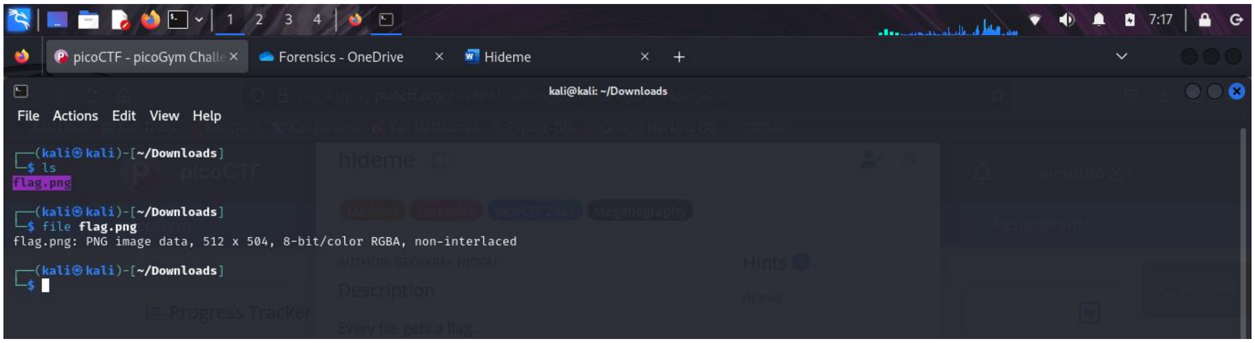
LinkedIn: Kelvin kimotho

Description

Every file gets a flag. The SOC analyst saw one image been sent back and forth between two people. They decided to investigate and found out that there was more than what meets the eye here.

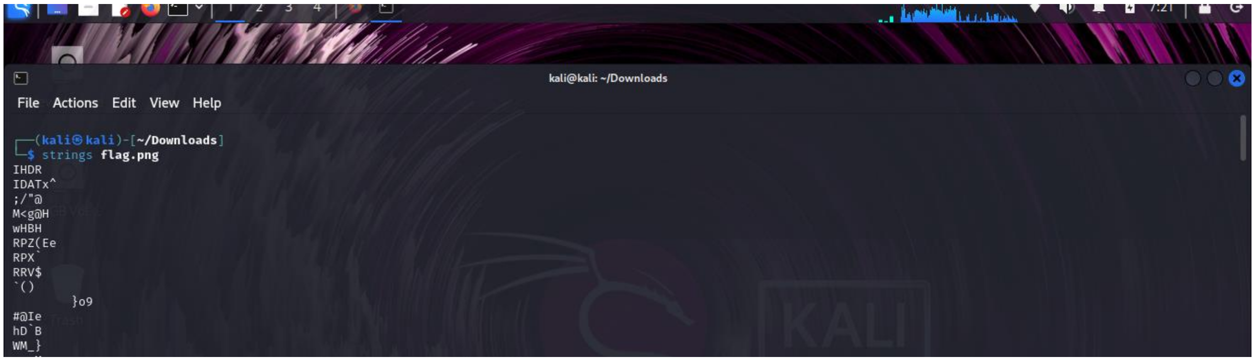
Solution

I began by downloading the image file for analysis.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ ls
flag.png

┌──(kali㉿kali)-[~/Downloads]
└─$ file flag.png
flag.png: PNG image data, 512 x 504, 8-bit/color RGBA, non-interlaced

┌──(kali㉿kali)-[~/Downloads]
└─$ 
```

I then used the strings command on the flag.png file to extract any embedded text or hidden data.

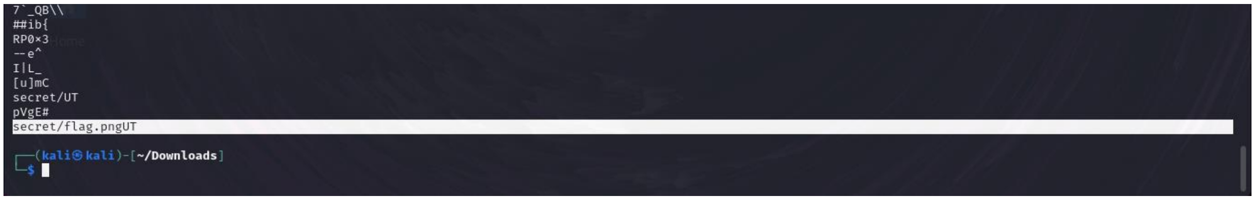


