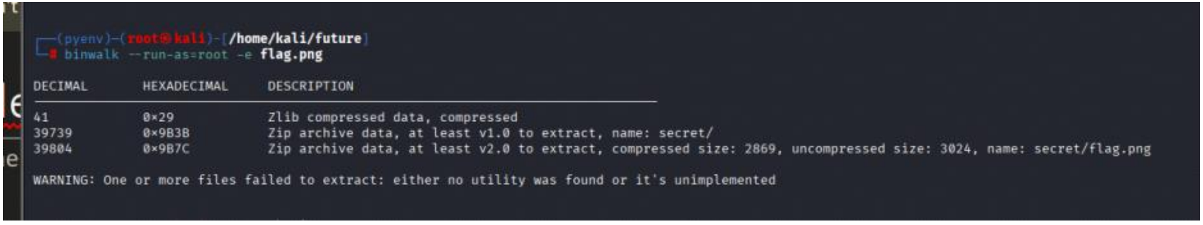
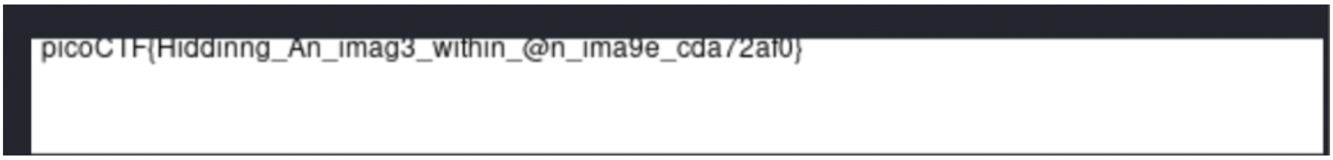
Hideme

18 June 2025 15:14

- On running exiftool on the image we find out something is off, as some warning is given
- Since it’s a png I ran it through zsteg to check if theres some message encoded in LSB it showed that theres a ZIP archive data
- So I used binwalk to extract the zip from the flag, which gives a folder In which theres another image which contains the flag.



```
┌──(pyenv)─(root㉿kali)-[/home/kali/future]
└─# binwalk --run-as=root -e flag.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
41            0x29            Zlib compressed data, compressed
39739         0x9B3B          Zip archive data, at least v1.0 to extract, name: secret/
39804         0x9B7C          Zip archive data, at least v2.0 to extract, compressed size: 2869, uncompressed size: 3024, name: secret/flag.png

WARNING: One or more files failed to extract: either no utility was found or it's unimplemented
```

