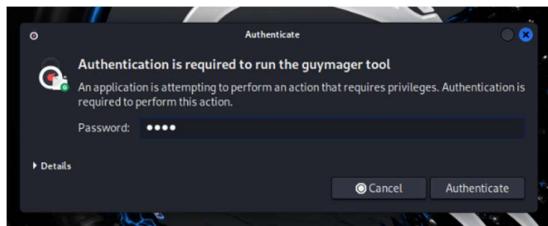
Directory of C:\Users\ADMIN\Desktop\USB Images

28/06/2024  04:35 PM    <DIR>      .
28/06/2024  04:35 PM    <DIR>      ..
28/06/2024  04:29 PM  1,572,864,000 USB_Image.raw.001
28/06/2024  04:36 PM          1,632 USB_Image.raw.001.txt
28/06/2024  04:31 PM  1,572,864,000 USB_Image.raw.002
28/06/2024  04:32 PM  1,572,864,000 USB_Image.raw.003
28/06/2024  04:34 PM  1,572,864,000 USB_Image.raw.004
28/06/2024  04:35 PM  1,572,864,000 USB_Image.raw.005
28/06/2024  04:35 PM          142,606,336 USB_Image.raw.006
                           7 File(s)   8,006,927,968 bytes
                           2 Dir(s)  141,518,319,616 bytes free
```

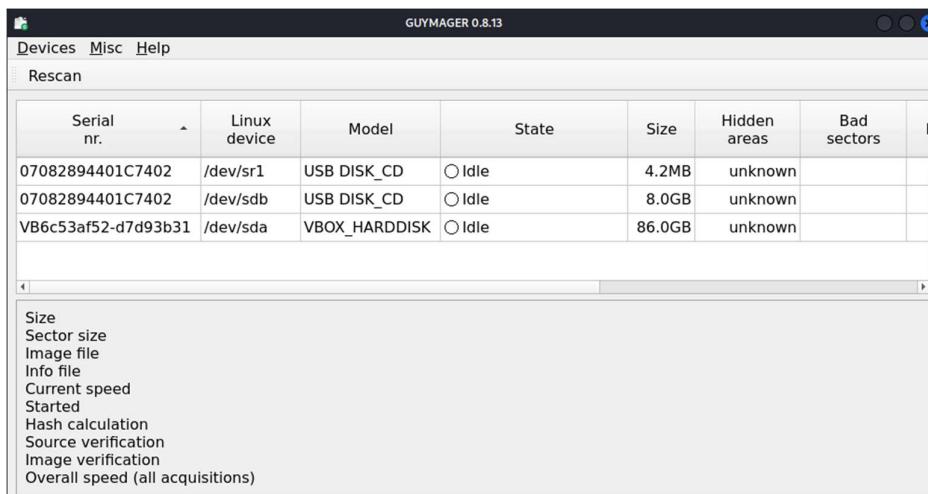
Guymager

To perform the USB drive imaging in Kali Linux, I will use the Guymager tool.

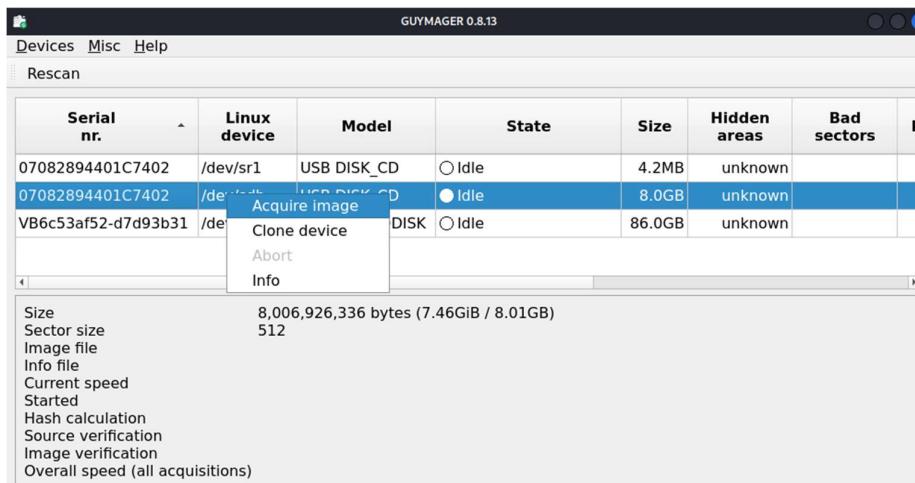
- First, when opening the app, it will ask for authentication since root access is required.



