Steganography

Pranav Mohanraj

Steganography Image Download Link -

https://drive.google.com/file/d/1nsZ97v4deyqJe4L367DJxLb0QL3MQrj5/view?usp=drive_link

Table of Contents

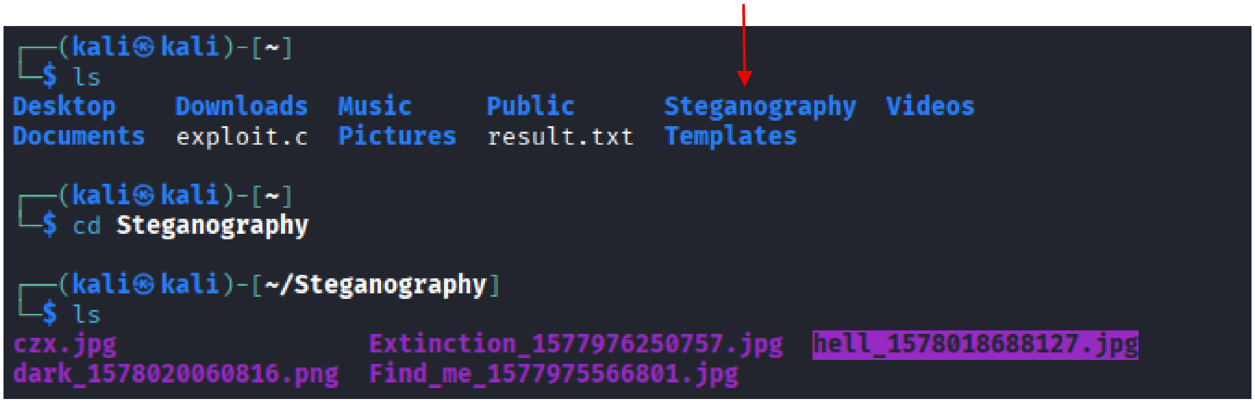
Introduction

Steganography is the technique of concealing sensitive information within ordinary digital media such as images, audio, video, or text in a manner that prevents detection of the hidden data. Unlike cryptography, which focuses on securing the content of a message, steganography aims to hide the very existence of communication. This makes it a valuable concept in the field of cybersecurity, particularly in areas such as secure communication, digital forensics, and information security analysis.

In this documentation, a practical demonstration of image-based steganography is carried out using a Kali Linux system with the IP address **192.168.28.130** as the attacker environment. The steganography process is performed on the image file **hell_1578018688127.jpg**, which is used to embed and analyze hidden data using standard steganography techniques and tools. This exercise is intended to provide a clear understanding of how data can be concealed within digital images, as well as to highlight the role of steganography in both offensive and defensive cybersecurity scenarios.

2. Steganography File Preparation

Prior to commencing the steganography analysis, a dedicated working directory named **Steganography** was manually created within the Kali Linux environment. This directory was used to store and manage image files intended for steganographic examination, ensuring a structured and controlled workflow throughout the documentation.

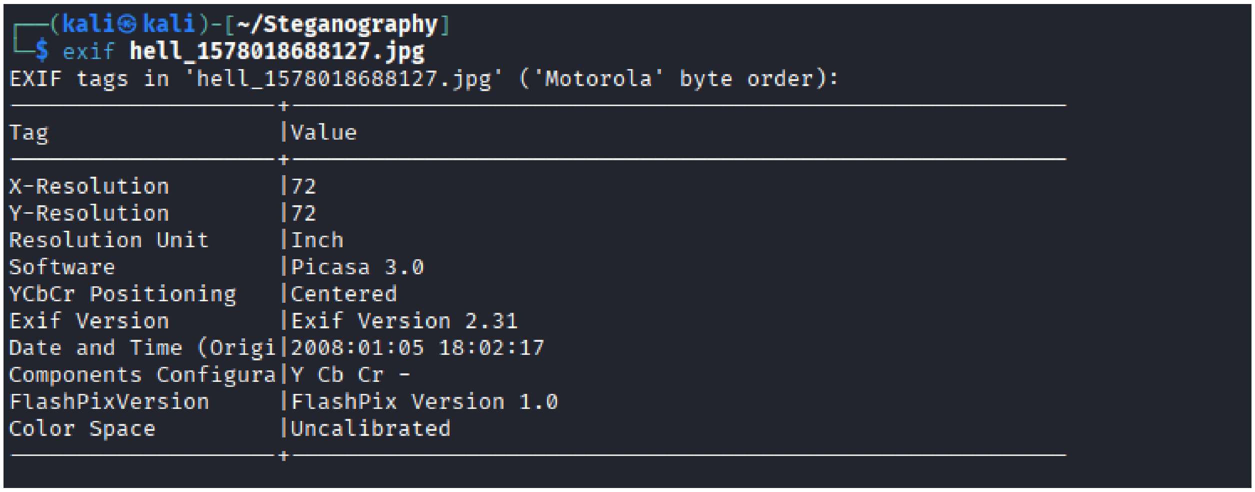


```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Downloads  Music     Public      Steganography  Videos
Documents  exploit.c  Pictures  result.txt  Templates

┌──(kali㉿kali)-[~]
└─$ cd Steganography

┌──(kali㉿kali)-[~/Steganography]
└─$ ls
czx.jpg                 Extinction_1577976250757.jpg  hell_1578018688127.jpg
dark_1578020060816.png  Find_me_1577975566801.jpg
```