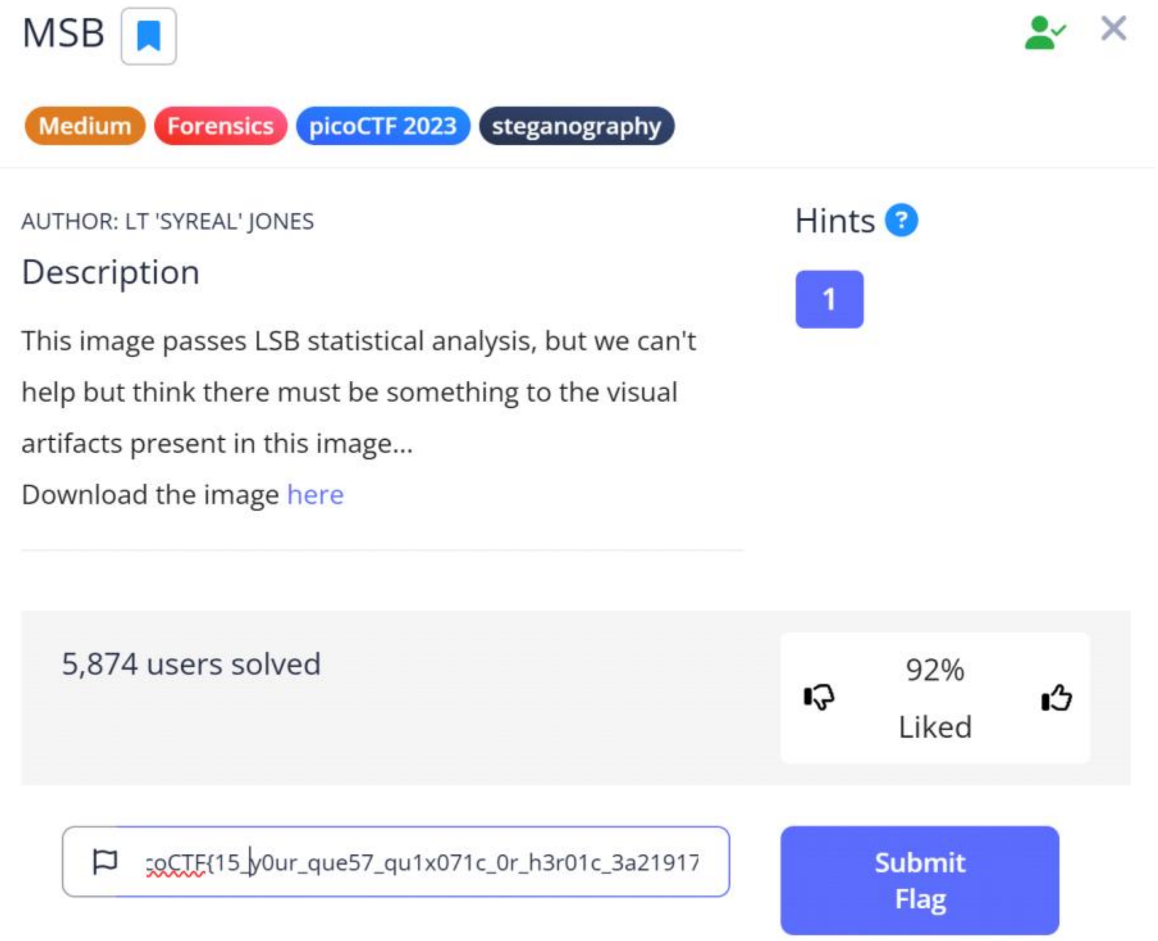


picoCTF{Hiddinng_An_imag3_within_@n_ima9e_cda72af0}

MSB

18 June 2025 15:14

Can be solved by taking the MSB instead of the LSB so using the MSB function on sigbits tool or instead of modifying the LSB on the python script you could modify the MSB in the decryptor, then use grep to find the flag



MSB

AUTHOR: LT 'SYREAL' JONES

Description

This image passes LSB statistical analysis, but we can't help but think there must be something to the visual artifacts present in this image...