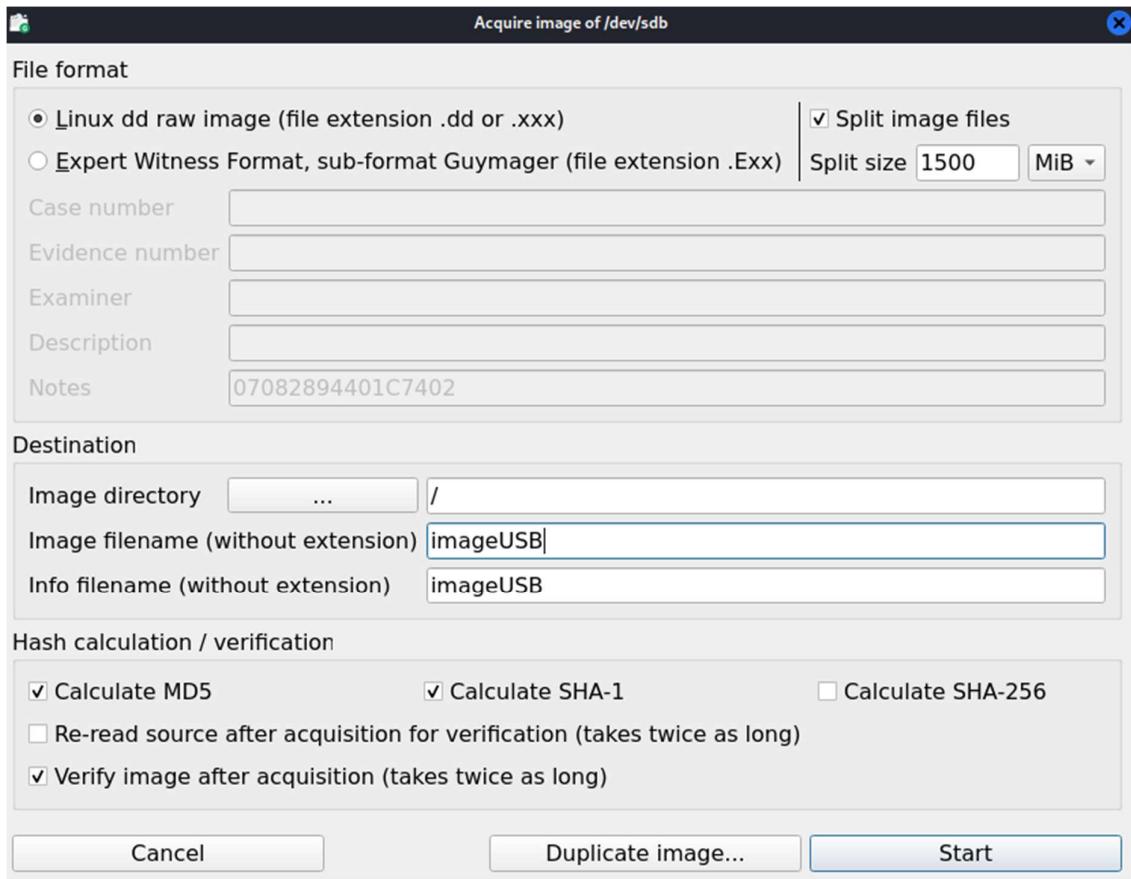
- Once opened, the Guymager app will show this interface.



- I selected my USB drive that is shown in the list of available drives. The image can then be acquired by clicking the 'Acquire image' option.



d. Next, we have to specify the file format and destination. I will save the image files as Raw (dd) type and save the results in a folder. The split size will be 1500 MB as well.

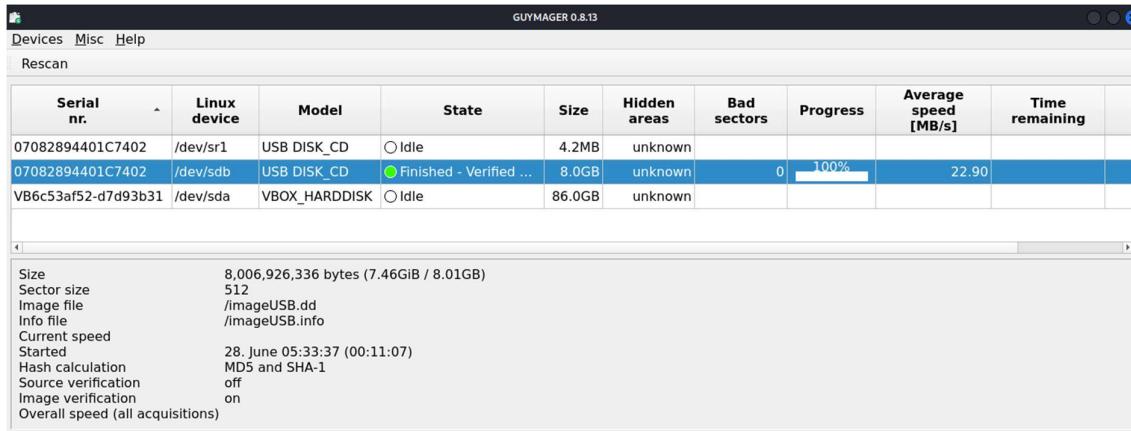


e. Once the process starts, the state is shown as Running and the progress bar is displayed. At the bottom part, additional information like the speed and time elapsed are also provided.

