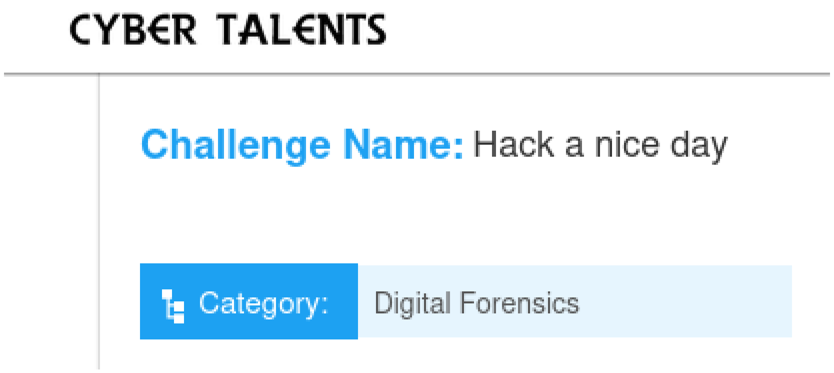
LinkedIn: Kelvin Kimotho


CYBER TALENTS

Challenge Name: Hack a nice day

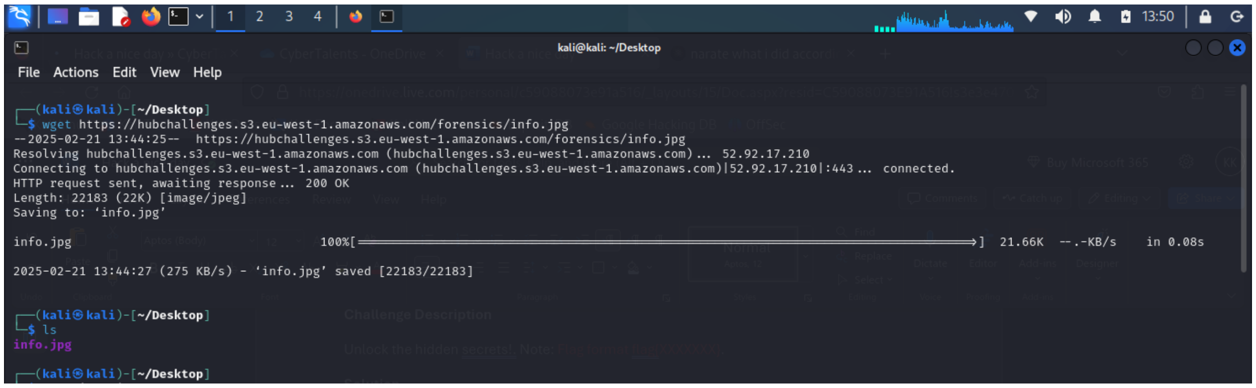
Category: Digital Forensics

Challenge Description

Unlock the hidden secrets!. Note: Flag format flag{XXXXXXX}.

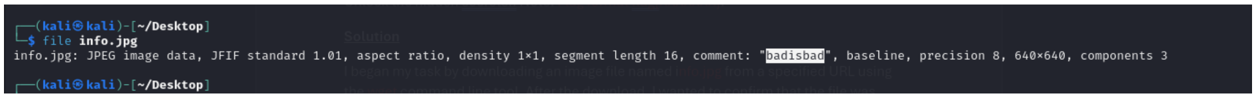
Solution

I began my task by downloading an image file named info.jpg from a specified URL using the wget command line tool. After the download, I wanted to confirm that the file was present on my Desktop, so I used the ls command to list the files in the directory. The output confirmed that info.jpg was indeed there.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ wget https://hubchallenges.s3.eu-west-1.amazonaws.com/forensics/info.jpg
--2025-02-21 13:44:25--  https://hubchallenges.s3.eu-west-1.amazonaws.com/forensics/info.jpg
Resolving hubchallenges.s3.eu-west-1.amazonaws.com (hubchallenges.s3.eu-west-1.amazonaws.com)... 52.92.17.210
Connecting to hubchallenges.s3.eu-west-1.amazonaws.com (hubchallenges.s3.eu-west-1.amazonaws.com)|52.92.17.210|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22183 (22K) [image/jpeg]
Saving to: ‘info.jpg’

info.jpg            100%[===================================================>]  21.66K  --.-KB/s    in 0.08s

2025-02-21 13:44:27 (275 KB/s) - ‘info.jpg’ saved [22183/22183]