Download the image here

5,874 users solved

92%

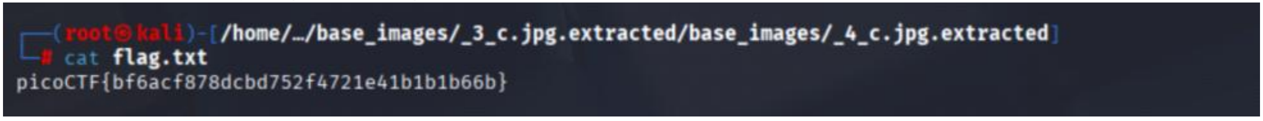
Liked

:oCTF{15_y0ur_que57_qu1x071c_0r_h3r01c_3a21917

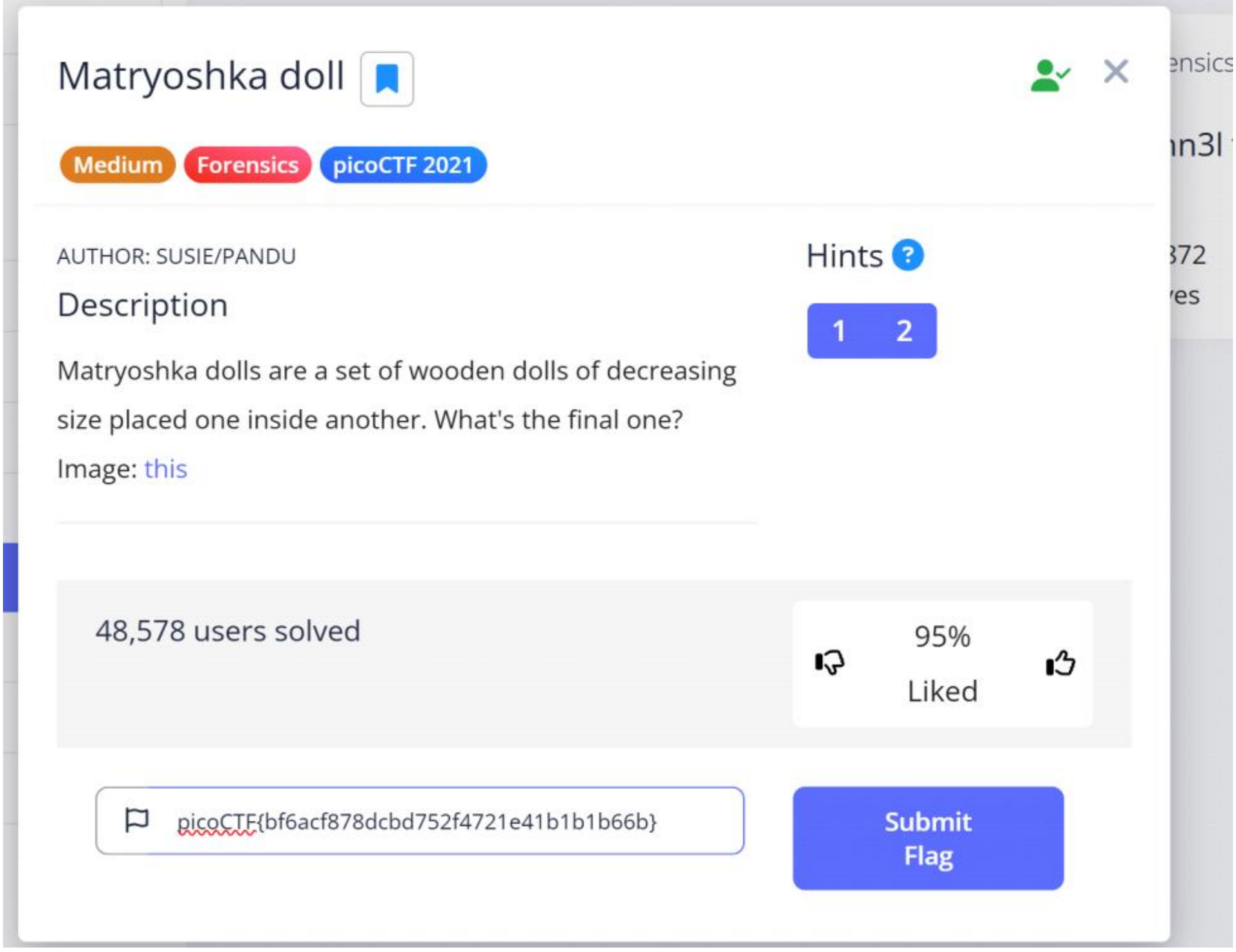
Matryoshka doll

18 June 2025 15:42

Can be solved using binwalk to check the hidden files inside the image , which then on further extraction(5 times) you get the flag.txt.



```
┌──(root㉿kali)-[/home/…/base_images/_3_c.jpg.extracted/base_images/_4_c.jpg.extracted]
└─# cat flag.txt
picoCTF{bf6acf878dcbd752f4721e41b1b1b66b}
```



Matryoshka doll

Medium Forensics picoCTF 2021

AUTHOR: SUSIE/PANDU

Description

Matryoshka dolls are a set of wooden dolls of decreasing size placed one inside another. What's the final one?

Image: this

Hints ?