GUYMAGER 0.8.13									
Devices Misc Help									
Rescan									
	Linux device	Model	State	Size	Hidden areas	Bad sectors	Progress	Average speed [MB/s]	Time remaining
02	/dev/sr1	USB DISK_CD	Idle	4.2MB	unknown				
02	/dev/sdb	USB DISK_CD	Running	8.0GB	unknown	0	1%	--	--
3b31	/dev/sda	VBOX_HARDDISK	Idle	86.0GB	unknown				r0 m0 w

Size	8,006,926,336 bytes (7.46GiB / 8.01GB)
Sector size	512
Image file	/imageUSB.dd
Info file	/imageUSB.info
Current speed	12.15 MB/s
Started	28. June 05:33:37 (00:00:15)
Hash calculation	MD5 and SHA-1
Source verification	off
Image verification	on
Overall speed (all acquisitions)	12.15 MB/s

f. Once the imaging is done, the state is changed to ‘Finished – Verified & ok’. A total of 11 minutes and 7 seconds is used for the entire process.



g. We can then check the image files using the terminal as shown below:

```
(kali㉿kali)-[~/Desktop/USB_IMAGE]
└─$ ls -l
total 7819296
-rw-r--r-- 1 kali kali 1572864000 Jun 28 05:35 imageUSB.000
-rw-r--r-- 1 kali kali 1572864000 Jun 28 05:37 imageUSB.001
-rw-r--r-- 1 kali kali 1572864000 Jun 28 05:39 imageUSB.002
-rw-r--r-- 1 kali kali 1572864000 Jun 28 05:41 imageUSB.003
-rw-r--r-- 1 kali kali 1572864000 Jun 28 05:43 imageUSB.004
-rw-r--r-- 1 kali kali 142606336 Jun 28 05:43 imageUSB.005
-rw-r--r-- 1 kali kali      5610 Jun 28 05:44 imageUSB.info
```

Comparison Between FTK Imager and Guymager

FTK Imager and Guymager both produced six files as the imaging results, and each corresponding image file from both tools has the same exact size. Additionally, the images from both tools were verified using the MD5 and SHA-1 hashes.

In Table 2 below, the differences between both tools will be discussed.

Table 2: Differences between FTK Imager and Guymager

FTK Imager	Guymager
Used in local Windows machine.	Used in Kali Linux virtual machine.
Faster, took 7 minutes and 41 seconds to complete the USB drive imaging and verification of image files.	Slower, took 11 minutes and 7 seconds to complete the USB drive imaging and verification of image files.

Offers a comprehensive graphical user interface (GUI) with intuitive options for various forensic imaging tasks.	Also provides a GUI, but its interface is simpler and more straightforward compared to FTK Imager.
Supports a wide range of imaging formats and offers advanced options like compression, verification, and imaging of specific partitions.	Focuses on core imaging tasks with support for various formats and basic imaging settings.
A basic version is available for free, but it requires a license for full forensic use.	Open-source and freely available, comes pre-installed in Kali Linux.

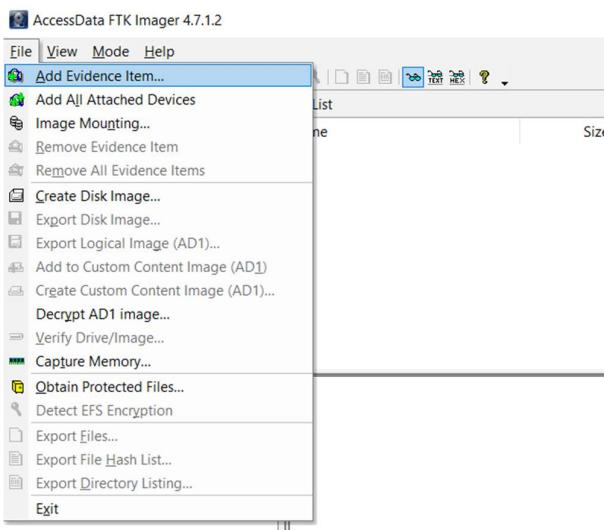
IV. Portfolio 3: Investigation Tools

An image file named ‘DraftComplete.dd’ that contains the forensic image of the Compact Flash (CF) memory card was provided. This file will be used to conduct a forensic investigation to find out the contents of the CF and determine whether Bruce Armiter was involved in unauthorized data transfer or theft of sensitive information from Draft Complete, Inc.