3. Metadata Analysis of the Target Image Using EXIF

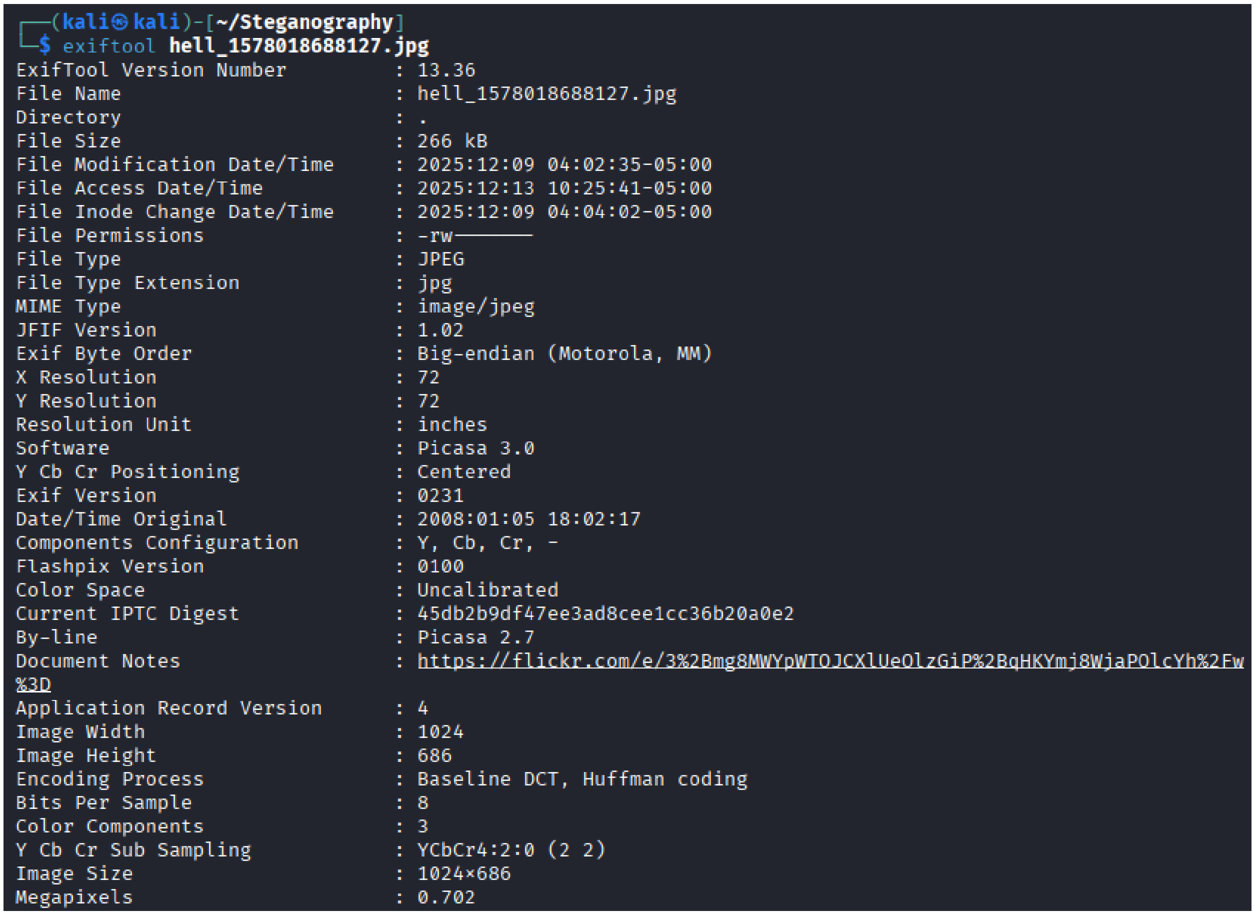


```
┌──(kali㉿kali)-[~/Steganography]
└─$ exif hell_1578018688127.jpg
EXIF tags in 'hell_1578018688127.jpg' ('Motorola' byte order):
--------------------+---------------------------------------------------------
Tag                 |Value
--------------------+---------------------------------------------------------
X-Resolution        |72
Y-Resolution        |72
Resolution Unit     |Inch
Software            |Picasa 3.0
YCbCr Positioning   |Centered
Exif Version        |Exif Version 2.31
Date and Time (Origi|2008:01:05 18:02:17
Components Configura|Y Cb Cr -
FlashPixVersion     |FlashPix Version 1.0
Color Space         |Uncalibrated
--------------------+---------------------------------------------------------
```