1 2

48,578 users solved

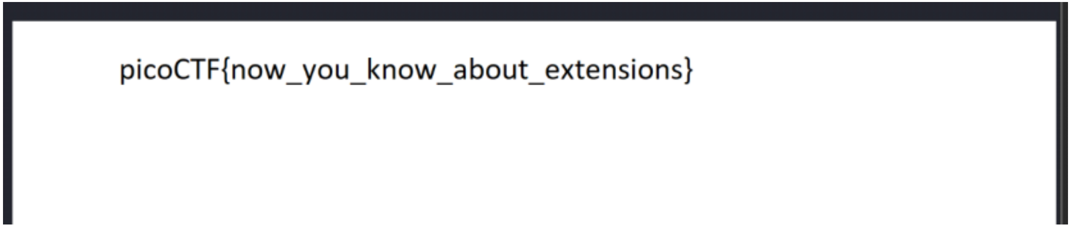
95% Liked

picoCTF{bf6acf878dcbd752f4721e41b1b1b66b}

Extensions

18 June 2025 15:44

- Download the file.
- On opening it the text looks weird
- The clue given says the same , so it might not be a txt file initially.
- So use exiftool to get information about the file
- We find out it’s a png file, so we use mv to convert it back to one.
- On opening we find the flag.

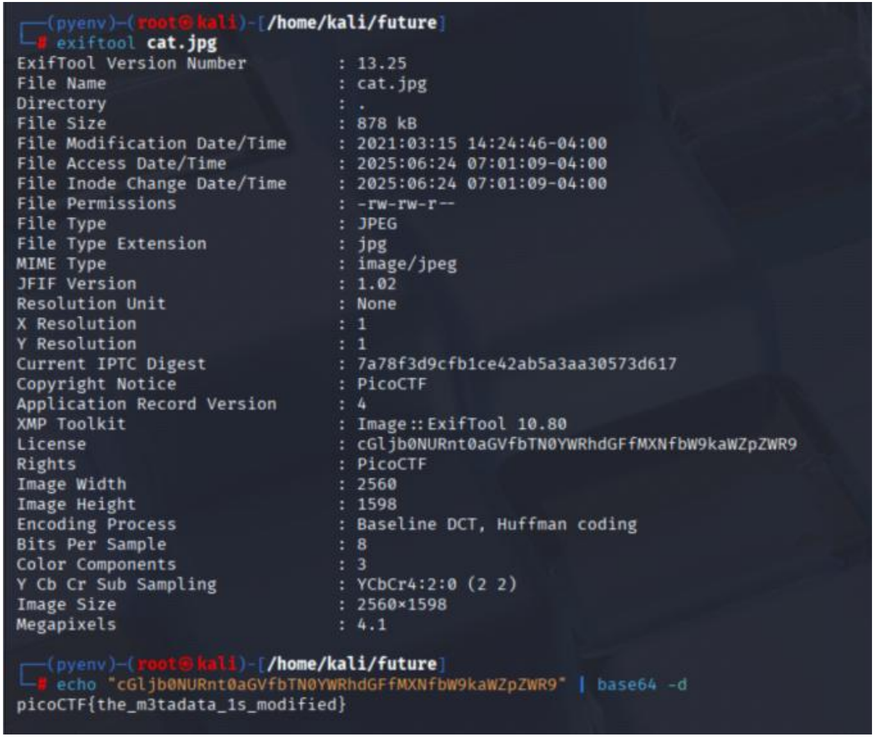


picoCTF{now_you_know_about_extensions}

Information

24 June 2025 16:30

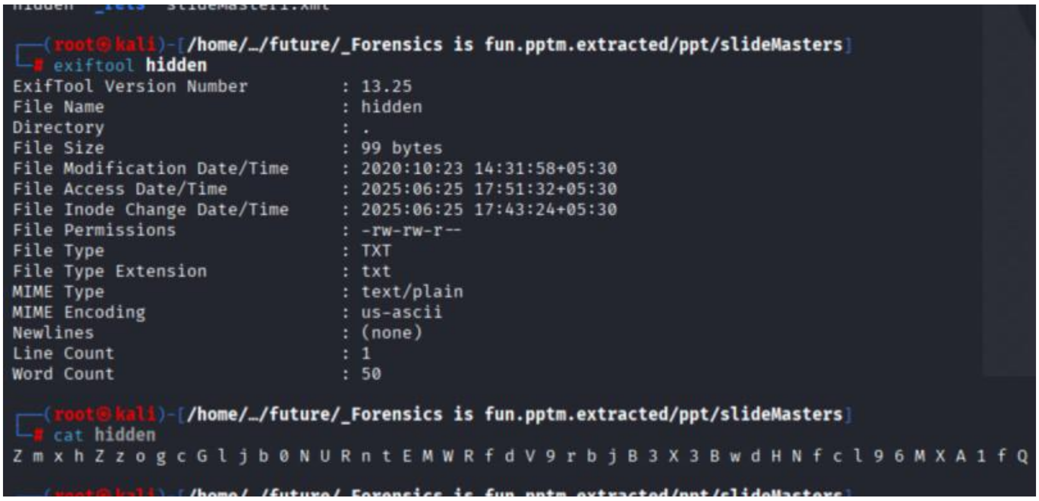
- After Downloading the file, like usually running exiftool to check the information on the file.
- Some parts of the file look distinct, the license looks like base64 which on checking confirms to be so.
- Hence Decoding it might give the answer and which it does



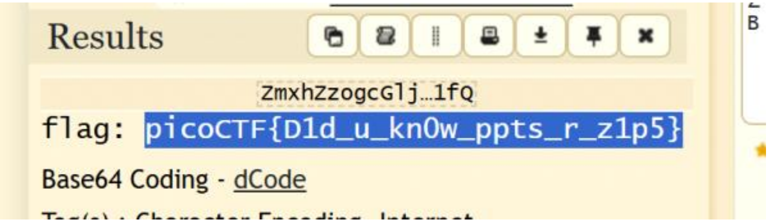
```
┌──(pyenv)─(root㉿kali)-[/home/kali/future]
└─# exiftool cat.jpg
ExifTool Version Number         : 13.25
File Name                       : cat.jpg
Directory                       : .
File Size                       : 878 kB
File Modification Date/Time     : 2021:03:15 14:24:46-04:00
File Access Date/Time           : 2025:06:24 07:01:09-04:00
File Inode Change Date/Time     : 2025:06:24 07:01:09-04:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Current IPTC Digest             : 7a78f3d9cfb1ce42ab5a3aa30573d617
Copyright Notice                : PicoCTF
Application Record Version      : 4
XMP Toolkit                     : Image::ExifTool 10.80
License                         : cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9
