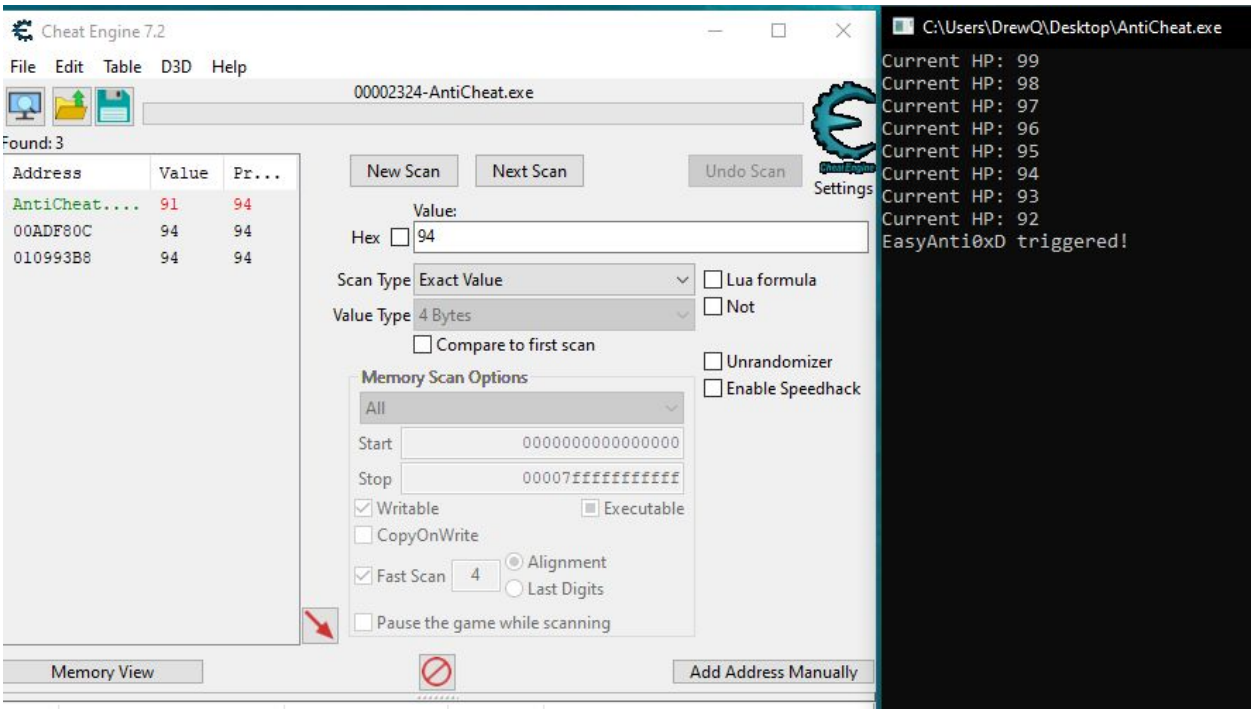
Crackmes 0xD HackMe

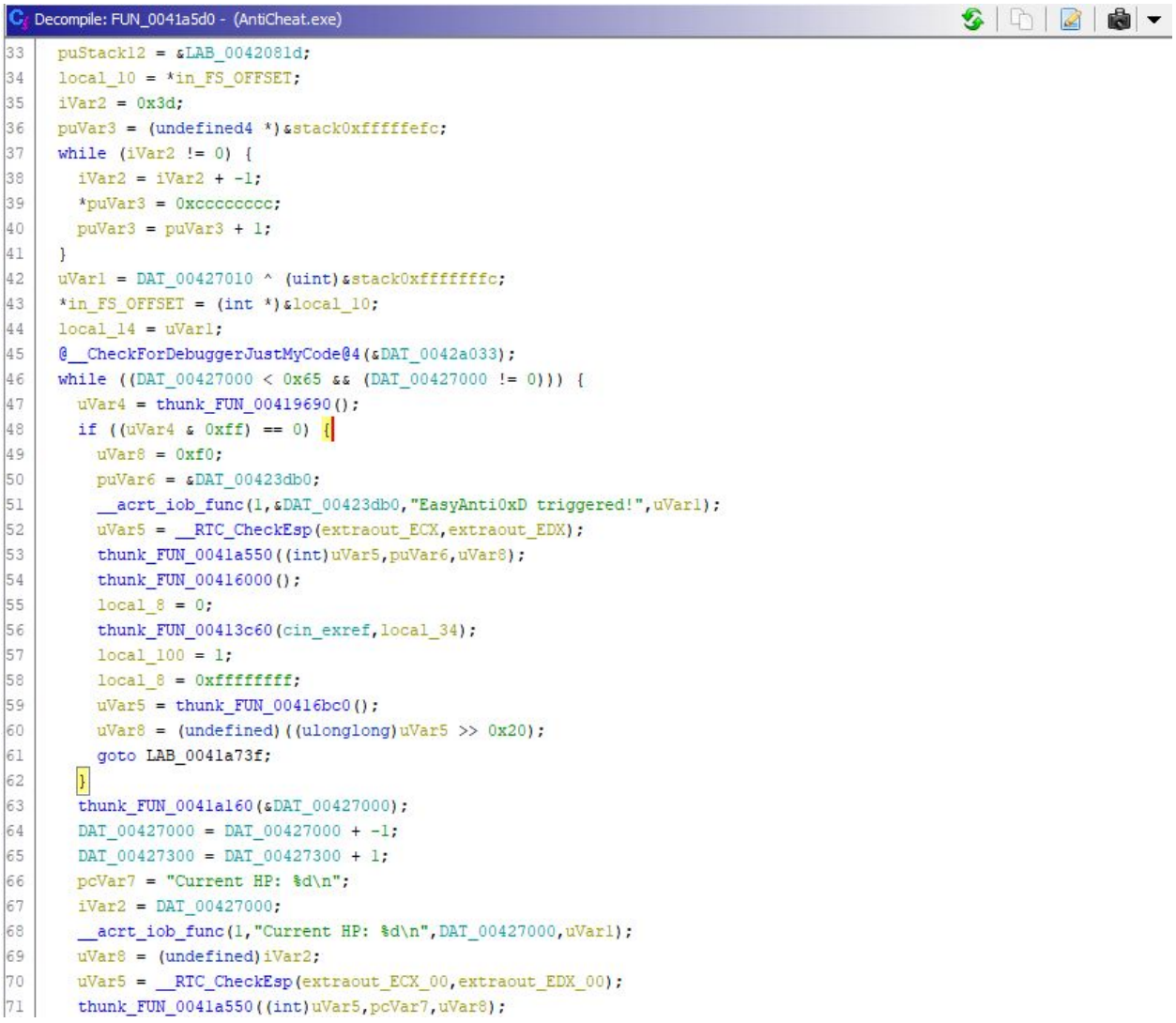
Language: C/C++
Level: Medium (3)
Description: Attempt to change the HP while it is running, without triggering the AntiCheat. If you cheat that is your problem otherwise, congratz!
*I hope this is the right place. src: https://github.com/alxalx14/hackMe/tree/master
Link: https://crackmes.one/crackme/5f38acbf33c5d42a7c667d18



Trying to change the value directly using Cheat Engine causes the AntiCheat to be triggered.

Analyzing The Binary With Ghidra

Searching for the text “Current HP: “ led me to FUN_0041a5d0



Line 46 looks like the main HP loop. It seems to check to see if DAT_00427000 is less than 100 but not equal to 0. 0x65 is 100 when converted to decimal . We can rewrite this line as:

while(HP < 100 && HP != 0)

Line 47 calls the function FUN_00419690 and stores the result into uVar4. If it returns 0 it prints "EasyAnti0xD triggered!". FUN_0041960 must be the anti-cheat function.