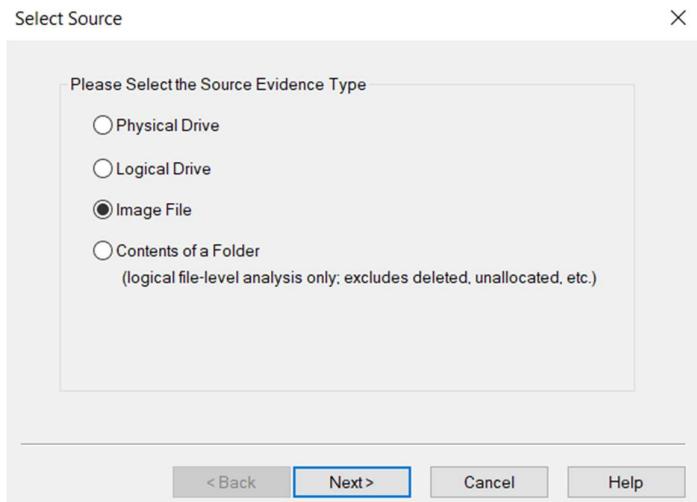
FTK Imager – Windows-based Tool

This investigation can be done using the FTK Imager, which is a powerful forensic tool that can not only be used to perform imaging, but it can also analyze disk images. The steps to conduct the investigation are as follows:

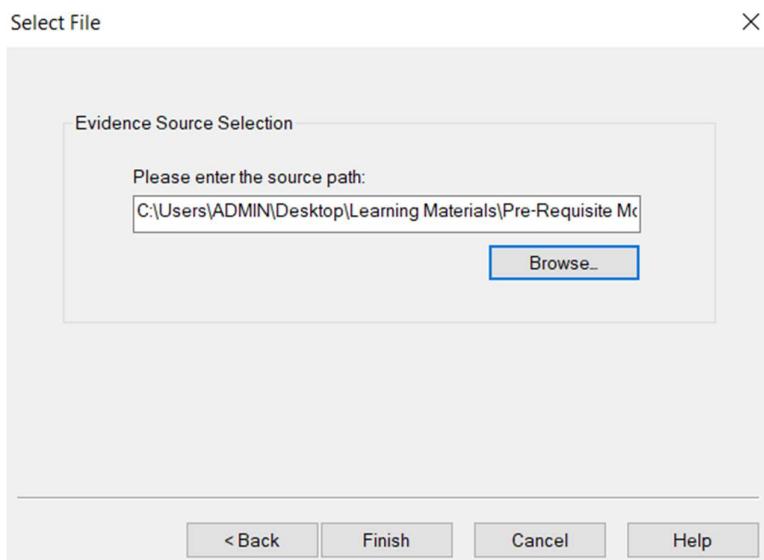
- Add the .dd file as an evidence item in FTK Imager.



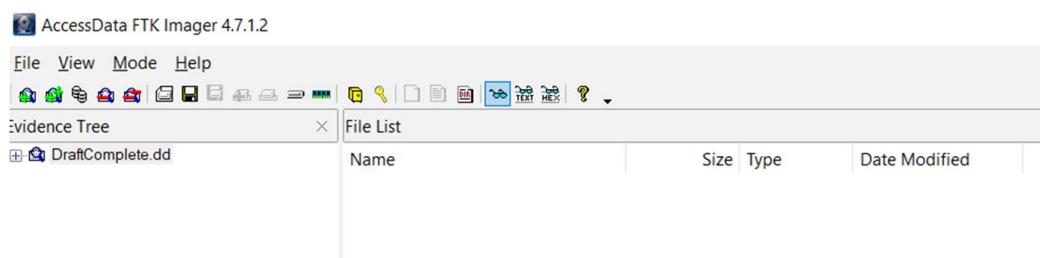
- b. It will then ask for the type of the source file, in this case we will be analyzing an image file.



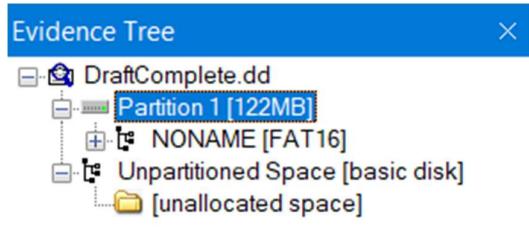
- c. Next, enter the source path of the DraftComplete.dd file.



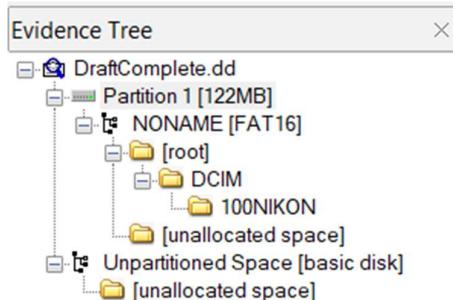
- d. After providing the source path and clicking Finish, the image file is now added to the evidence tree.



