

LinkedIn

Description

The Network Operations Center (NOC) of your local institution picked up a suspicious file, they're getting conflicting information on what type of file it is. They've brought you in as an external expert to examine the file. Can you extract all the information from this strange file?

Download the suspicious file [here](#).

***Hint:** This problem can be solved by just opening the file in different ways*

Solution

I started by downloading the suspicious file, which had a .pdf extension. Upon closer inspection using the **file** command, I found that it wasn't actually a PDF but a PNG image. I renamed the file to flag.png to reflect its true format and opened it in Firefox.

```
File Actions Edit View Help
(kali@kali)-[~/Downloads]
└─$ ls
flag2of2-final.pdf
(kali@kali)-[~/Downloads]
└─$ file flag2of2-final.pdf
flag2of2-final.pdf: PNG image data, 50 x 50, 8-bit/color RGBA, non-interlaced
(kali@kali)-[~/Downloads]
└─$ mv flag2of2-final.pdf flag.png
(kali@kali)-[~/Downloads]
└─$ ls
flag.png
(kali@kali)-[~/Downloads]
└─$ firefox flag.png
(kali@kali)-[~/Downloads]
└─$
```

Inside the image, I found part of the flag: **picoCTF{f1u3n7_**



Next, I used binwalk to analyze the file further. The tool revealed that the PNG image contained embedded data, including a Zlib compressed block. I extracted the files, which placed a file named 47D.zlib in a folder called _flag.png.extracted. Inside the 47D file, I found ASCII text that appeared to be a continuation of the flag.

```
(kali@kali)~/Downloads
$ binwalk -e flag.png

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          PNG image, 50 x 50, 8-bit/color RGBA, non-interlaced
914         0x392          PDF document, version: "1.4"
1149        0x47D          Zlib compressed data, default compression

(kali@kali)~/Downloads
$ ls
flag.png  _flag.png.extracted

(kali@kali)~/Downloads
$ cd _flag.png.extracted

(kali@kali)~/Downloads/_flag.png.extracted
$ ls
47D  47D.zlib

(kali@kali)~/Downloads/_flag.png.extracted
$ file 47D
47D: ASCII text

(kali@kali)~/Downloads/_flag.png.extracted
$ cat 47D
q 0.1 0 0 0.1 0 0 cm
0 g
q
10 0 0 10 0 0 cm BT
/R7 16 Tf
1 0 0 1 50 250 Tm
(in_pn9_6_pdf_249d05c0))Tj
ET
Q
Q
```

After combining this with the partial flag from the image, I recovered the full flag:

picoCTF{f1u3n7_1n_pn9_&_pdf_249d05c0}