Rights                          : PicoCTF
Image Width                     : 2560
Image Height                    : 1598
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                      : 2560x1598
Megapixels                      : 4.1

┌──(pyenv)─(root㉿kali)-[/home/kali/future]
└─# echo "cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9" | base64 -d
picoCTF{the_m3tadata_1s_modified}
```

MacroHard WeakEdge

24 June 2025 16:40

- Downloaded the ppt in the question.
- Opened it but had no luck.
- Tried extracting with binwalk and it did extract
- Looked the folders and found a file named hidden which struck.
- Proceeded to read the contents which seemed to be encoded.
- 

```
┌──(root㉿kali)-[/home/…/future/_Forensics is fun.pptm.extracted/ppt/slideMasters]
└─# exiftool hidden
ExifTool Version Number         : 13.25
File Name                       : hidden
Directory                       : .
File Size                       : 99 bytes
File Modification Date/Time     : 2020:10:23 14:31:58+05:30
File Access Date/Time           : 2025:06:25 17:51:32+05:30
File Inode Change Date/Time     : 2025:06:25 17:43:24+05:30
File Permissions                : -rw-rw-r--
File Type                       : TXT
File Type Extension             : txt
MIME Type                       : text/plain
MIME Encoding                   : us-ascii
Newlines                        : (none)
Line Count                      : 1
Word Count                      : 50

┌──(root㉿kali)-[/home/…/future/_Forensics is fun.pptm.extracted/ppt/slideMasters]
└─# cat hidden
Z m x h Z z o g c G l j b 0 N U R n t E M W R f d V 9 r b j B 3 X 3 B w d H N f c l 9 6 M X A 1 f Q
```

