

## GDB baby step 3



Medium

Reverse Engineering

picoGym Exclusive

x86\_64

AUTHOR: LT 'SYREAL' JONES

Hints 

### Description

Now for something a little different. `0x2262c96b` is loaded into memory in the `main` function. Examine byte-wise the memory that the constant is loaded in by using the GDB command `x/4xb addr`. The flag is the four bytes as they are stored in memory. If you find the bytes `0x11 0x22 0x33 0x44` in the memory location, your flag would be: `picoCTF{0x11223344}`.  
Debug [this](#).

1 2 3 4 5

### Solusi dari write up lain :

Download binary and run gdb with 'gdb debugger0\_c'.  
Dissassemble main() function with 'disassemble main' and you will get the dump:

```
0x0000000000401106 <+0>:  endbr64
0x000000000040110a <+4>:  push  %rbp
0x000000000040110b <+5>:  mov   %rsp,%rbp
0x000000000040110e <+8>:  mov   %edi,-0x14(%rbp)
0x0000000000401111 <+11>: mov   %rsi,-0x20(%rbp)
0x0000000000401115 <+15>: movl  $0x2262c96b,-0x4(%rbp)
0x000000000040111c <+22>: mov   -0x4(%rbp),%eax
0x000000000040111f <+25>: pop   %rbp
0x0000000000401120 <+26>: ret
```

Our memory load is at <+15>, meaning we have to set a breakpoint on next instruction:  
`b *(main+22)`

and run the program with 'run'. Now we can inspect the address of `$rbp-0x4`:  
`x/4xb $rbp-0x4`

which outputs our bytes as they are stored in memory:

```
0x7fffffffdddec: 0x6b 0xc9 0x62 0x22
```

We got our flag:

```
picoCTF{0x6bc96222}
```

### Solusiku :

Buka dengan binary ninja dan langsung pergi ke fungsi main.

```
00401106  int32_t main(int32_t argc, char** argv, char** envp) __pure

00401106  f30f1efa      endbr64
0040110a  55            push    rbp {__saved_rbp}
0040110b  4889e5        mov     rbp, rsp {__saved_rbp}
0040110e  897dec        mov     dword [rbp-0x14 {argc_1}], edi
00401111  488975e0      mov     qword [rbp-0x20 {argv_1}], rsi
00401115  c745fc6bc96222  mov     dword [rbp-0x4 {var_c}], 0x2262c96b
0040111c  8b45fc        mov     eax, dword [rbp-0x4] {0x2262c96b}
0040111f  5d            pop     rbp {__saved_rbp}
00401120  c3            retn    {__return_addr}
```

Gets stack frame offset -0x8 from rbp  
Sets eax to 0x2262c96b  
Opcode: 8b 45 fc

Disini kita bisa melihat bahwa fungsi main mereturn nilai rbp yang menyimpan 0x2262c96b. Karena dalam hints menyinggung endianness, jadi aku coba balik nilainya menjadi 0x6bc96222 sehingga flagnya adalah picoCTF{0x6bc96222}