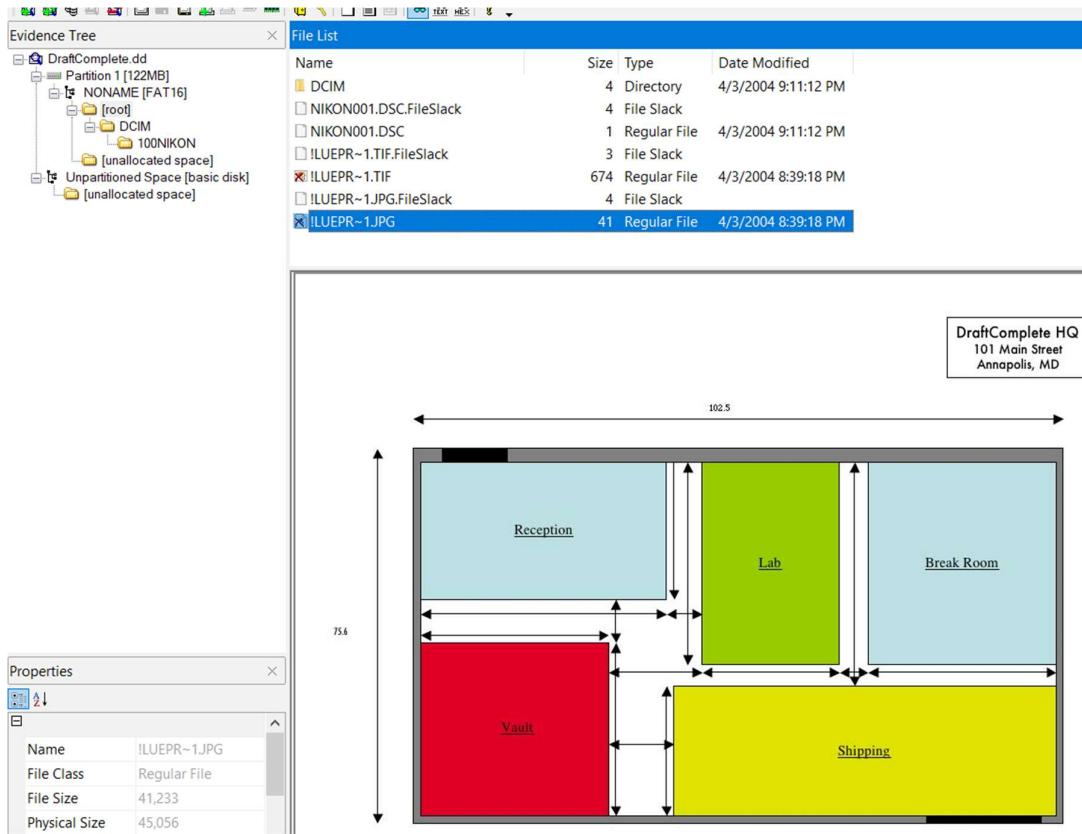
- e. By expanding the image file, we can see that there are two partitions, which are Partition 1 and Unpartitioned Space. According to Lelii (n.d.), a partition is a logical segment of a hard disk that operating systems and file systems treat as an individual unit. This allows the OSes and file systems to handle data on each partition as though it were a separate hard drive.



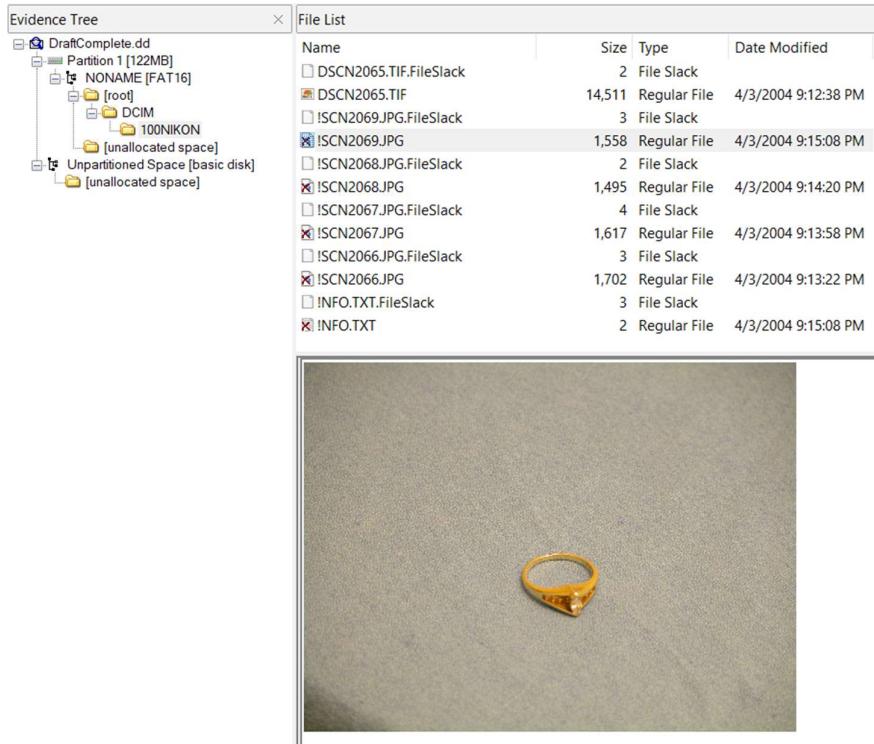
- f. By expanding Partition 1, we can see that there is a root folder with subfolders. We can examine each folder individually to find any information related to pictures of new products and the Draft Complete, Inc. HQ building schematics.



