Challenge 2 Day 1 -420

By team Bingsu Strawberry from IIUM

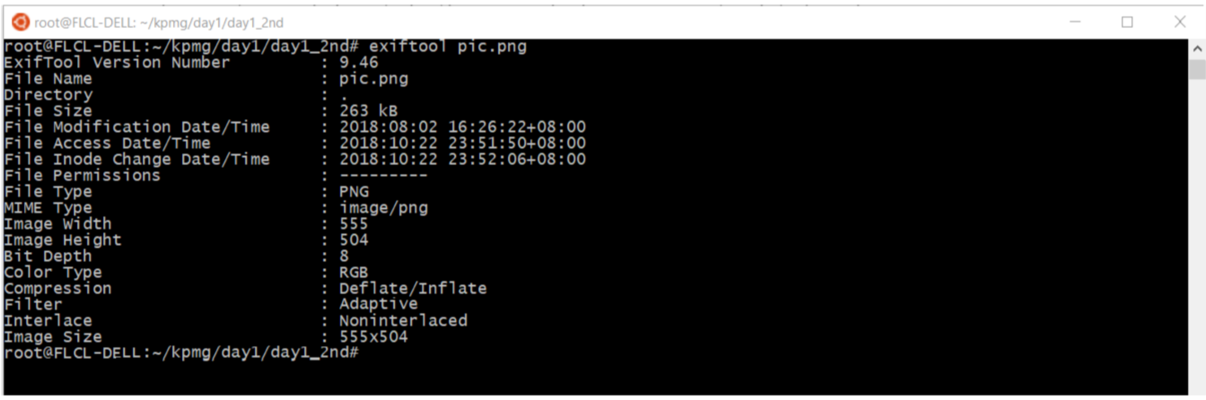
We were given a single image and are to find the flag from it.

Step 1:

Before we go deeper into complex stuff, we zoomed in to look for hidden text but to no avail and found nothing.

Step 2:

We ran exiftool on the image to find anything peculiar about the exifdata

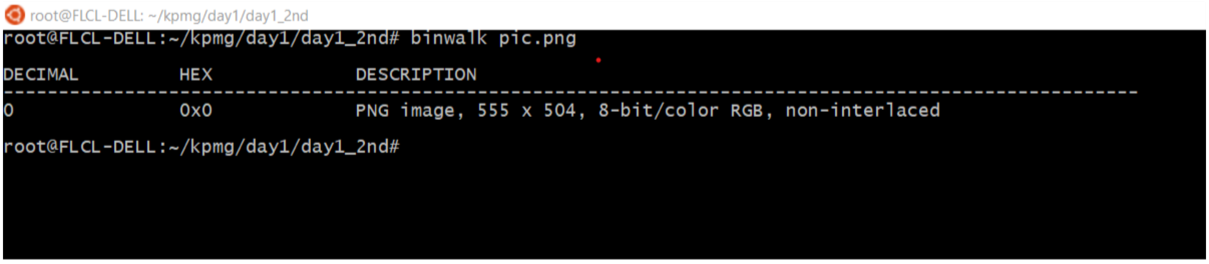


```
root@FLCL-DELL:~/kpmg/day1/day1_2nd# exiftool pic.png
ExifTool Version Number         : 9.46
File Name                       : pic.png
Directory                       : .
File Size                       : 263 kB
File Modification Date/Time     : 2018:08:02 16:26:22+08:00
File Access Date/Time           : 2018:10:22 23:51:50+08:00
File Inode Change Date/Time     : 2018:10:22 23:52:06+08:00
File Permissions                : ---------
File Type                       : PNG
MIME Type                       : image/png
Image Width                     : 555
Image Height                    : 504
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Image Size                      : 555x504
root@FLCL-DELL:~/kpmg/day1/day1_2nd#
```

Unfortunately there’s nothing that stood out for us.

Step 3 :

We ran some more analysis tools on the image. First we used binwalk to find any hidden file hidden in the image.



