



Written by Billie

PIE TIME 1 from PicoCTF <https://play.picoctf.org/practice/challenge/490>

PIE TIME

Easy

Binary Exploitation

picoCTF 2025

browser\_webshell\_solvable

AUTHOR: DARKRAICG492

### Description

Can you try to get the flag? Beware we have PIE!

Connect to the program with netcat:

```
$ nc rescued-float.picoctf.net 54799
```

The program's source code can be downloaded [here](#). The binary can be downloaded [here](#).

This challenge launches an instance on demand.

Its current status is: **RUNNING**

Instance Time Remaining: **11:51**


Restart Instance

Hints ?

1

3,001 users solved

95% Liked

 picoCTF{FLAG}

Submit Flag

The shortest writeup I've made so far.

The challenge above will give us a binary that if run will tell us the current address of main.

After disassembling the binary using GDB and running the command "info functions" we get all of the GOT's of all the functions. Here we have the GOT address of win and main

```
0x00000000000012a7 win
0x000000000000133d main
```

After connecting the netcat that is given to us, we get our current main address. Note that:

Our current main address = GOT main address + offset

And to find our current win address we just have to do:  
Our current win address = GOT win address + offset.

With basic algebra we can find our offset which is

Hex value:

$$5ca0e57a233d - 133d = 5CA0E57A1000$$

After adding the offset to our GOT win address we got our current win address

Hex value:

$$5CA0E57A1000 + 12a7 = 5CA0E57A22A7$$

Which will get us the flag

```
Enter the address to jump to, ex => 0x12345: 0x5CA0E57A22A7
Your input: 5ca0e57a22a7
You won!
picoCTF{b4s1c_p051t10n_1nd3p3nd3nc3_a267144a}
```