- Then ran it through a decoder to get the flag.
- 

ZmxhZzogcGlj…1fQ

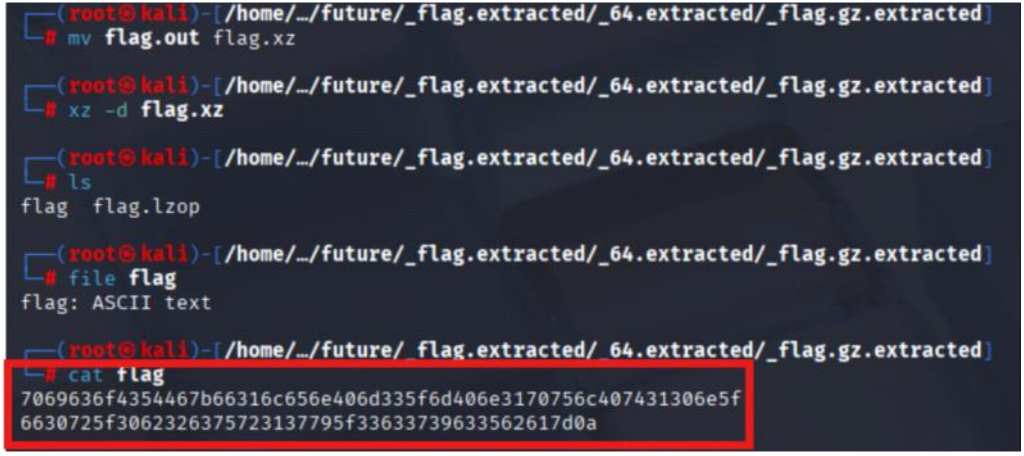
flag: picoCTF{D1d_u_kn0w_ppts_r_z1p5}

Base64 Coding - dCode

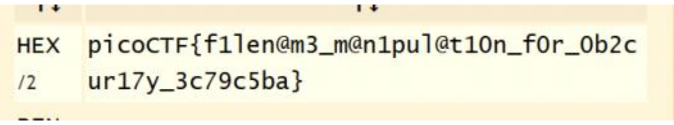
File Types

25 June 2025 15:59

- A lengthier challenge in comparison.
- On downloading we try opening the file which results in failure.
- And as the hint given says check the file types and nests inside it we run the file w the file tool to check the actual file type which turns out to be a script.
- Then we try to run the script as said using sh, but I ran into an error because of the absence of uudecode so I had to install
- Then a flag is created which doesn’t open either so we check the file type its an archive.
- Then on it’s a lot of nested archives of different type extensions like lzop,xz, lzma,etc.
- Finally we'd get an ascii text which is in hex and which on decoding gives the required flag.