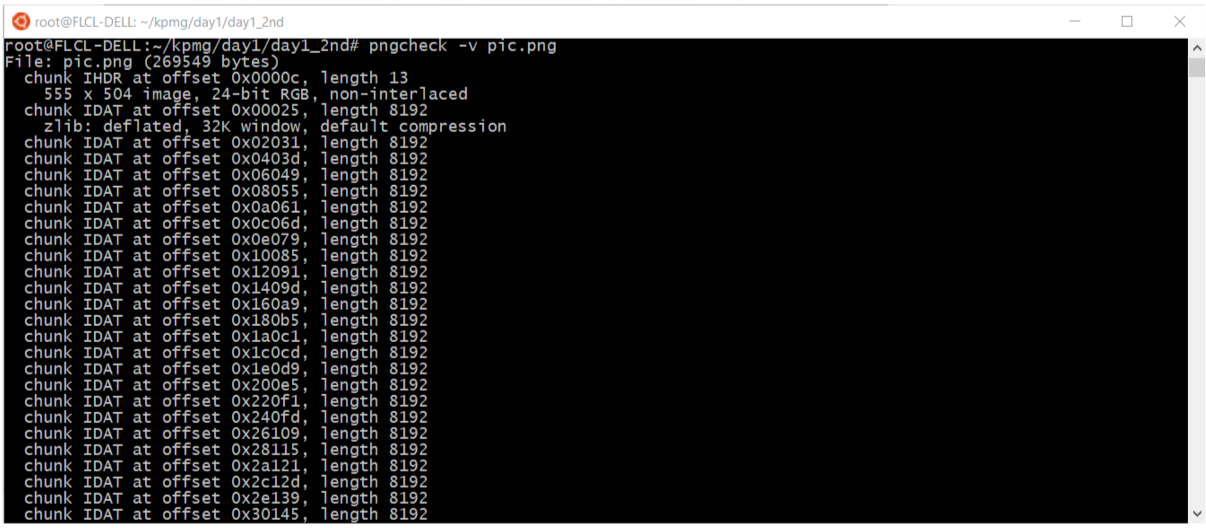
```
root@FLCL-DELL:~/kpmg/day1/day1_2nd# binwalk pic.png

DECIMAL       HEX           DESCRIPTION
--------------------------------------------------------------------------------
0             0x0           PNG image, 555 x 504, 8-bit/color RGB, non-interlaced

root@FLCL-DELL:~/kpmg/day1/day1_2nd#
```

Also to no avail. Binwalk didn’t find anything hidden in between the picture’s pixels.