```
┌──(kali㉿kali)-[~/Downloads]
└─$ strings flag.png
IHDR
IDATx^
;/"@
M<g@H
wHBH
RPZ(Ee
RPX`
RRV$
`()
        }o9
#@Ie
hD`B
WM_}
```

After running the strings command on the flag.png file, I carefully analyzed the output for anything unusual or suspicious. Among the extracted text, I noticed the following string:

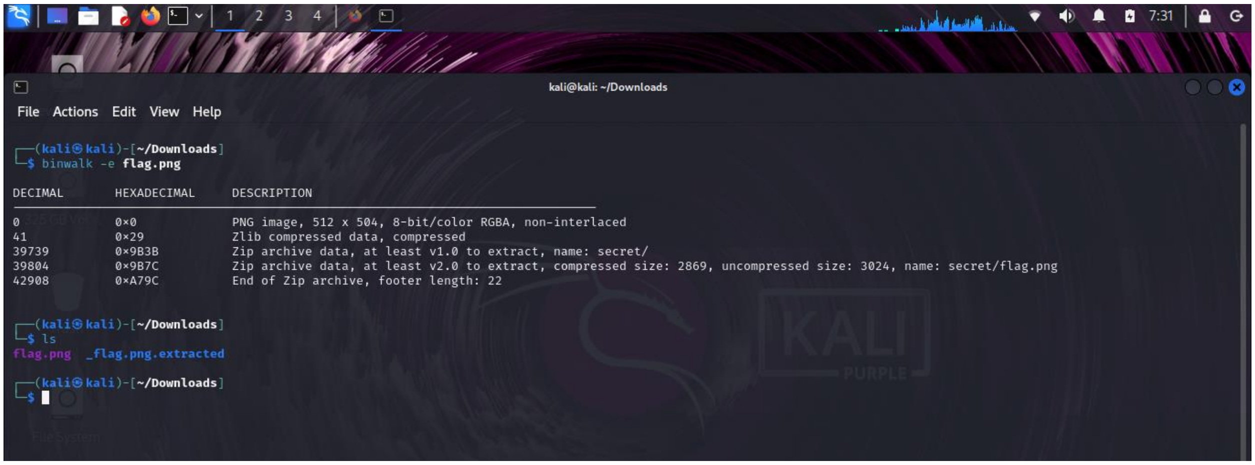
secret/flag.pngUT

This string immediately stood out because it suggests the potential presence of a hidden file path or directory (secret/flag.png). This hinted at the possibility that the real flag might be concealed within a subdirectory or embedded in another file. This clue guided the next steps in solving the challenge.



```
7`_QB\\
##ib{
RP0×3
--e^
I|L_
[u]mC
secret/UT
pVgE#
secret/flag.pngUT

┌──(kali㉿kali)-[~/Downloads]
└─$ 
```

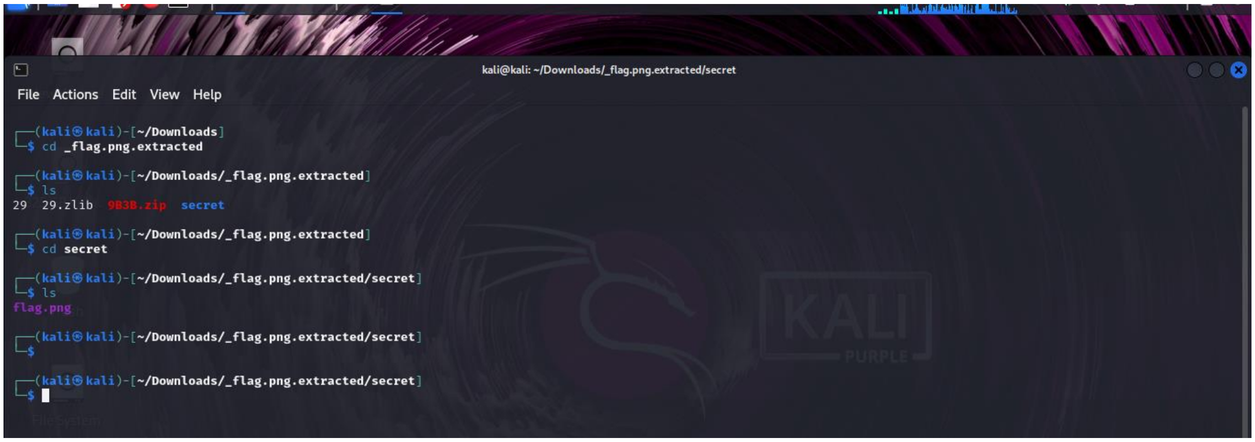
To further investigate the contents of the flag.png file, I used the binwalk tool. This tool is designed to analyze and extract embedded files and data from binary files, such as images or executables. It scans the file for known file signatures, reporting any hidden or embedded files it detects.



