Crackmes.one Easy - Medium (My first CrackMe)

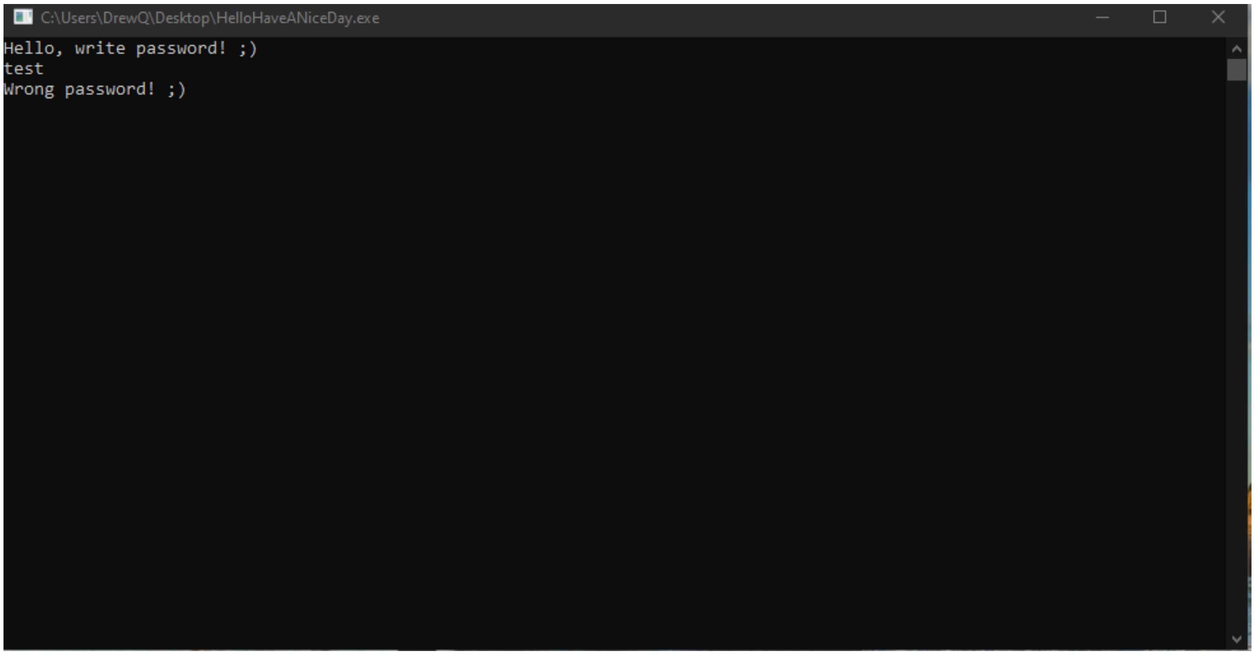
Link:https://crackmes.one/crackme/5ec0207b33c5d449d91ae508

Platform: Windows

Level: 2

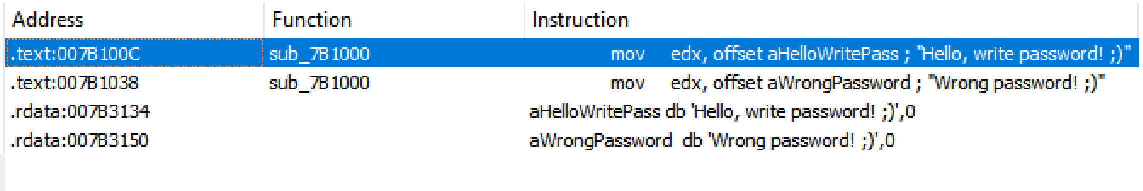
Description: All you have to do - enter password.Good luck, and Have A Nice Day :)

