

# Write up - Challenge XOR

Attention ce challenge n'a qu'une seule possibilité de flag, il faut donc être sûr de reverse l'image avant de valider.

Donc en analysant l'image avec un outil comme xxd :

```
[grootme@grootme-parl0c] ~/Documents/ctf/2017/ctf/parl0c
$xxd \? \? \? \? .jpg
00000000: 9bb7 908d 6574 2e29 262b 6565 656f 6f6c  ....et.)&+eeeeool
00000010: 6565 646f 90b6 65e0 646a 6a68 6061 616a  eedo..e.djjh`aaj
00000020: 696b 606c 6c68 6765 6e6e 6d66 6566 7468  ik`llhgenmfefth
00000030: 6963 6261 747e 747c 7f7d 7674 7e78 747b  icbat~t|.}vt~xt{
00000040: 7072 7f78 464d 7978 4446 404a 4043 4b56  pr.xFMyxDF@J@CKV
00000050: 5c5e 5c23 2028 3230 1865 616a 6a68 6061  \^ \# (20.eajjh`a
00000060: 6169 6968 6d6c 6367 6766 6f6d 6d65 647c  aihmlcggfommed|
00000070: 6969 6862 637c 7f74 777f 7f7e 757e 7374  iihbc|.tw..~u~st
00000080: 7978 737f 7346 4f71 7944 4d40 4848 424b  yxs.sFOqyDM@HHBK
00000090: 5d5c 5c54 2220 2332 3210 9aa6 647e 676c  ]\T" #22...d~gl
000000a0: ed65 7c6c 6e4f 6566 756e 6c7c 649b a06f  .elln0efunlld..o
000000b0: 596d 6565 606c 6e6c 6564 646f 6f6d 6564  Ymee`lnleddoomed
000000c0: 646f 6a6e 6162 636f 6d65 646d 656f 6d6e  dojnabcomedmeomn
000000d0: 6465 656f 6f6d 6564 646f 6f6d 6560 616d  deeoomeddoome`am
000000e0: 6c6b 6464 6390 b56d 6967 656f 6d7d 6674  lkddc..migeom}ft
000000f0: 646f 6fcd 77a9 1710 03ba 0a59 f176 80e4  doo.w.....Y.v..
00000100: 1ebf 8ef0 fc78 3927 416a de52 1881 3204  ....x9'Aj.R..2.
00000110: d364 19cf 7986 5ebf c871 eb32 8d13 3079  .d..y.^..q.2..0y
```

On se rend compte d'un mot ou plusieurs qui apparaissent, "doom", "doomed", "doome". Ceci est dû à des bytes égaux à 0, car  $a^0 = a$ .

En créant un script python avec ces 3 clés là nous pouvons réussir à Xorer l'image à nouveau et à retrouver notre poster !!!

Script python de résolution avec la bonne clé:

Pour tester avec d'autre clé, il suffit de changer le paramètre key et la condition `if(i==6):`

```
def xor(img1, key):
    i=0
    for y in range(len(img1)):
        img1[y] ^= key[i]
        i += 1
        if(i==6):
            i = 0

def main():

    img1 = bytearray(open('?????.jpg', 'rb').read())
    xor(img1, [0x64, 0x6f, 0x6f, 0x6d, 0x65, 0x64]) # doomed
    open('doom_unxored2.jpg', 'wb').write(img1)

if __name__ == '__main__':
    main()
```

Et hop nous retrouvons notre poste !!!



Le flag de validation est donc : NHM2I{doomed}