```
┌──(kali㉿kali)-[~/Downloads]
└─$ binwalk -e flag.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PNG image, 512 x 504, 8-bit/color RGBA, non-interlaced
41            0×29            Zlib compressed data, compressed
39739         0×9B3B          Zip archive data, at least v1.0 to extract, name: secret/
39804         0×9B7C          Zip archive data, at least v2.0 to extract, compressed size: 2869, uncompressed size: 3024, name: secret/flag.png
42908         0×A79C          End of Zip archive, footer length: 22

┌──(kali㉿kali)-[~/Downloads]
└─$ ls
flag.png  _flag.png.extracted

┌──(kali㉿kali)-[~/Downloads]
└─$ 
```

From this, it became clear that the file contained a hidden ZIP archive with a directory named secret/ and an embedded file named flag.png. After the extraction process, a new folder _flag.png.extracted appeared in the directory, which I decided to investigate next.

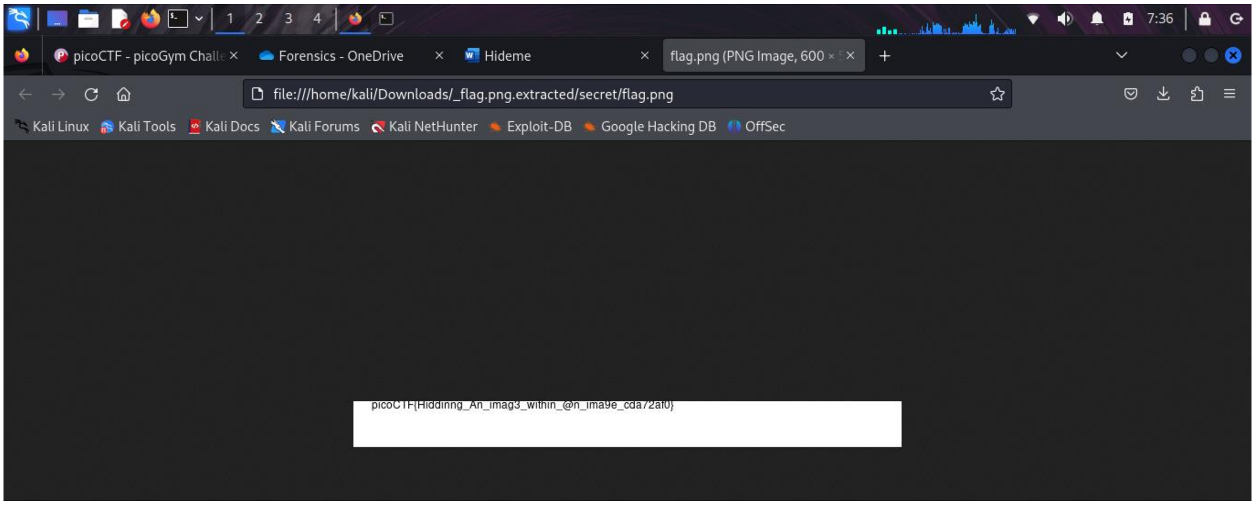
After binwalk completed its extraction, I navigated into the _flag.png.extracted folder. Inside, I observed several files and directories, including 29, 29.zlib, 9B3B.zip, and a directory named secret. Upon further inspection by changing into the secret directory.



kali@kali: ~/Downloads/_flag.png.extracted/secret
File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cd _flag.png.extracted

┌──(kali㉿kali)-[~/Downloads/_flag.png.extracted]
└─$ ls
29  29.zlib  9B3B.zip  secret

┌──(kali㉿kali)-[~/Downloads/_flag.png.extracted]
└─$ cd secret

┌──(kali㉿kali)-[~/Downloads/_flag.png.extracted/secret]
└─$ ls
flag.png

┌──(kali㉿kali)-[~/Downloads/_flag.png.extracted/secret]
└─$

┌──(kali㉿kali)-[~/Downloads/_flag.png.extracted/secret]
└─$ 
```

I discovered the file flag.png, which likely contained the flag. This confirmed the earlier suspicion of a hidden file path indicated by the strings output.



```
picoCTF{Hiddinng_An_imag3_within_@n_ima9e_cda72af0}
```

And that's how i captured the flag.

Conclusion

By systematically analyzing the file, I unraveled the layers of hidden information step by step. Using the strings command revealed a potential hidden file path, which led me to employ binwalk to uncover embedded files. Navigating through the extracted contents confirmed the presence of a hidden directory, and within it, I discovered the flag.png file. This step-by-step approach demonstrated the importance of combining multiple tools and techniques to uncover hidden data in digital forensics. Ultimately, it was through persistence and methodical investigation that I successfully captured the flag, unraveling the mystery hidden within the seemingly ordinary image file.