Language : C/C++

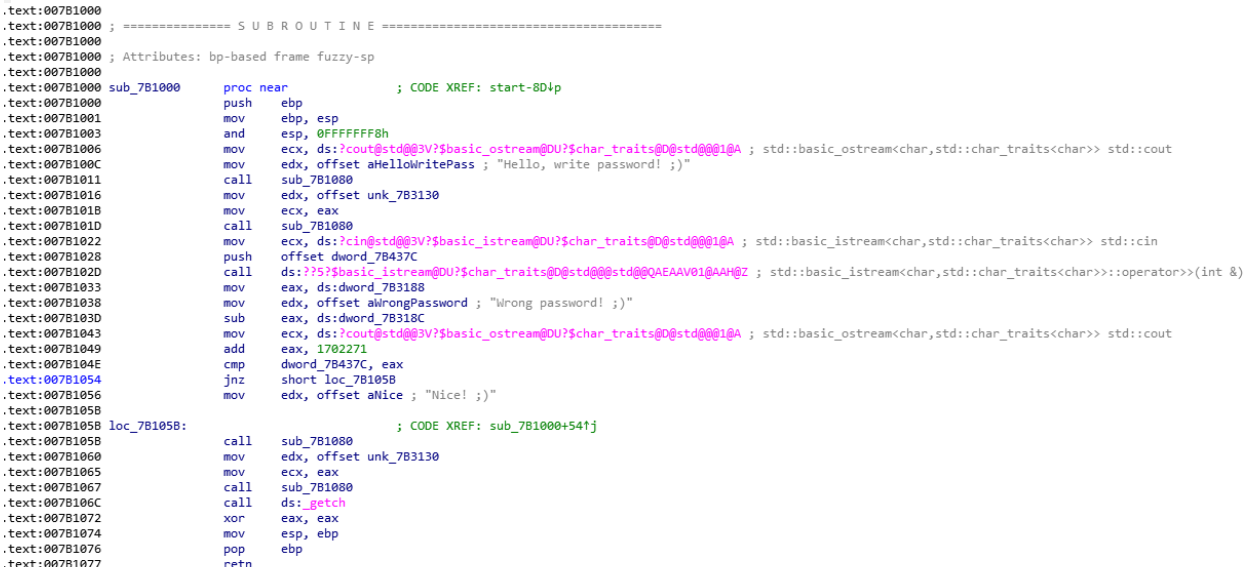

C:\Users\DrewQ\Desktop\HelloHaveANiceDay.exe

```
Hello, write password! ;)
test
Wrong password! ;)
```

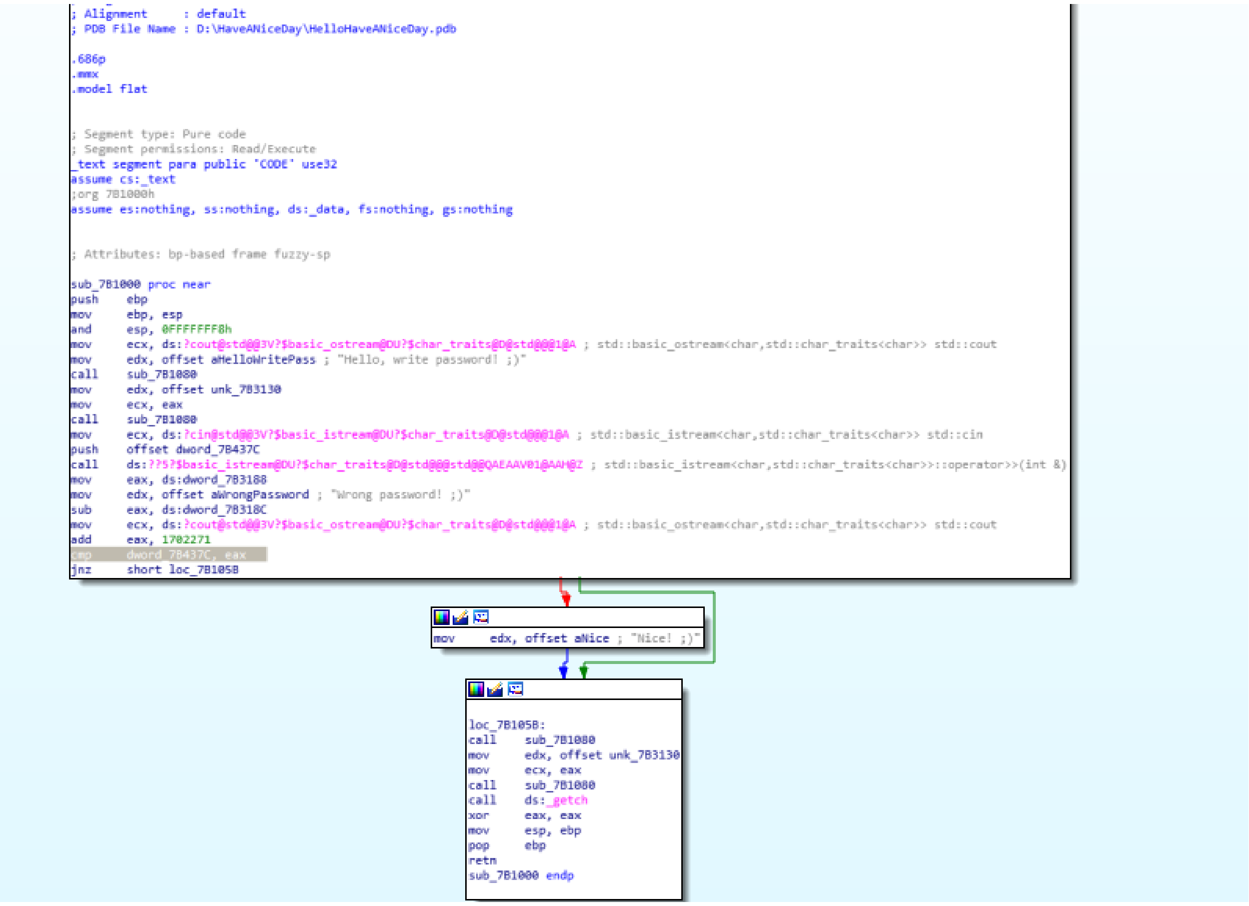
Searching for the string “password” yielded in 2 results both in the same function.



