PicoCTF – Low Level Binary Intro

By zxhry

Intro to Assembly

Bit-O-Asm-1

Bit-O-Asm-1

Medium Reverse Engineering picoGym Exclusive x86_64

AUTHOR: LT 'SYREAL' JONES

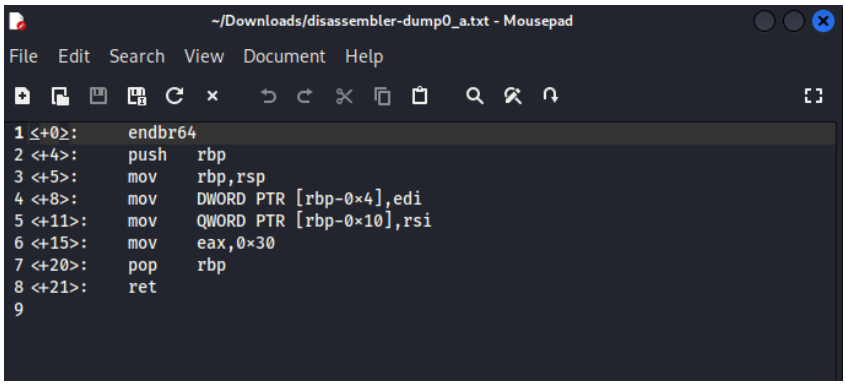
Description

Can you figure out what is in the eax register? Put your answer in the picoCTF flag format: picoCTF{n} where n is the contents of the eax register in the decimal number base. If the answer was 0x11 your flag would be picoCTF{17}.
Download the assembly dump here.

Hints

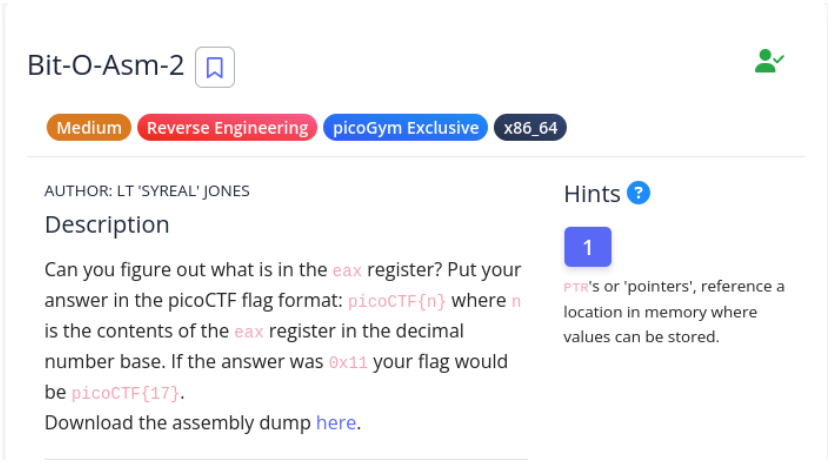
1

Hint: As with most assembly, there is a lot of noise in the instruction dump. Find the one line that pertains to this question and don't second guess yourself!

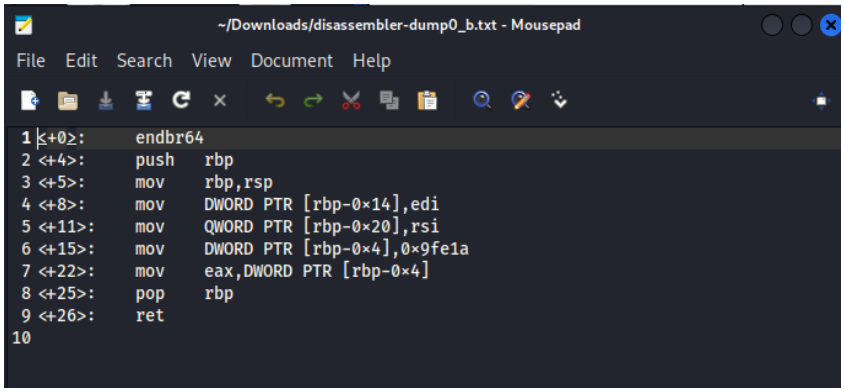


0x30 = 48

Bit-O-Asm-2

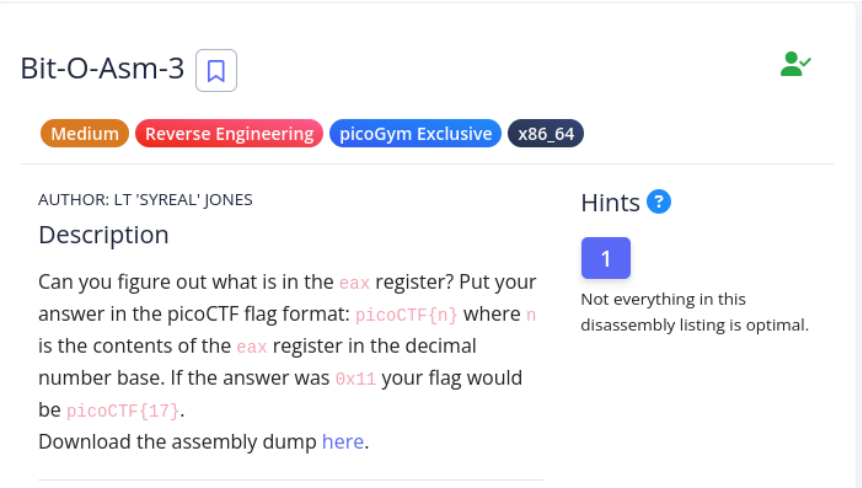


Hint: PTR's or 'pointers', reference a location in memory where values can be stored.