We can patch the Exe so the anti-cheat function never gets called.

```
0041a639 0f b6 c0       MOVZX     antiCheatResult,antiCheatResult
0041a63c 85 c0          TEST      antiCheatResult,antiCheatResult
0041a63e 74 2b          JZ        LAB_0041a66b
```

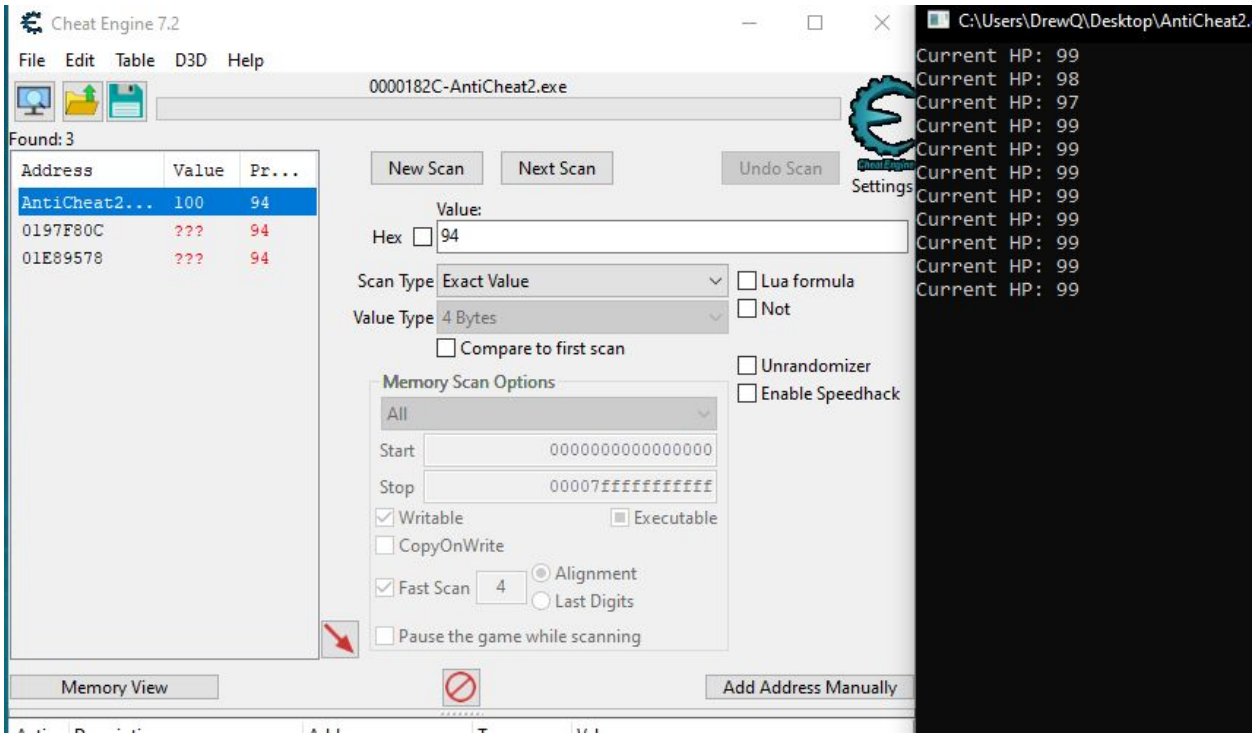
This is the assembly instructions for line 48.

Line 48: if ((antiCheatResult & 0xff) == 0)

If we change the jump instruction to JS instead of JZ everything under the if statement will never be called.

JS means jump if only the previous value is negative. Since the output will never be negative it will never trigger!

```
0041a639 0f b6 c0        MOVZX      antiCheatResult,antiCheatResult
0041a63c 85 c0           TEST       antiCheatResult,antiCheatResult
0041a63e 78 2b           JS         LAB_0041a66b
```



The anti cheat has been avoided!