Address	Function	Instruction
.text:007B100C	sub_7B1000	mov edx, offset aHelloWritePass ; "Hello, write password! ;)"
.text:007B1038	sub_7B1000	mov edx, offset aWrongPassword ; "Wrong password! ;)"
.rdata:007B3134		aHelloWritePass db 'Hello, write password! ;)',0
.rdata:007B3150		aWrongPassword db 'Wrong password! ;)',0



```
.text:007B1000
.text:007B1000 ; =============== S U B R O U T I N E =======================================
.text:007B1000
.text:007B1000 ; Attributes: bp-based frame fuzzy-sp
.text:007B1000
.text:007B1000 sub_7B1000      proc near               ; CODE XREF: start-8D↓p
.text:007B1000                 push    ebp
.text:007B1001                 mov     ebp, esp
.text:007B1003                 and     esp, 0FFFFFFF8h
.text:007B1006                 mov     ecx, ds:?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::basic_ostream<char,std::char_traits<char>> std::cout
.text:007B100C                 mov     edx, offset aHelloWritePass ; "Hello, write password! ;)"
.text:007B1011                 call    sub_7B1080
.text:007B1016                 mov     edx, offset unk_7B3130
.text:007B101B                 mov     ecx, eax
.text:007B101D                 call    sub_7B1080
.text:007B1022                 mov     ecx, ds:?cin@std@@3V?$basic_istream@DU?$char_traits@D@std@@@1@A ; std::basic_istream<char,std::char_traits<char>> std::cin
.text:007B1028                 push    offset dword_7B437C
.text:007B102D                 call    ds:??5?$basic_istream@DU?$char_traits@D@std@@@std@@QAEAAV01@AAH@Z ; std::basic_istream<char,std::char_traits<char>>::operator>>(int &)
.text:007B1033                 mov     eax, ds:dword_7B3188
.text:007B1038                 mov     edx, offset aWrongPassword ; "Wrong password! ;)"
.text:007B103D                 sub     eax, ds:dword_7B318C
.text:007B1043                 mov     ecx, ds:?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::basic_ostream<char,std::char_traits<char>> std::cout
.text:007B1049                 add     eax, 1702271
.text:007B104E                 cmp     dword_7B437C, eax
.text:007B1054                 jnz     short loc_7B105B
.text:007B1056                 mov     edx, offset aNice ; "Nice! ;)"
.text:007B105B
.text:007B105B loc_7B105B:                             ; CODE XREF: sub_7B1000+54↑j
.text:007B105B                 call    sub_7B1080
.text:007B1060                 mov     edx, offset unk_7B3130
.text:007B1065                 mov     ecx, eax
.text:007B1067                 call    sub_7B1080
.text:007B106C                 call    ds:_getch
.text:007B1072                 xor     eax, eax
.text:007B1074                 mov     esp, ebp
.text:007B1076                 pop     ebp
.text:007B1077                 retn
```