- g. In the root folder, there is a JPG file that shows the company's building schematics. This evidence supports the guard's statement as the file obtained matches the guard's suspicion.



- h. Further inspection of the files revealed images of a few different types of jewelry, saved as JPG files. These images can prove the claims of the guard.



Evidence Tree

- DraftComplete.dd
 - Partition 1 [122MB]
 - NONAME [FAT16]
 - [root]
 - DCIM
 - 100NIKON
 - [unallocated space]
 - Unpartitioned Space [basic disk]
 - [unallocated space]

File List

Name	Size	Type	Date Modified
DSCN2065.TIF.FileSlack	2	File Slack	
DSCN2065.TIF	14,511	Regular File	4/3/2004 9:12:38 PM
!SCN2069.JPG.FileSlack	3	File Slack	
!SCN2069.JPG	1,558	Regular File	4/3/2004 9:15:08 PM
!SCN2068.JPG.FileSlack	2	File Slack	
!SCN2068.JPG	1,495	Regular File	4/3/2004 9:14:20 PM
!SCN2067.JPG.FileSlack	4	File Slack	
!SCN2067.JPG	1,617	Regular File	4/3/2004 9:13:58 PM
!SCN2066.JPG.FileSlack	3	File Slack	
!SCN2066.JPG	1,702	Regular File	4/3/2004 9:13:22 PM
INFO.TXT.FileSlack	3	File Slack	
INFO.TXT	2	Regular File	4/3/2004 9:15:08 PM



Evidence Tree

- DraftComplete.dd
 - Partition 1 [122MB]
 - NONAME [FAT16]
 - [root]
 - DCIM
 - 100NIKON
 - [unallocated space]
 - Unpartitioned Space [basic disk]
 - [unallocated space]

File List

Name	Size	Type	Date Modified
DSCN2065.TIF.FileSlack	2	File Slack	
DSCN2065.TIF	14,511	Regular File	4/3/2004 9:12:38 PM
!SCN2069.JPG.FileSlack	3	File Slack	
!SCN2069.JPG	1,558	Regular File	4/3/2004 9:15:08 PM
!SCN2068.JPG.FileSlack	2	File Slack	
!SCN2068.JPG	1,495	Regular File	4/3/2004 9:14:20 PM
!SCN2067.JPG.FileSlack	4	File Slack	
!SCN2067.JPG	1,617	Regular File	4/3/2004 9:13:58 PM
!SCN2066.JPG.FileSlack	3	File Slack	
!SCN2066.JPG	1,702	Regular File	4/3/2004 9:13:22 PM
INFO.TXT.FileSlack	3	File Slack	
INFO.TXT	2	Regular File	4/3/2004 9:15:08 PM



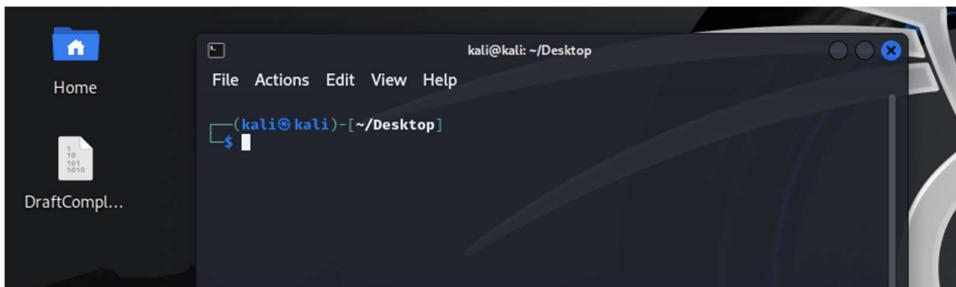
Foremost is a forensic program to recover lost files based on their headers, footers, and internal data structures. Foremost can work on image files, such as those generated by dd.

