

Metadata-Extraction-using-ExifTool-log2timeline-and-Hidden-Data-Search-using-Steganography-Tools

AIM:

To extract metadata, perform timeline analysis, and search for hidden data using forensic tools like ExifTool, log2timeline, and steganography detection tools.

DESIGN STEPS:

Step 1:

Use exiftool to extract metadata from files such as images, documents, and videos.

Step 2:

Use log2timeline and plaso to create and analyze event timelines from system logs and file metadata.

Step 3:

Apply steganography detection tools like steghide, zsteg, or binwalk to uncover hidden data in media files.

PROGRAM:

Metadata and Timeline Forensics, Steganography Analysis Steps

OUTPUT:

✓ A. Using ExifTool – for file metadata

- 📦 Install:

```
sudo apt update  
sudo apt install exiftool -y
```



- 📁 Extract metadata from a file:

EXAMPLE

```
exiftool image.jpg
```



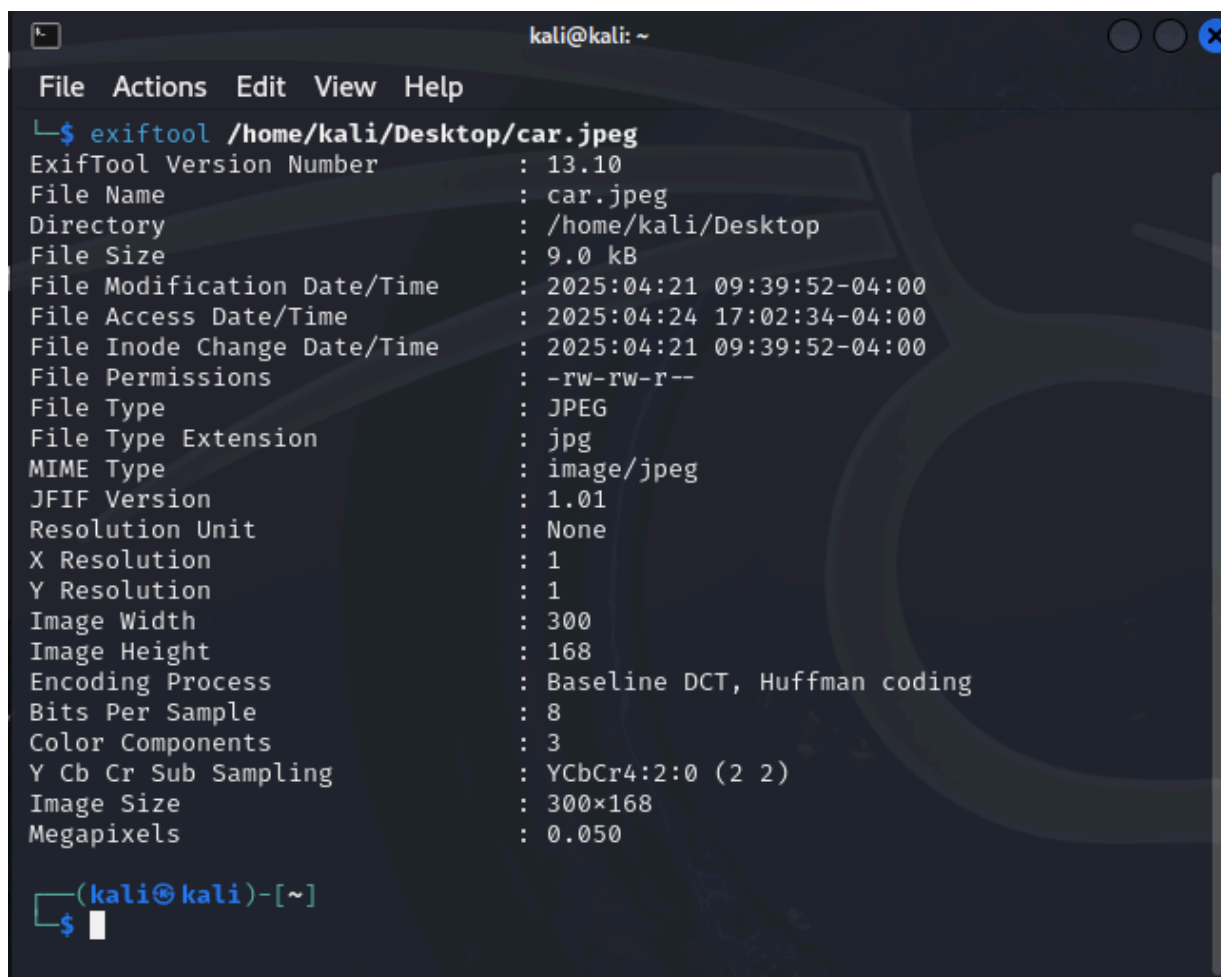
-  **Batch process a folder:**

```
exiftool -r /path/to/folder
```



-  **Useful flags:**

- `-G`: Show metadata group
- `-time:all`: Show only timestamps
- `-GPSLatitude -GPSLongitude`: Extract GPS data



```
kali@kali: ~  
File Actions Edit View Help  
$ exiftool /home/kali/Desktop/car.jpeg  
ExifTool Version Number      : 13.10  
File Name                    : car.jpeg  
Directory                    : /home/kali/Desktop  
File Size                    : 9.0 kB  
File Modification Date/Time   : 2025:04:21 09:39:52-04:00  
File Access Date/Time        : 2025:04:24 17:02:34-04:00  
File Inode Change Date/Time   : 2025:04:21 09:39:52-04:00  
File Permissions              : -rw-rw-r--  
File Type                    : JPEG  
File Type Extension          : jpg  
MIME Type                     : image/jpeg  
JFIF Version                  : 1.01  
Resolution Unit               : None  
X Resolution                  : 1  
Y Resolution                  : 1  
Image Width                   : 300  
Image Height                  : 168  
Encoding Process              : Baseline DCT, Huffman coding  
Bits Per Sample               : 8  
Color Components              : 3  
Y Cb Cr Sub Sampling          : YCbCr4:2:0 (2 2)  
Image Size                    : 300x168  
Megapixels                    : 0.050  
  
(kali@kali)-[~]  
$
```

install log2timeline

```
sudo apt install plaso -y
```



```
sudo apt install steghide -y
```



- **Embed data**

```
steghide embed -cf /home/kali/Desktop/car.jpeg -ef /home/kali/Desktop/message.txt
```



```
(kali㉿kali)-[~]
$ steghide embed -cf /home/kali/Desktop/car.jpeg -ef /home/kali/Desktop/message.txt
Enter passphrase:
Re-Enter passphrase:
embedding "/home/kali/Desktop/message.txt" in "/home/kali/Desktop/car.jpeg" ... done

(kali㉿kali)-[~]
$
```

- Extract hidden data:

Example: `$ steghide extract -sf hidden.jpg`

```
steghide extract -sf /home/kali/Desktop/car.jpeg
```



```
(kali㉿kali)-[~]
$ steghide extract -sf /home/kali/Desktop/car.jpeg

Enter passphrase:
wrote extracted data to "message.txt".

(kali㉿kali)-[~]
$
```

Using binwalk – for file analysis

```
sudo apt install binwalk -y
binwalk suspicious.jpg
```



```
binwalk /home/kali/Desktop/car.jpeg
```



```
(kali㉿kali)-[~]
$ binwalk /home/kali/Desktop/car.jpeg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01

```
(kali㉿kali)-[~]
$
```

RESULT:

Metadata was successfully extracted, timeline analysis was completed, and hidden data was identified using steganography tools.