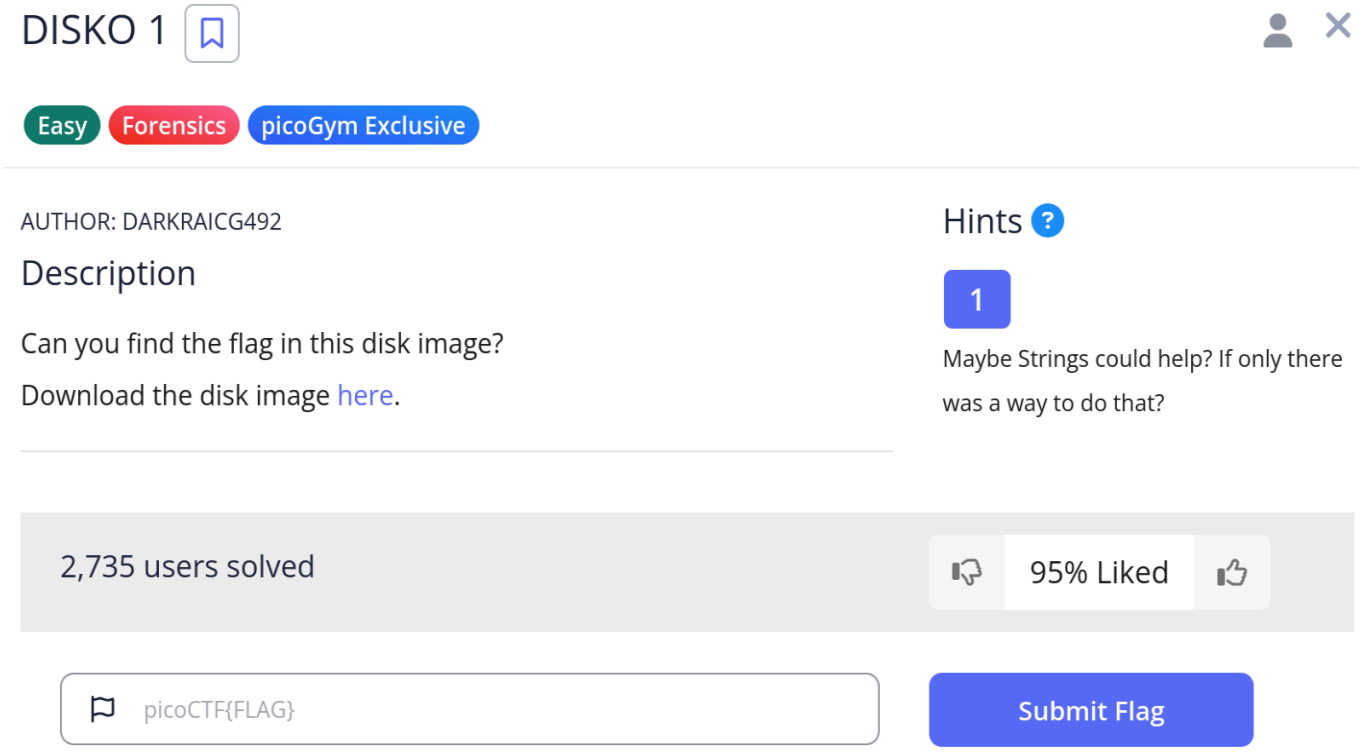
DISKO1



The file was a `.dd` extention. I've never handled such file so I did a bunch of research.

went down a rabbit hole trying to read the disk image by mounting it to my file system and eventually found out about the strings command.

1. Used `gzip` to decompress the zip file

2. Used `strings` piped into `grep` to look for strings in the disk image with the key word pico

```
┌──(kali㉿kali)-[~/CTF_Files/Pico/DISKO1]
└─$ strings disko-1.dd | grep pico
:/icons/appicon
# $Id: piconv,v 2.8 2016/08/04 03:15:58 dankogai Exp $
piconv -- iconv(1), reinvented in perl
  piconv [-f from_encoding] [-t to_encoding]
  piconv -l
  piconv -r encoding_alias
  piconv -h
B<piconv> is perl version of B<iconv>, a character encoding converter
a technology demonstrator for Perl 5.8.0, but you can use piconv in the
piconv converts the character encoding of either STDIN or files
Therefore, when both -f and -t are omitted, B<piconv> just acts
picoCTF{1t5_ju5t_4_5tr1n9_e3408eef}

┌──(kali㉿kali)-[~/CTF_Files/Pico/DISKO1]
└─$ strings disko-1.dd | grep picoCTF
picoCTF{1t5_ju5t_4_5tr1n9_e3408eef}
```