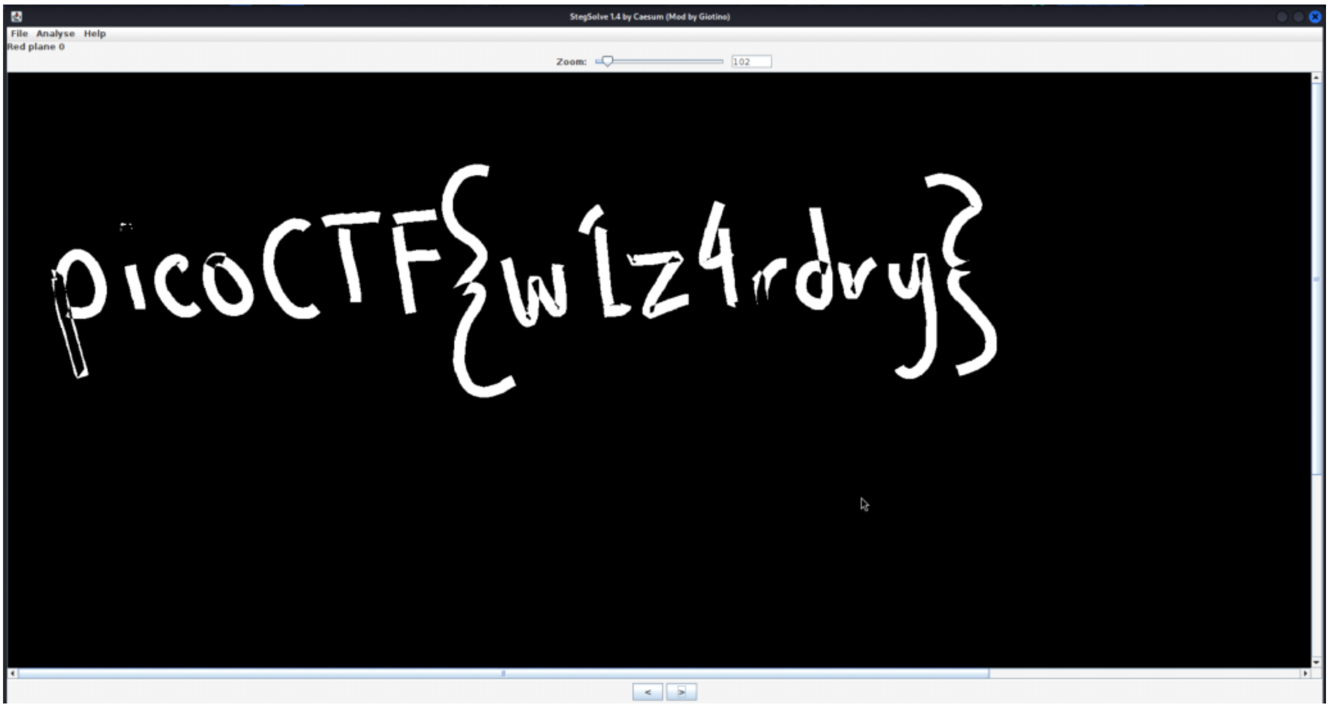
```
┌──(root㉿kali)-[/home/…/future/_flag.extracted/_64.extracted/_flag.gz.extracted]
└─# mv flag.out flag.xz

┌──(root㉿kali)-[/home/…/future/_flag.extracted/_64.extracted/_flag.gz.extracted]
└─# xz -d flag.xz

┌──(root㉿kali)-[/home/…/future/_flag.extracted/_64.extracted/_flag.gz.extracted]
└─# ls
flag  flag.lzop

┌──(root㉿kali)-[/home/…/future/_flag.extracted/_64.extracted/_flag.gz.extracted]
└─# file flag
flag: ASCII text

┌──(root㉿kali)-[/home/…/future/_flag.extracted/_64.extracted/_flag.gz.extracted]
└─# cat flag
7069636f4354467b66316c656e406d335f6d406e3170756c407431306e5f
6630725f3062326375723137795f3363373963356261 7d0a
```



```
HEX  picoCTF{f1len@m3_m@n1pul@t10n_f0r_0b2c
/2   ur17y_3c79c5ba}
```

Advanced Potion Making

26 June 2025 12:04

- Downloaded the file and running it through exiftool and file tools.
- Exiftool returned unknown and File returned "data".
- When read contents of file not readable human language
- Opened it in hexedit, thought maybe the header is off and it was leading w a P.
- So decided to try PNG header, searched up google and put that in
- It did work now as a file but it was an empty red photo.
- Ran it through all types in zsteg tool and no luck
- Then used stegseek to iterate to different types of colour planes and Found the flag on Red Plane 0.
- 

StegSolve 1.4 by Caesum (Mod by Giotino)