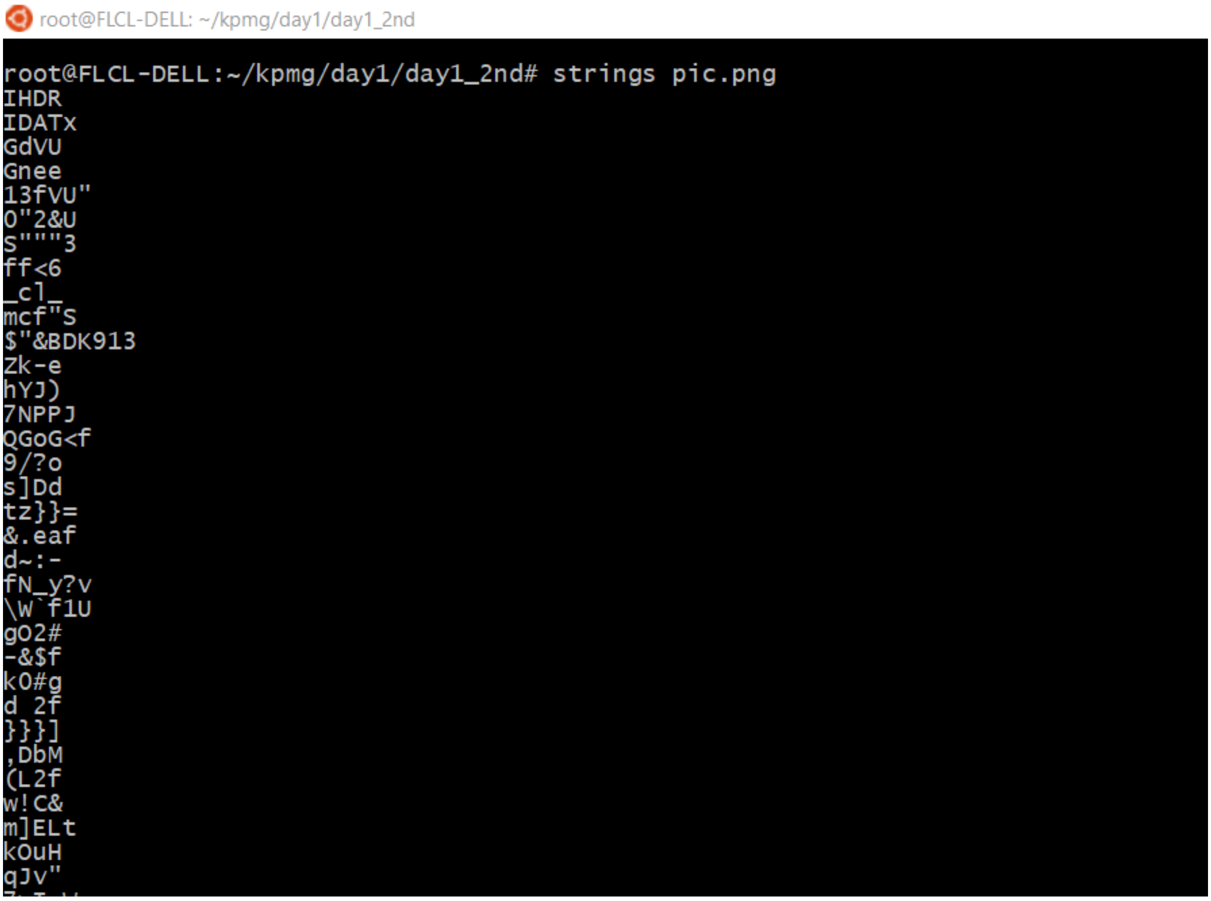
Next is using pngcheck.



```
root@FLCL-DELL:~/kpmg/day1/day1_2nd# pngcheck -v pic.png
File: pic.png (269549 bytes)
  chunk IHDR at offset 0x0000c, length 13
    555 x 504 image, 24-bit RGB, non-interlaced
  chunk IDAT at offset 0x00025, length 8192
    zlib: deflated, 32K window, default compression
  chunk IDAT at offset 0x02031, length 8192
  chunk IDAT at offset 0x0403d, length 8192
  chunk IDAT at offset 0x06049, length 8192
  chunk IDAT at offset 0x08055, length 8192
  chunk IDAT at offset 0x0a061, length 8192
  chunk IDAT at offset 0x0c06d, length 8192
  chunk IDAT at offset 0x0e079, length 8192
  chunk IDAT at offset 0x10085, length 8192
  chunk IDAT at offset 0x12091, length 8192
  chunk IDAT at offset 0x1409d, length 8192
  chunk IDAT at offset 0x160a9, length 8192
  chunk IDAT at offset 0x180b5, length 8192
  chunk IDAT at offset 0x1a0c1, length 8192
  chunk IDAT at offset 0x1c0cd, length 8192
  chunk IDAT at offset 0x1e0d9, length 8192
  chunk IDAT at offset 0x200e5, length 8192
  chunk IDAT at offset 0x220f1, length 8192
  chunk IDAT at offset 0x240fd, length 8192
  chunk IDAT at offset 0x26109, length 8192
  chunk IDAT at offset 0x28115, length 8192
  chunk IDAT at offset 0x2a121, length 8192
  chunk IDAT at offset 0x2c12d, length 8192
  chunk IDAT at offset 0x2e139, length 8192
  chunk IDAT at offset 0x30145, length 8192
```

Pngcheck didn’t provide any other clue for us aswell. We are quickly running out of ideas.