The screenshot shows the use of the exif tool to extract metadata from the image **hell_1578018688127.jpg**. The output provides basic information about the image, including resolution, color space, EXIF version, and the software used to process the file. The metadata indicates that the image was edited using **Picasa 3.0**, suggesting that the file has undergone post-processing. No anomalies or visible indicators of embedded data are observed at this stage;

however, metadata analysis is an essential initial step in steganalysis, as it helps identify image modifications that may support further investigation.

4. Detailed Metadata Inspection Using ExifTool

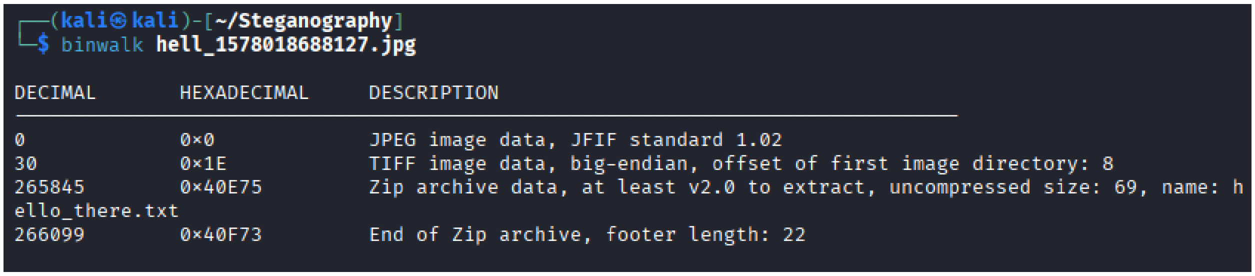


```
┌──(kali㉿kali)-[~/Steganography]
└─$ exiftool hell_1578018688127.jpg
ExifTool Version Number         : 13.36
File Name                       : hell_1578018688127.jpg
Directory                       : .
File Size                       : 266 kB
File Modification Date/Time     : 2025:12:09 04:02:35-05:00
File Access Date/Time           : 2025:12:13 10:25:41-05:00
File Inode Change Date/Time     : 2025:12:09 04:04:02-05:00
File Permissions                : -rw-------
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : Picasa 3.0
Y Cb Cr Positioning             : Centered
Exif Version                    : 0231
Date/Time Original              : 2008:01:05 18:02:17
Components Configuration        : Y, Cb, Cr, -
Flashpix Version                : 0100
Color Space                     : Uncalibrated
Current IPTC Digest             : 45db2b9df47ee3ad8cee1cc36b20a0e2
By-line                         : Picasa 2.7
Document Notes                  : https://flickr.com/e/3%2Bmg8MWYpWTOJCXlUeOlzGiP%2BqHKYmj8WjaPOlcYh%2Fw
%3D
Application Record Version      : 4
Image Width                     : 1024
Image Height                    : 686
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                      : 1024x686
Megapixels                      : 0.702
```

The screenshot shows the use of the exiftool utility to perform an extended metadata analysis on the image **hell_1578018688127.jpg**. The output provides detailed file and image metadata, including file properties, encoding information, and software details. Most notably, the presence of a **Current IPTC Digest** value (45db2b9df47ee3ad8cee1cc36b20a0e2) confirms that **IPTC metadata exists and has been written to the image**. IPTC metadata is not generated automatically by image capture devices and is typically added manually or through software, indicating intentional metadata modification. This finding is significant in steganalysis, as IPTC fields can be used to store hidden or encoded information without altering the visible content of the image. The existence of this digest therefore represents **direct metadata-level evidence of**

potential concealed data, justifying further investigation using file structure analysis and extraction techniques.