Red plane 0

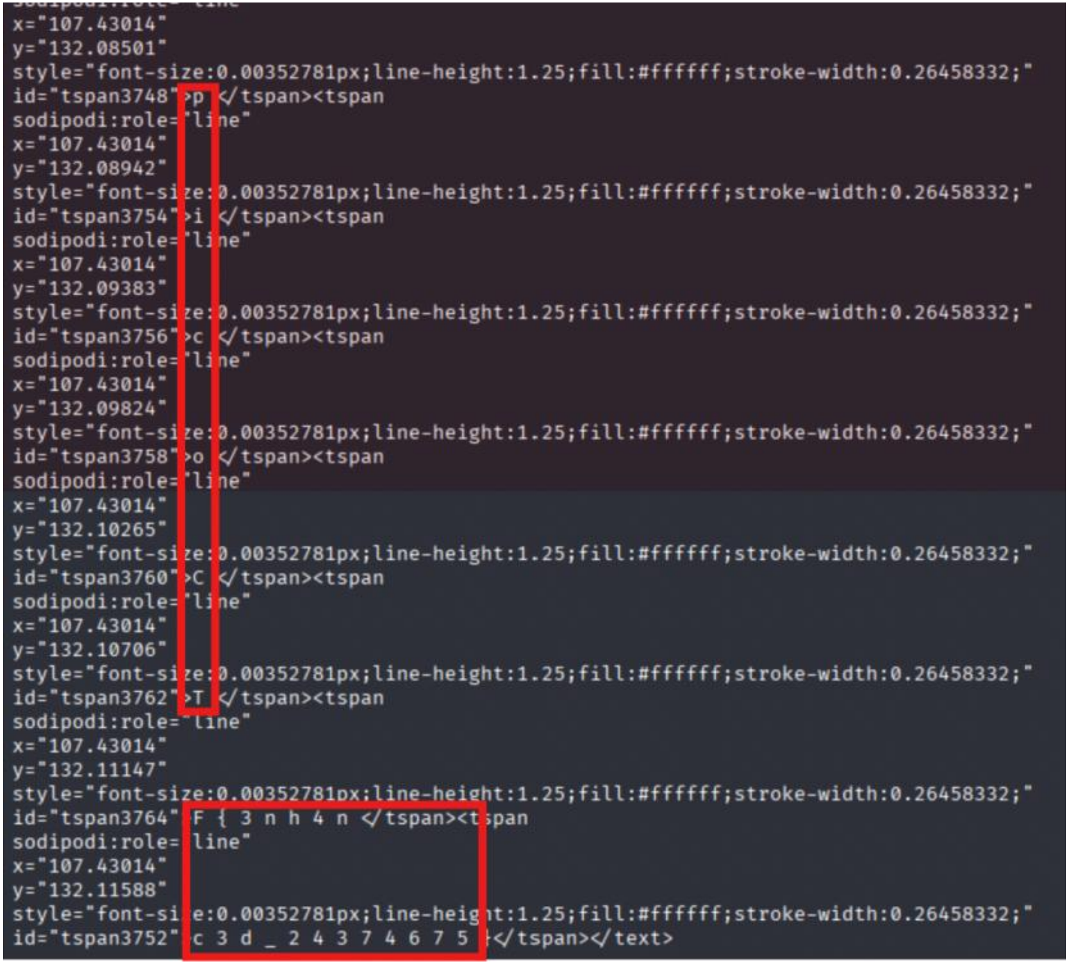
Zoom: 102

picoCTF{w1z4rdry}

Enhance!

26 June 2025 12:09

- On reading the contents of the file we can see



```
sodipodi:role="line"
x="107.43014"
y="132.08501"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3748">p </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.08942"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3754">i </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.09383"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3756">c </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.09824"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3758">o </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.10265"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3760">C </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.10706"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3762">T </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.11147"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3764">F { 3 n h 4 n </tspan><tspan
sodipodi:role="line"
x="107.43014"
y="132.11588"
style="font-size:0.00352781px;line-height:1.25;fill:#ffffff;stroke-width:0.26458332;"
id="tspan3752">c 3 d _ 2 4 3 7 4 6 7 5 }</tspan></text>
```

Which is the flag.

Tunn3l V1s10n

26 June 2025 12:09

- On Downloading and running the file through exiftool we find out it’s a bmp file.
- But it still doesn’t execute on change the extension, so we open it in hexedit to fix the header.
- On opening now we get a fake flag