┌──(kali㉿kali)-[~/Desktop]
└─$ ls
info.jpg

┌──(kali㉿kali)-[~/Desktop]
```

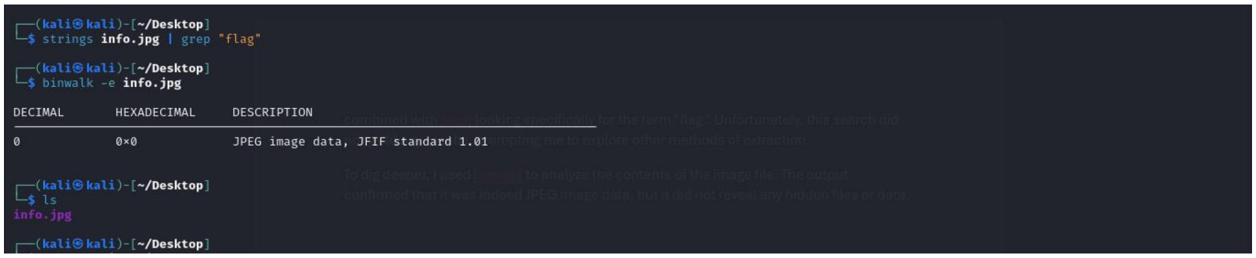
Next, I checked the file type of info.jpg using the file command. The output revealed that it was a JPEG image, providing additional details such as its resolution, aspect ratio, and an interesting comment: "badisbad." This comment caught my attention as it could potentially be relevant for further analysis.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ file info.jpg
info.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1×1, segment length 16, comment: "badisbad", baseline, precision 8, 640×640, components 3

┌──(kali㉿kali)-[~/Desktop]
```

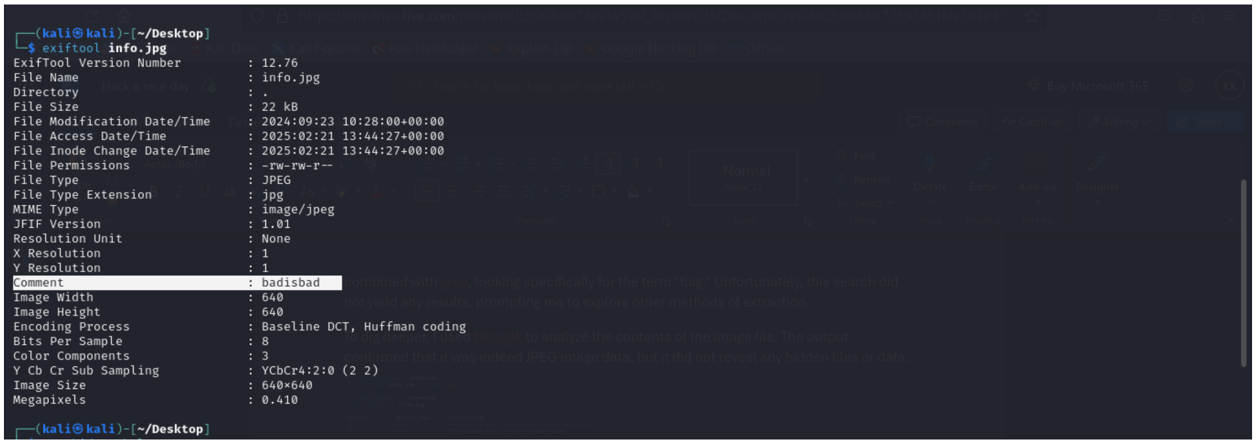
I then attempted to search for any flags within the image using the strings command combined with grep, looking specifically for the term "flag." Unfortunately, this search did not yield any results, prompting me to explore other methods of extraction.

To dig deeper, I used binwalk to analyze the contents of the image file. The output confirmed that it was indeed JPEG image data, but it did not reveal any hidden files or data.



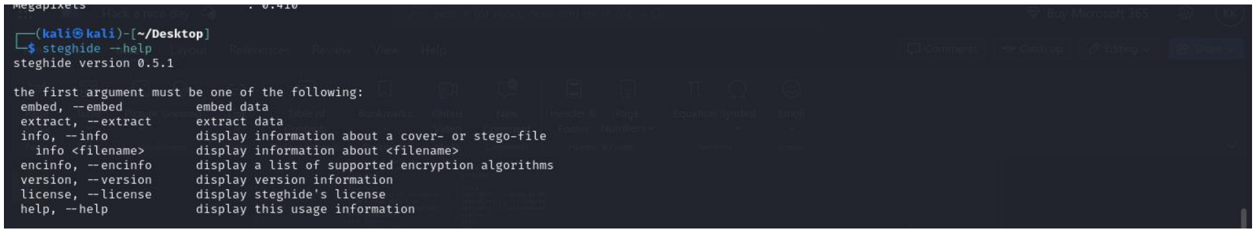
```
┌──(kali㉿kali)-[~/Desktop]
└─$ strings info.jpg | grep "flag"

┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk -e info.jpg

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


┌──(kali㉿kali)-[~/Desktop]
└─$ ls
info.jpg

┌──(kali㉿kali)-[~/Desktop]
```