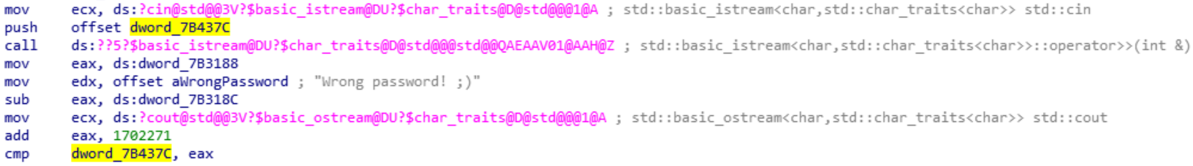


```
; Alignment     : default
; PDB File Name : D:\HaveANiceDay\HelloHaveANiceDay.pdb

.686p
.mmx
.model flat


; Segment type: Pure code
; Segment permissions: Read/Execute
_text segment para public 'CODE' use32
assume cs:_text
;org 7B1000h
assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing


; Attributes: bp-based frame fuzzy-sp

sub_7B1000 proc near
push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
mov     ecx, ds:?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::basic_ostream<char,std::char_traits<char>> std::cout
mov     edx, offset aHelloWritePass ; "Hello, write password! ;)"
call    sub_7B1080
mov     edx, offset unk_7B3130
mov     ecx, eax
call    sub_7B1080
mov     ecx, ds:?cin@std@@3V?$basic_istream@DU?$char_traits@D@std@@@1@A ; std::basic_istream<char,std::char_traits<char>> std::cin
push    offset dword_7B437C
call    ds:??5?$basic_istream@DU?$char_traits@D@std@@@std@@QAEAAV01@AAH@Z ; std::basic_istream<char,std::char_traits<char>>::operator>>(int &)
mov     eax, ds:dword_7B3188
mov     edx, offset aWrongPassword ; "Wrong password! ;)"
sub     eax, ds:dword_7B318C
mov     ecx, ds:?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::basic_ostream<char,std::char_traits<char>> std::cout
add     eax, 1702271
cmp     dword_7B437C, eax
jnz     short loc_7B105B
```

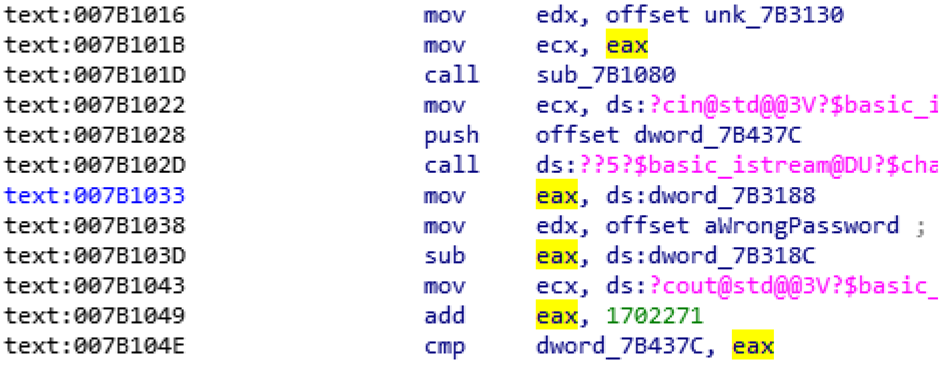
```
mov     edx, offset aNice ; "Nice! ;)"
```

```
loc_7B105B:
call    sub_7B1080
mov     edx, offset unk_7B3130
mov     ecx, eax
call    sub_7B1080
call    ds:_getch
xor     eax, eax
mov     esp, ebp
pop     ebp
retn
sub_7B1000 endp
```

At 0x007B103D edx is set to “Wrong password ;)” and at 0x007B104E dword_7B437C is being compared to eax. If this returns true edx is set to “Nice! ;)” else it remains “Wrong password ;)” dword_7B437C is the user's input. We also know the program is looking for an integer.