Running strings command on the image provided us with this result that lacks any real information.

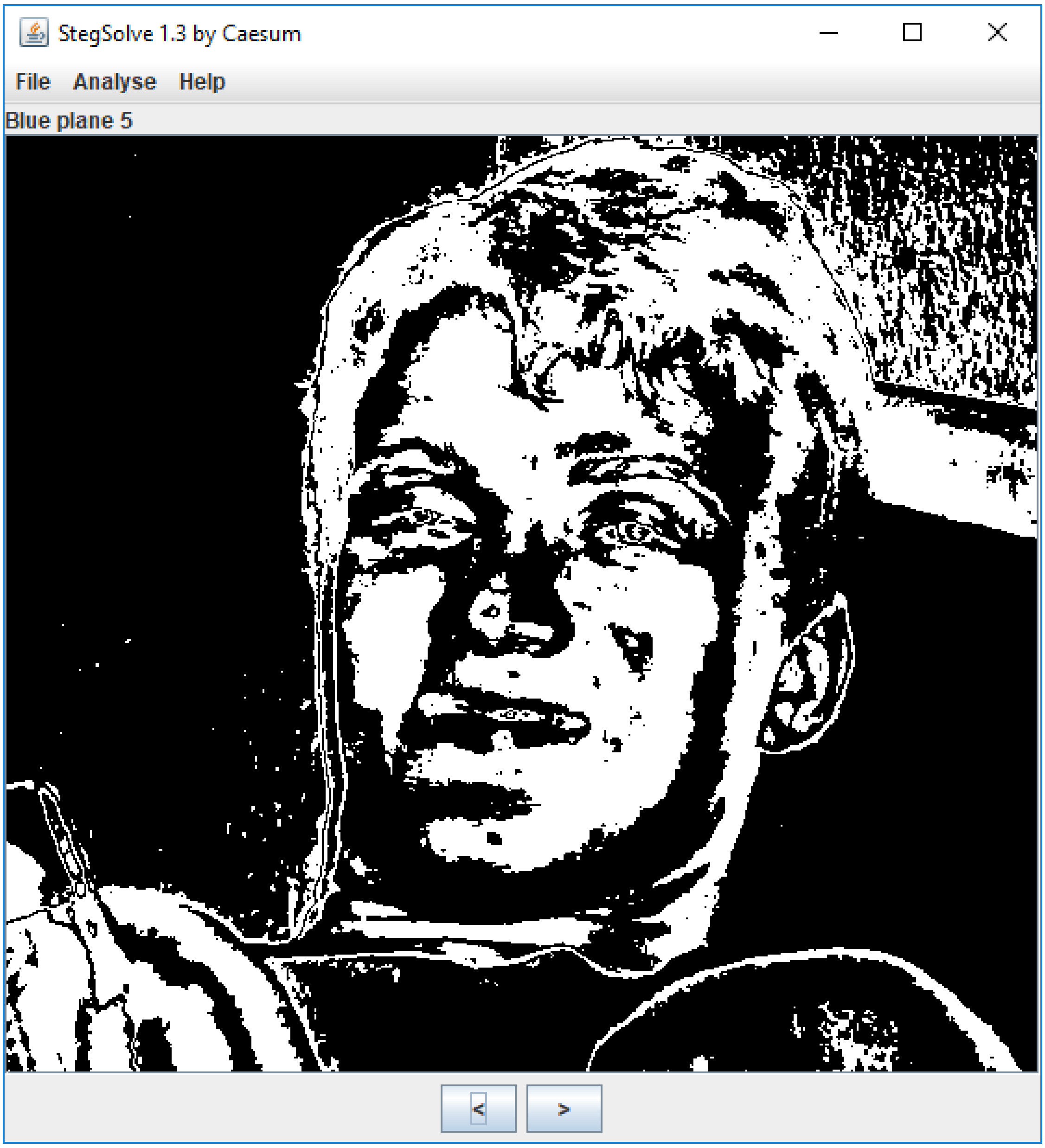


```
root@FLCL-DELL:~/kpmg/day1/day1_2nd# strings pic.png
IHDR
IDATx
GdVU
Gnee
13fVU"
0"2&U
S"""3
ff<6
_cl_
mcf"S
$"&BDK913
Zk-e
hYJ)
7NPPJ
QGoG<f
9/?o
s]Dd
tz}}=
&.eaf
d~:-
fN_y?v
\W`f1U
gO2#
-&$f
k0#g
d 2f
}}}]
,DbM
(L2f
w!C&
m]ELt
kOuH
qJv"
```