Foremost – Linux-based Tool

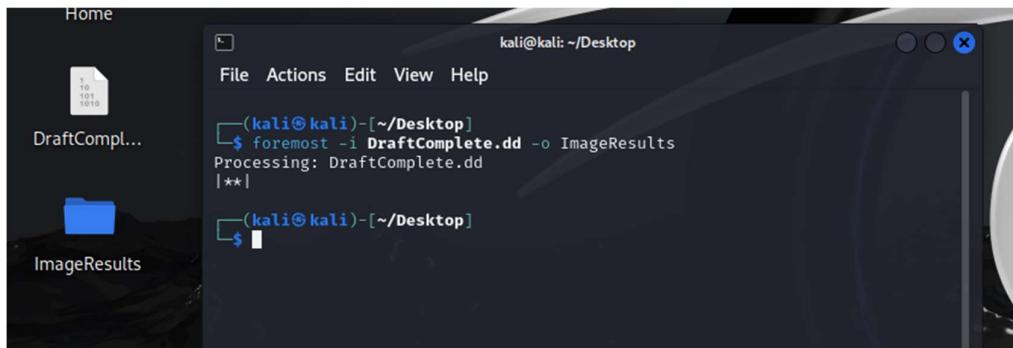
Similarly, the same investigation can be done using a Linux-based tool. We will be using Foremost in Kali Linux to analyze the image file. According to Kali (n.d.), Foremost is a forensic application designed to recover lost files by identifying their headers, footers, and internal data structures. It can operate on image files, such as those created by dd.

The steps to operate on the image file are:

- a. Open the terminal and change directory to where the DraftComplete.dd is saved, in this case the file was saved in /Desktop.



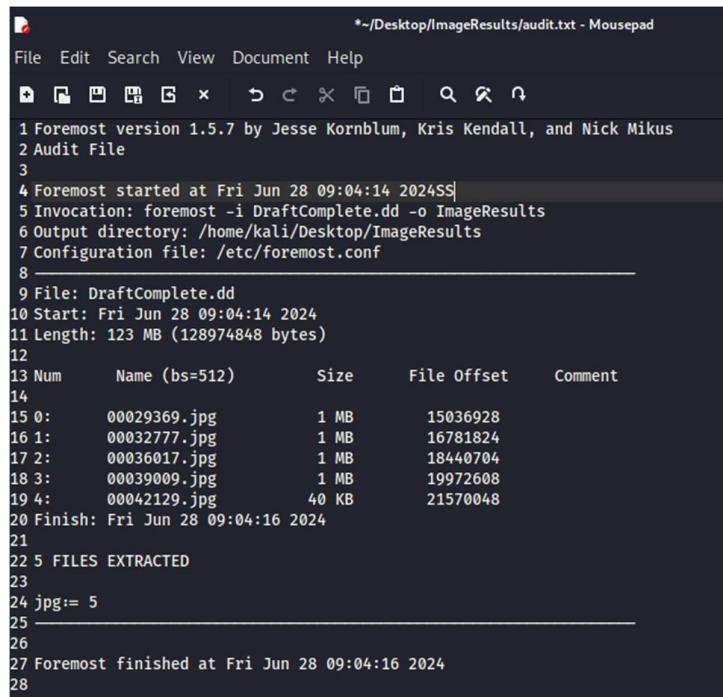
- b. Next, we can use commands the foremost -i DraftComplete.dd -o ImageResults command to analyze the image file and store the results in a directory called ‘ImageResults’.



The screenshot shows a Kali Linux desktop environment. On the left, there's a dock with icons for Home, DraftComple..., and ImageResults. A terminal window is open in the center, showing the command:

```
(kali㉿kali)-[~/Desktop]
$ foremost -i DraftComplete.dd -o ImageResults
Processing: DraftComplete.dd
|**|
(kali㉿kali)-[~/Desktop]
$
```

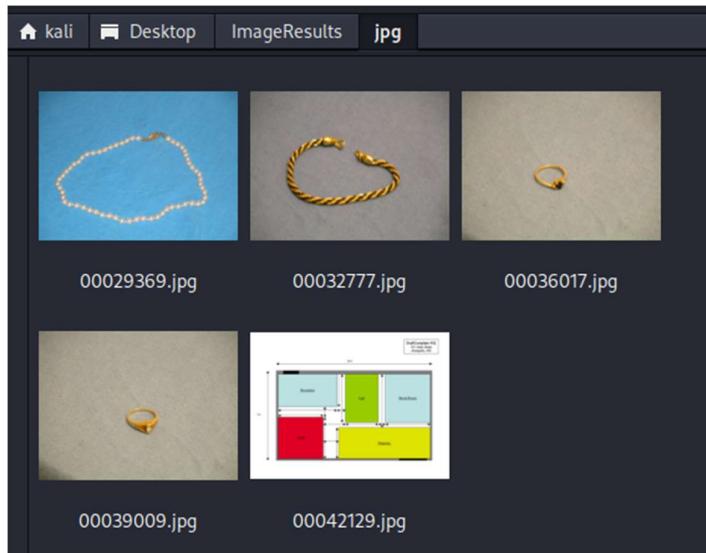
- c. In the ImageResults folder, there is a subfolder and a .txt file which is an audit file that contains information about the findings. The content of the audit file is:



The screenshot shows a Mousepad application window with the file `*~/Desktop/ImageResults/audit.txt`. The content of the audit.txt file is as follows:

```
1 Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
2 Audit File
3
4 Foremost started at Fri Jun 28 09:04:14 2024$|
5 Invocation: foremost -i DraftComplete.dd -o ImageResults
6 Output directory: /home/kali/Desktop/ImageResults
7 Configuration file: /etc/foremost.conf
8
9 File: DraftComplete.dd
10 Start: Fri Jun 28 09:04:14 2024
11 Length: 123 MB (128974848 bytes)
12
13 Num      Name (bs=512)      Size      File Offset      Comment
14
15 0:    00029369.jpg        1 MB      15036928
16 1:    00032777.jpg        1 MB      16781824
17 2:    00036017.jpg        1 MB      18440704
18 3:    00039009.jpg        1 MB      19972608
19 4:    00042129.jpg        40 KB     21570048
20 Finish: Fri Jun 28 09:04:16 2024
21
22 5 FILES EXTRACTED
23
24 jpg:= 5
25
26
27 Foremost finished at Fri Jun 28 09:04:16 2024
28
```

- d. From the audit file, we can see that there are a total of 5 JPG files extracted from the image file. When we open the jpg subfolder, the 5 JPG files that were extracted are saved here.



- e. These 5 JPG files are the same images that we extracted using FTK Imager, which supports the guard's statement.

Summary of Case Findings

During the forensic investigation of the disk image file DraftComplete.dd using FTK Imager, several key discoveries were made that support the security guard's suspicions. The disk image was found to contain two partitions, which are Partition 1 and Unpartitioned Space. Partition 1 contains the main file system, while the unpartitioned space remains unused and does not contain any files. By expanding Partition 1, we accessed the root folder, which contains subfolders and several files relevant to the investigation.