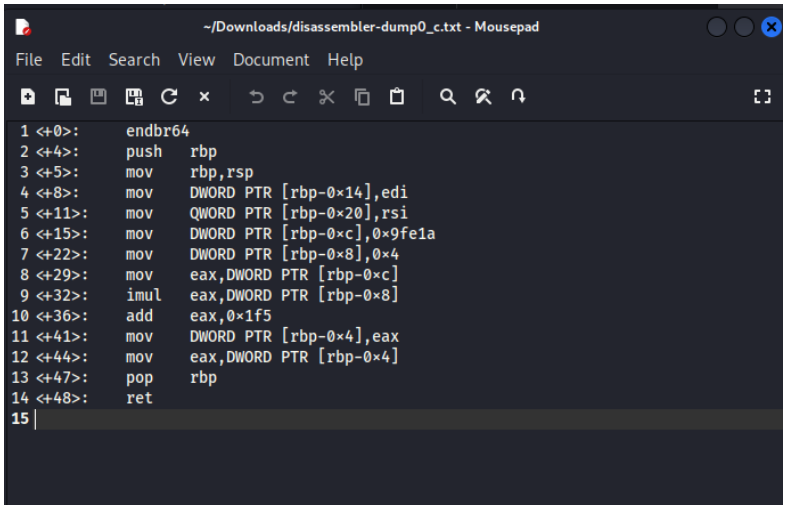


0x9fe1a = 654874

Bit-O-Asm-3



Hint: Not everything in this disassembly listing is optimal.



<+15>: mov DWORD PTR [rbp-0xc],0x9fe1a → 654874

<+22>: mov DWORD PTR [rbp-0x8],0x4 → 4

<+29>: mov eax,DWORD PTR [rbp-0xc] → eax = 654874

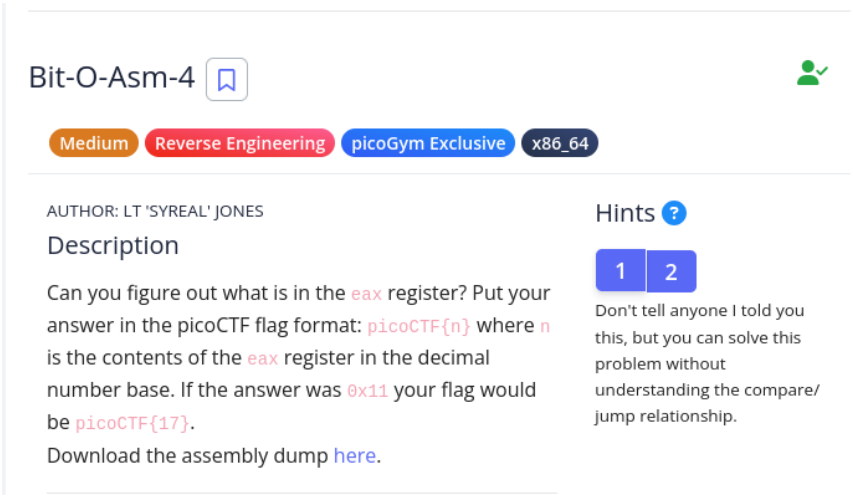
<+32>: imul eax,DWORD PTR [rbp-0x8] → eax = eax * 4 = 2619496

<+36>: add eax,0x1f5 → eax += 501 = 2619997

<+41>: mov DWORD PTR [rbp-0x4],eax → [rbp-0x4] = eax

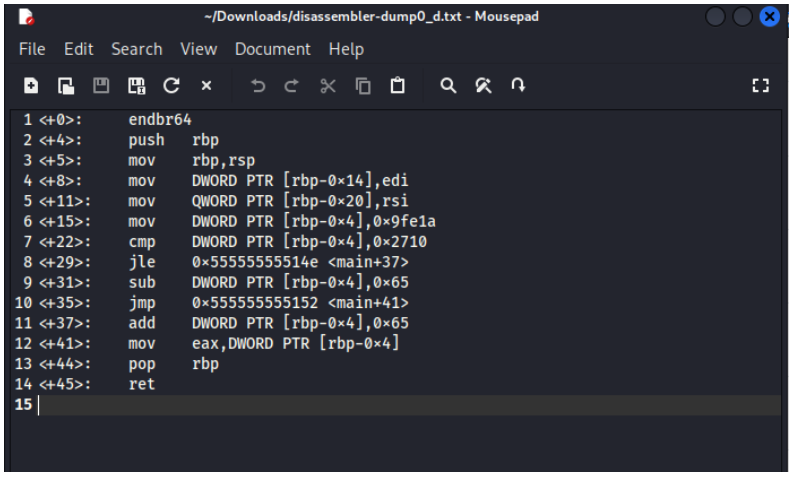
<+44>: mov eax,DWORD PTR [rbp-0x4] → return value in eax, 2619997

Bit-O-Asm-4



Hint 1: Don't tell anyone I told you this, but you can solve this problem without understanding the compare/jump relationship.

Hint 2: Of course, if you're really good, you'll only need one attempt to solve this problem.



<+15>: mov DWORD PTR [rbp-0x4],0x9fe1a → 654874

<+22>: cmp DWORD PTR [rbp-0x4],0x2710 → compare with 10000

<+29>: jle 0x55555555514e <main+37> → if <= 10000, jump to add

<+31>: sub DWORD PTR [rbp-0x4],0x65 → else, subtract 101 = 654773

<+35>: jmp 0x555555555152 <main+41> → skip add

<+37>: add DWORD PTR [rbp-0x4],0x65 → branch target if <= 10000

<+41>: mov eax,DWORD PTR [rbp-0x4] → return value in eax, 654773