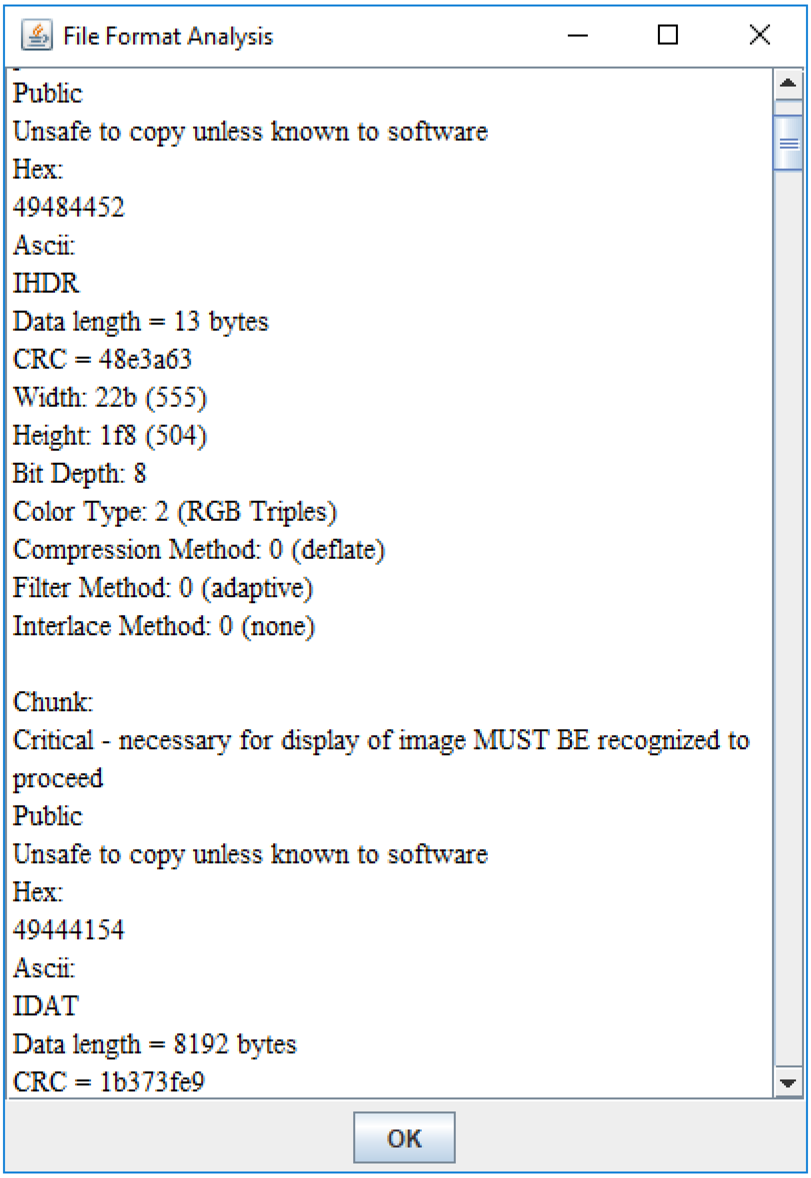
We are quickly running out of ideas and now are just trying anything that we can.