Detailed examination of the root folder revealed various files, including a JPG file that depicts the schematics of Draft Complete, Inc.'s headquarters. This file directly supports the security guard's suspicion that sensitive information about the building layout was being smuggled out. Additionally, several JPG files containing images of different types of high-end jewelry were discovered in the same directory. These files indicate potential unauthorized access and illegal disclosure of product designs.

Both FTK Imager and Foremost were used to extract data from the DraftComplete.dd disk image. While FTK Imager extracted the entire file system, providing a complete view of all directories and files, Foremost focused on recovering files based on their headers, footers, and internal data structures. In this case, both tools successfully extracted the same JPG files containing the company's building schematics and jewelry images. However, FTK Imager

extracted data from the whole file system, while Foremost only retrieved the JPG files, demonstrating the different capabilities and focuses of these forensic tools.

The forensic analysis of the DraftComplete.dd disk image provides substantial evidence to support the security guard's concerns. The presence of the company's building schematics and images of new jewelry products on the Compact Flash (CF) memory card suggests that Bruce Armiter was indeed involved in the unauthorized transfer of sensitive information from the company. This evidence can be used to support further investigation and potential legal actions against the suspect for data theft and intellectual property violations.

V. Conclusion

In conclusion, this portfolio has provided a practical exploration of essential tools and methodologies in digital forensics and cybersecurity. From Hex Editor and Wireshark for analysing files and network traffic to Exif Tool for extracting metadata, each tool serves a critical role in uncovering insights and securing digital environments. The demonstrations using FTK Imager and Guymager in Kali Linux underscore the importance of proper imaging techniques in preserving digital evidence and conducting thorough investigations. These tools not only facilitate technical proficiency but also highlight the interdisciplinary nature of cybersecurity, where proactive measures and effective response strategies are essential.

The investigation into the Draft Complete, Inc. case further illustrates the practical application of these forensic tools. By utilizing FTK Imager and Foremost, we successfully identified and extracted crucial evidence, such as the company's building schematics and images of high-end jewelry products. This evidence supports the suspicion that Bruce Armiter was involved in the unauthorized transfer of sensitive information from Draft Complete, Inc. The case underscores the value of thorough forensic analysis in uncovering illicit activities and providing a basis for further legal action. Overall, this portfolio demonstrates the significance of comprehensive forensic methodologies in both routine cybersecurity tasks and complex investigations.

VI. References

Kali (n.d.). foremost | Kali Linux tools. <https://www.kali.org/tools/foremost/>

Lelli, S. (n.d.). What is partition? | Definition from TechTarget. TechTarget.

<https://www.techtarget.com/searchstorage/definition/partition>

VII. Appendix

None