Next, I decided to extract metadata from the image using exiftool. The output provided various details about the file, including its size, modification date, and the previously noted comment "badisbad." This comment seemed significant.

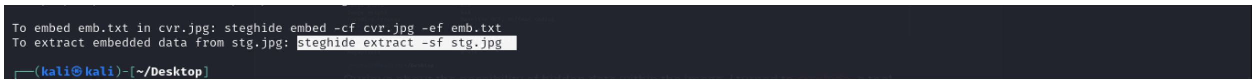


```
┌──(kali㉿kali)-[~/Desktop]
└─$ exiftool info.jpg
ExifTool Version Number         : 12.76
File Name                       : info.jpg
Directory                       : .
File Size                       : 22 kB
File Modification Date/Time     : 2024:09:23 10:28:00+00:00
File Access Date/Time           : 2025:02:21 13:44:27+00:00
File Inode Change Date/Time     : 2025:02:21 13:44:27+00:00
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
Comment                         : badisbad
Image Width                     : 640
Image Height                    : 640
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                      : 640x640
Megapixels                      : 0.410

┌──(kali㉿kali)-[~/Desktop]
```

Curious about the possibility of hidden data within the image, I turned to steghide, a tool designed for embedding and extracting data from image files. I first checked the help documentation for steghide to understand the commands available for extraction.

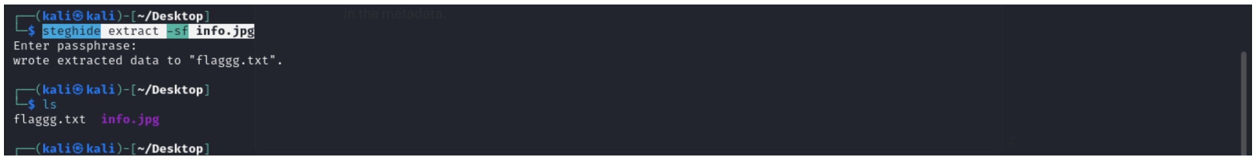


```
Megapixels                      : 0.410

┌──(kali㉿kali)-[~/Desktop]
└─$ steghide --help
steghide version 0.5.1

the first argument must be one of the following:
 embed, --embed          embed data
 extract, --extract      extract data
 info, --info            display information about a cover- or stego-file
   info <filename>       display information about <filename>
 encinfo, --encinfo      display a list of supported encryption algorithms
 version, --version      display version information
 license, --license      display steghide's license
 help, --help            display this usage information
```



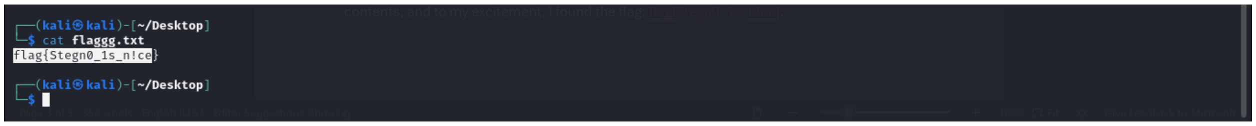
```
To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg

┌──(kali㉿kali)-[~/Desktop]
```

I then proceeded to extract any hidden data from info.jpg using steghide. I ran the command " steghide extract -sf info.jpg " When prompted, I entered the passphrase "badisbad," which I had discovered in the metadata.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ steghide extract -sf info.jpg
Enter passphrase:
wrote extracted data to "flaggg.txt".

┌──(kali㉿kali)-[~/Desktop]
└─$ ls
flaggg.txt  info.jpg

┌──(kali㉿kali)-[~/Desktop]
```

The extraction was successful, and I received confirmation that the data had been written to a file named flaggg.txt. After the extraction, I listed the files in the directory again and confirmed that flaggg.txt was now present alongside info.jpg. I opened flaggg.txt to view its contents, and to my excitement, I found the flag: flag{Stegn0_1s_n!ce}.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat flaggg.txt
flag{Stegn0_1s_n!ce}

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```