Step 4:

It’s time for the ol’ Google trip. We found out about a steganography method that hides texts using different colormaps. We downloaded a tool named Stegsolve that allow us to scroll through different colormaps and find any texts.



Unfortunately, we failed to find any texts hidden in the colormaps. We ran some more analysis tools provided by Stegsolve.



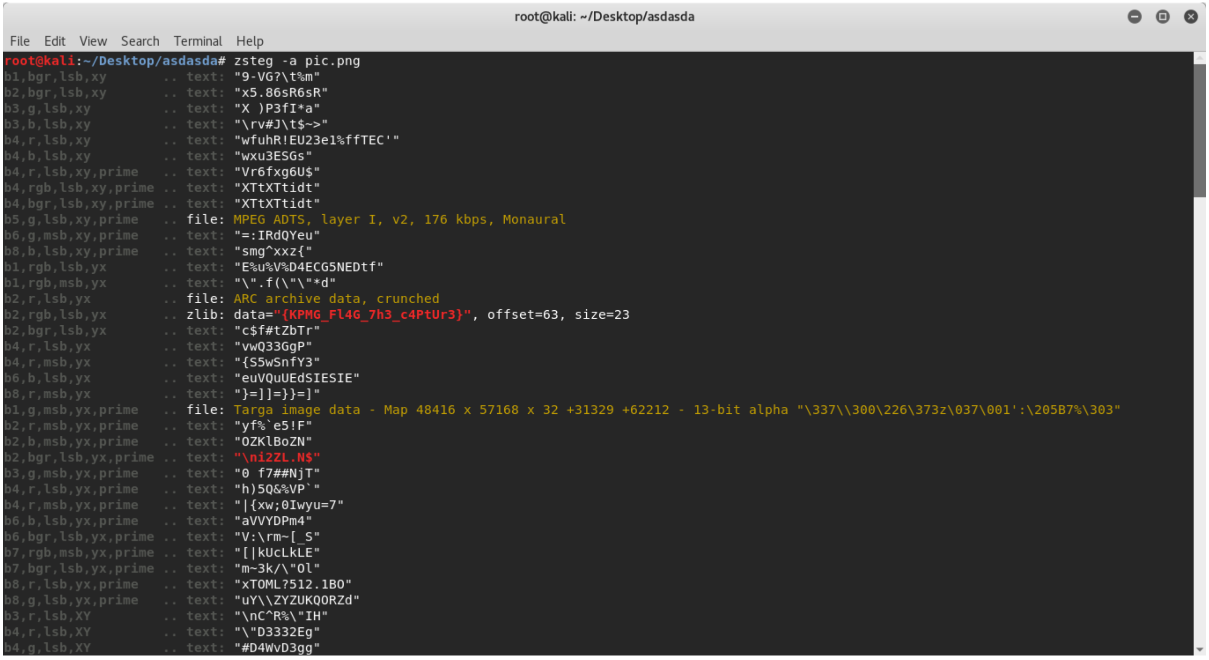
```
Public
Unsafe to copy unless known to software
Hex:
49484452
Ascii:
IHDR
Data length = 13 bytes
CRC = 48e3a63
Width: 22b (555)
Height: 1f8 (504)
Bit Depth: 8
Color Type: 2 (RGB Triples)
Compression Method: 0 (deflate)
Filter Method: 0 (adaptive)
Interlace Method: 0 (none)

Chunk:
Critical - necessary for display of image MUST BE recognized to
proceed
Public
Unsafe to copy unless known to software
Hex:
49444154
Ascii:
IDAT
Data length = 8192 bytes
CRC = 1b373fe9
```

We scrolled through the text and found nothing. We are now like an adventurer stuck in a desert. We are dying to get our hands on something that’s not a barren wasteland.

