Opening the binary and studying it, we can see that there is the *decrypt\_flag* function. Disassemble it and break just before it prints the break-line.

After it decrypts, print the flag as string. We can use printf and cast the *flag\_buf* variable into *char \** by doing printf "%s", (char \*) flag\_buf.

(gdb) disas decrypt\_flag

...

...

0x0000000000400878 <+242>:

0x0000000000400896 <+272>: mov rdx,QWORD PTR [rip+0x200b4b] # 0x6013e8 <flag\_buf>

0x000000000040089d <+279>: mov eax,DWORD PTR [rbp-0x20]

0x00000000004008a0 <+282>: cdqe

0x00000000004008a2 <+284>: add rax,rdx

0x00000000004008a5 <+287>: mov BYTE PTR [rax],0x0

0x00000000004008a8 <+290>: mov edi,0xa

0x00000000004008ad <+295>: call 0x4005f0 <putchar@plt> ; Prints break-line

...

...

(gdb) b \*0x00000000004008a8

Breakpoint 1 at 0x4008a8

(gdb) r

Starting program: run

Decrypting the Flag into global variable 'flag\_buf'

.....................................

(gdb) printf "%s", (char\*) flag\_buf

picoCTF{gDb\_iS\_sUp3r\_u53fuL\_a6c61d82}

Alternatively, you can use the command x/s flag\_buf to see the content of the variable.

So the flag is: picoCTF{gDb\_iS\_sUp3r\_u53fuL\_a6c61d82}