



CAPTURE THE FLAG – I'M A DUMP

Autor: ETR00M

Github: <https://github.com/ETR00M/>

Linkedin: <https://www.linkedin.com/in/ls-anderson/>

Link da Challenge: <https://ctflearn.com/challenge/883>

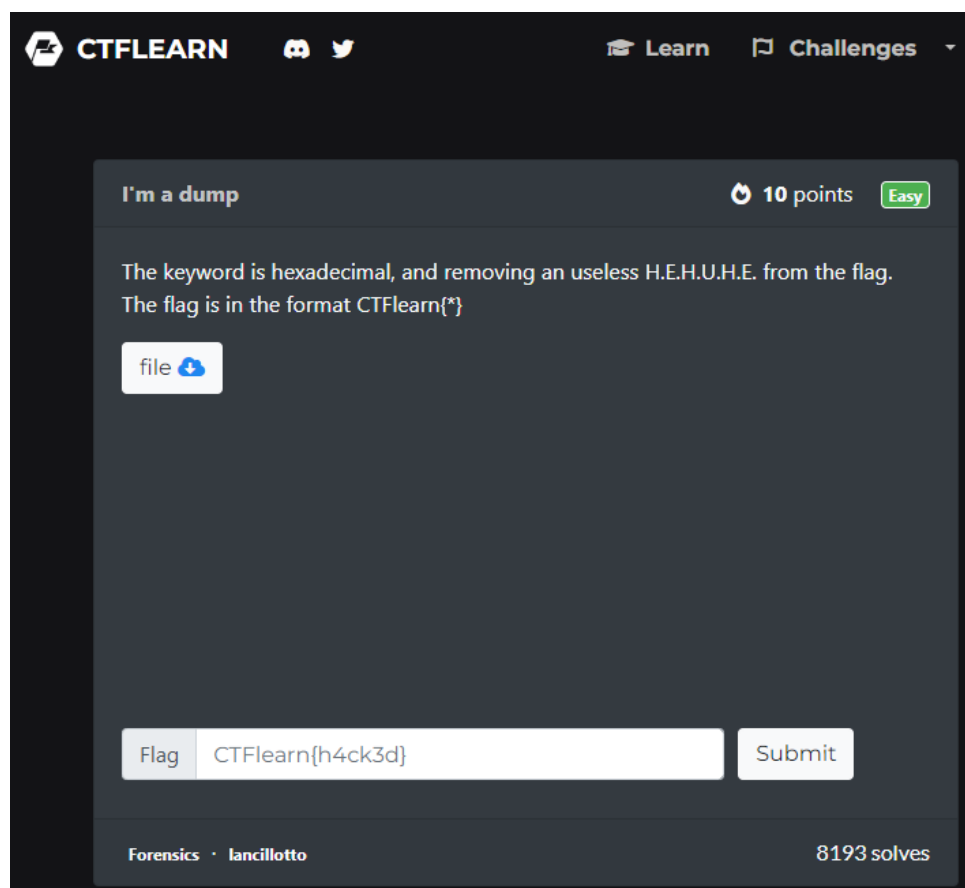
Nível: fácil;

Categoria: *Forensics*;

Tag: hexadecimal, comandos Linux (xxd, hexdump), pensamento linear.



Neste desafio do **CTFLearn** precisaremos analisar o hexadecimal do arquivo “**file**” disponibilizado pelo autor da *challenge*, pelas informações na descrição do desafio após avaliar o arquivo em hexadecimal teremos que remover os caracteres “**H.E.H.U.H.E**” para montar a *flag*.





Verificando as propriedades do arquivo baixado notamos que ele é um executável para sistemas operacionais Linux:

```
(kali㉿kali)-[~/Downloads]
$ ls -la
total 28
drwxr-xr-x  2 kali kali  4096 Feb 21 18:28 .
drwx----- 25 kali kali  4096 Feb 21 16:55 ..
-rw-r--r--  1 kali kali 16560 Feb 21 18:26 file

(kali㉿kali)-[~/Downloads]
$ file file
file: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, inter
preter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=672d1ab79b5c1f063344be7b8edbda2219d899
1d, for GNU/Linux 3.2.0, not stripped
```

Por se tratar de um executável temos algumas opções para avaliar seu conteúdo, porém a proposta do desafio é efetuar um *dump* hexadecimal do arquivo, sendo assim, utilizarei o *xxd* para coletar essas informações, caso você não tenha conhecimento sobre sistema hexadecimal e comandos Linux para *dump* hexadecimal, recomendo o estudo desses assuntos antes de seguir com o *Writeup*.