Step 5:

Now we truly have no more idea on what to do. We just looked through some old CTF online on CTFtime.org .One of the writeup had to deal with a .png like us and used a tool called zsteg. We downloaded it and ran it on our image .



root@kali: ~/Desktop/asdasda
File Edit View Search Terminal Help

```
root@kali:~/Desktop/asdasda# zsteg -a pic.png
b1,bgr,lsb,xy       .. text: "9-VG?\t%m"
b2,bgr,lsb,xy       .. text: "x5.86sR6sR"
b3,g,lsb,xy         .. text: "X )P3fI*a"
b3,b,lsb,xy         .. text: "\rv#J\t$~>"
b4,r,lsb,xy         .. text: "wfuhR!EU23e1%ffTEC'"
b4,b,lsb,xy         .. text: "wxu3ESGs"
b4,r,lsb,xy,prime   .. text: "Vr6fxg6U$"
b4,rgb,lsb,xy,prime .. text: "XTtXTtidt"
b4,bgr,lsb,xy,prime .. text: "XTtXTtidt"
b5,g,lsb,xy,prime   .. file: MPEG ADTS, layer I, v2, 176 kbps, Monaural
b6,g,msb,xy,prime   .. text: "=:IRdQYeu"
b8,b,lsb,xy,prime   .. text: "smg^xxz{"
b1,rgb,lsb,yx       .. text: "E%u%V%D4ECG5NEDtf"
b1,rgb,msb,yx       .. text: "\".f(\"\"*d"
b2,r,lsb,yx         .. file: ARC archive data, crunched
b2,rgb,lsb,yx       .. zlib: data="{KPMG_Fl4G_7h3_c4PtUr3}", offset=63, size=23
b2,bgr,lsb,yx       .. text: "c$f#tZbTr"
b4,r,lsb,yx         .. text: "vwQ33GgP"
b4,r,msb,yx         .. text: "{S5wSnfY3"
b6,b,lsb,yx         .. text: "euVQuUEdSIESIE"
b8,r,msb,yx         .. text: "}=]]=}}=]"
b1,g,msb,yx,prime   .. file: Targa image data - Map 48416 x 57168 x 32 +31329 +62212 - 13-bit alpha "\337\\300\226\373z\037\001':\205B7%\303"
b2,r,msb,yx,prime   .. text: "yf%`e5!F"
b2,b,msb,yx,prime   .. text: "OZKlBoZN"
b2,bgr,lsb,yx,prime .. text: "\ni2ZL.N$"
b3,g,msb,yx,prime   .. text: "0 f7##NjT"
b4,r,lsb,yx,prime   .. text: "h)5Q&%VP`"
b4,r,msb,yx,prime   .. text: "|{xw;0Iwyu=7"
b6,b,lsb,yx,prime   .. text: "aVVYDPm4"
b6,bgr,lsb,yx,prime .. text: "V:\rm~[_S"
b7,rgb,msb,yx,prime .. text: "[|kUcLkLE"
b7,bgr,lsb,yx,prime .. text: "m~3k/\"Ol"
b8,r,lsb,yx,prime   .. text: "xTOML?512.1BO"
b8,g,lsb,yx,prime   .. text: "uY\\ZYZUKQORZd"
b3,r,lsb,XY         .. text: "\nC^R%\"IH"
b4,r,lsb,XY         .. text: "\"D3332Eg"
b4,g,lsb,XY         .. text: "#D4WvD3gg"
```

We finally see something weird. The zlib file is a data with the label {KPMG_Fl4G_7h3_C4PtUr3}. Can this be the flag? But the format is different.

But we ran out of time and options. So, we are submitting this as our flag.

{KPMG_Fl4G_7h3_C4PtUr3}.