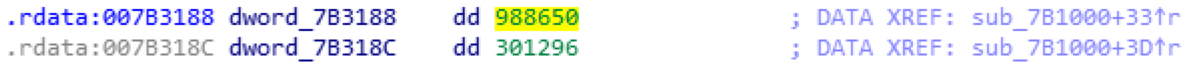


```
mov     ecx, ds:?cin@std@@3V?$basic_istream@DU?$char_traits@D@std@@@1@A ; std::basic_istream<char,std::char_traits<char>> std::cin
push    offset dword_7B437C
call    ds:??5?$basic_istream@DU?$char_traits@D@std@@@std@@QAEAAV01@AAH@Z ; std::basic_istream<char,std::char_traits<char>>::operator>>(int &)
mov     eax, ds:dword_7B3188
mov     edx, offset aWrongPassword ; "Wrong password! ;)"
sub     eax, ds:dword_7B318C
mov     ecx, ds:?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::basic_ostream<char,std::char_traits<char>> std::cout
add     eax, 1702271
cmp     dword_7B437C, eax
```



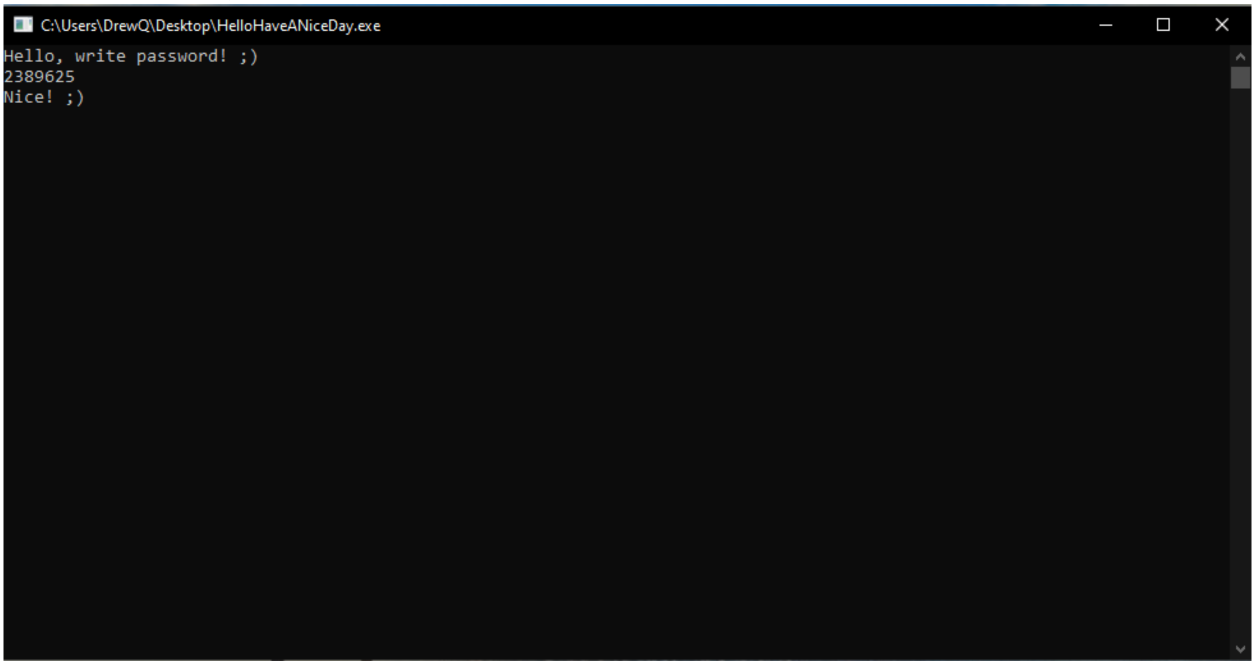
```
text:007B1016                 mov     edx, offset unk_7B3130
text:007B101B                 mov     ecx, eax
text:007B101D                 call    sub_7B1080
text:007B1022                 mov     ecx, ds:?cin@std@@3V?$basic_i
text:007B1028                 push    offset dword_7B437C
text:007B102D                 call    ds:??5?$basic_istream@DU?$cha
text:007B1033                 mov     eax, ds:dword_7B3188
text:007B1038                 mov     edx, offset aWrongPassword ;
text:007B103D                 sub     eax, ds:dword_7B318C
text:007B1043                 mov     ecx, ds:?cout@std@@3V?$basic_
text:007B1049                 add     eax, 1702271
text:007B104E                 cmp     dword_7B437C, eax
```

The value of eax is getting set to the value of dword_7B3188. Then eax is subtracted by dword_7B318C. Finally 1702271 is added to eax.



```
.rdata:007B3188 dword_7B3188    dd 988650               ; DATA XREF: sub_7B1000+33↑r
.rdata:007B318C dword_7B318C    dd 301296               ; DATA XREF: sub_7B1000+3D↑r
```

(988650 - 301296) + 1702271 = 2389625



```
Hello, write password! ;)
2389625
Nice! ;)
```