Dicas de materiais para estudo:

- https://www.youtube.com/watch?v=gIXiFhEA-Qw&ab_channel=CursoemV%C3%ADdeo
- <https://www.ibm.com/docs/pt-br/aix/7.3?topic=adapters-ascii-decimal-hexadecimal-octal-binary-conversion-table>
- https://www.youtube.com/watch?v=IN9EI090uLc&ab_channel=MenteBin%C3%A1ria
- https://www.youtube.com/watch?v=LYyseHh43vU&ab_channel=MenteBin%C3%A1ria

Comando: **xxd file**




```
(kali㉿kali)-[~/Downloads]
$ xxd file
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF ...
00000010: 0300 3e00 0100 0000 4010 0000 0000 0000  .>...@.
00000020: 4000 0000 0000 0000 7039 0000 0000 0000  @.....p9.
00000030: 0000 0000 4000 3800 0b00 4000 1d00 1c00  ....@.8...@.
00000040: 0600 0000 0400 0000 4000 0000 0000 0000  . ... ..@.
00000050: 4000 0000 0000 0000 4000 0000 0000 0000  @.....@.
00000060: 6802 0000 0000 0000 6802 0000 0000 0000  h.....h.
00000070: 0800 0000 0000 0000 0300 0000 0400 0000  ..... ..
```

A princípio temos como resposta ao comando algumas informações padrão do *ELF Header*, seguindo mais adiante no arquivo encontraremos a *string* “CTFLearn” que conforme a descrição do desafio indica o início da *flag*, porém precisaremos ignorar as sequências de caracteres extras “H.E.U” para remontar a *string* correta:



```
000010d0: ffe0 660f 1f44 0000 c30f 1f80 0000 0000 ..f..D.. .....
000010e0: f30f 1efa 803d 452f 0000 0075 3355 4883 .....=E/ ...u3UH.
000010f0: 3d02 2f00 0000 4889 e574 0d48 8b3d 262f =./ ...H..t.H.=δ/
00001100: 0000 ff15 f02e 0000 e863 ffff ffc6 051c .. ... ..c... ..
00001110: 2f00 0001 5dc3 662e 0f1f 8400 0000 0000 /..].f. ....
00001120: c366 662e 0f1f 8400 0000 0000 0f1f 4000 .ff. ....@.
00001130: f30f 1efa e967 ffff ff55 4889 e548 83ec .....g...UH..H..
00001140: 3064 488b 0425 2800 0000 4889 45f8 31c0 0dH..%( ...H.E.1.
00001150: 48b8 4354 466c 6561 726e 48ba 7b66 6c34 H. ....H.
00001160: 6767 7966 4889 45d0 4889 55d8 48c7 45e0 ....H.E.H.U.H.E.
00001170: 6c34 677d 48c7 45e8 0000 0000 48c7 45f0 ....H.E....H.E.
00001180: 0000 0000 9048 8b45 f864 4833 0425 2800 .....H.E.dH3.%(
00001190: 0000 7405 e897 feff ffc9 c30f 1f44 0000 ..t.... ..D..
000011a0: f30f 1efa 4157 4c8d 3d3b 2c00 0041 5649 ....AWL.=; ..AVI
000011b0: 89d6 4155 4989 f541 5441 89fc 5548 8d2d ..AUI ..ATA ..UH.-
000011c0: 2c2c 0000 534c 29fd 4883 ec08 e82f feff ,, ..SL).H..../.
000011d0: ff48 c1fd 0374 1f31 db0f 1f80 0000 0000 .H ...t.1.....
000011e0: 4c89 f24c 89ee 4489 e741 ff14 df48 83c3 L..L..D..A..H..
000011f0: 0148 39dd 75ea 4883 c408 5b5d 415c 415d .H9.u.H...[A\A]
00001200: 415e 415f c366 662e 0f1f 8400 0000 0000 A^A_.ff. ....
00001210: f30f 1efa c300 0000 f30f 1efa 4883 ec08 ..... ..H...
00001220: 4883 c408 c300 0000 0000 0000 0000 0000 H.....
00001230: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001240: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00001250: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```


Para concluir a *challenge* basta submetermos a *flag* na plataforma do **CTFLearn**, conforme a seguir:

 CTFLearn  

Learn Challenges

I'm a dump ✓ 10 points Easy

The keyword is hexadecimal, and removing an useless H.E.H.U.H.E. from the flag.
The flag is in the format CTFLearn{*}

file 

Flag Solved

Forensics · lancillotto 8165 solves