5. File Structure Analysis Using Binwalk

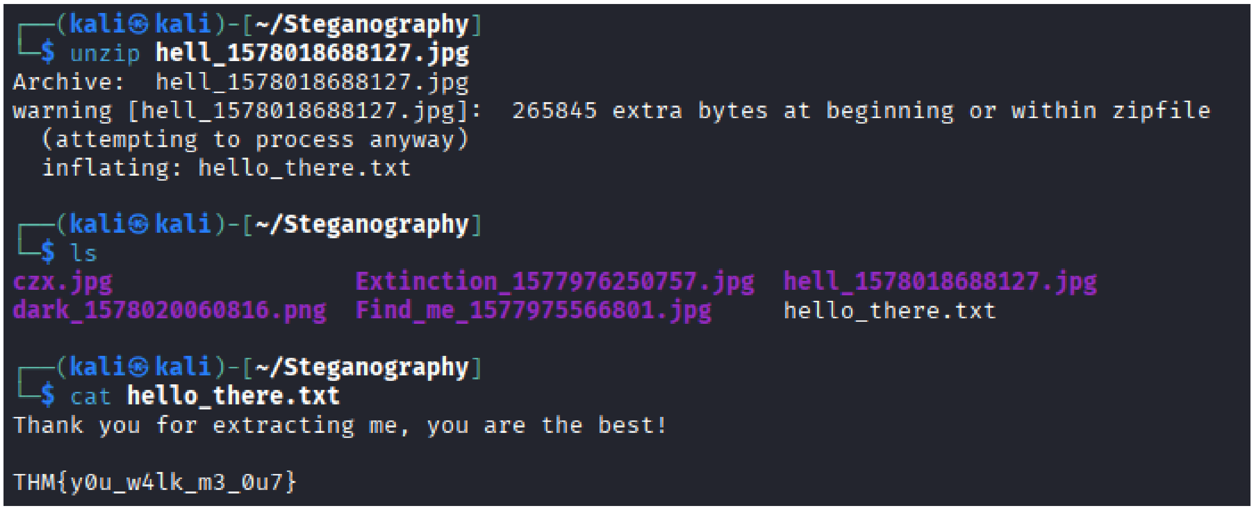


```
┌──(kali㉿kali)-[~/Steganography]
└─$ binwalk hell_1578018688127.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             JPEG image data, JFIF standard 1.02
30            0×1E            TIFF image data, big-endian, offset of first image directory: 8
265845        0×40E75         Zip archive data, at least v2.0 to extract, uncompressed size: 69, name: h
ello_there.txt
266099        0×40F73         End of Zip archive, footer length: 22
```

The screenshot shows the use of the binwalk tool to analyze the internal file structure of the image **hell_1578018688127.jpg**. The output confirms that, in addition to standard JPEG image data, the file contains **embedded ZIP archive data**. Binwalk identifies a ZIP archive beginning at offset 265845, with the embedded file named **hello_there.txt**, followed by a valid ZIP archive footer. This finding provides **direct evidence of hidden data embedded within the image**, confirming the use of steganographic techniques beyond metadata manipulation. The presence of a valid compressed archive inside the image justifies proceeding to data extraction in order to recover the concealed content.

6. Extraction of Embedded Data Using Unzip



```
┌──(kali㉿kali)-[~/Steganography]
└─$ unzip hell_1578018688127.jpg
Archive:  hell_1578018688127.jpg
warning [hell_1578018688127.jpg]:  265845 extra bytes at beginning or within zipfile
  (attempting to process anyway)
  inflating: hello_there.txt

┌──(kali㉿kali)-[~/Steganography]
└─$ ls
czx.jpg                  Extinction_1577976250757.jpg  hell_1578018688127.jpg
dark_1578020060816.png   Find_me_1577975566801.jpg     hello_there.txt

┌──(kali㉿kali)-[~/Steganography]
└─$ cat hello_there.txt
Thank you for extracting me, you are the best!

THM{y0u_w4lk_m3_0u7}
```

The screenshot shows the extraction of hidden data from the image **hell_1578018688127.jpg** using the unzip utility. Based on prior Binwalk analysis, the image was confirmed to contain embedded ZIP archive data. During extraction, a warning message indicating the presence of extra bytes was displayed, which is consistent with ZIP data appended to a JPEG file and does not indicate extraction failure. The extraction was completed successfully, resulting in the recovery of a file named **hello_there.txt**.

Upon examining the extracted file, the contents revealed a hidden message and flag. The output displayed the message *“Thank you for extracting me, you are the best!”* followed by the flag **THM{y0u_w4lk_m3_0u7}**, confirming that concealed data had been successfully embedded within the image and accurately extracted. This result validates the steganalysis process and demonstrates effective identification and recovery of hidden information from the target image.