Challenge: Hunting License

Challenge Description :

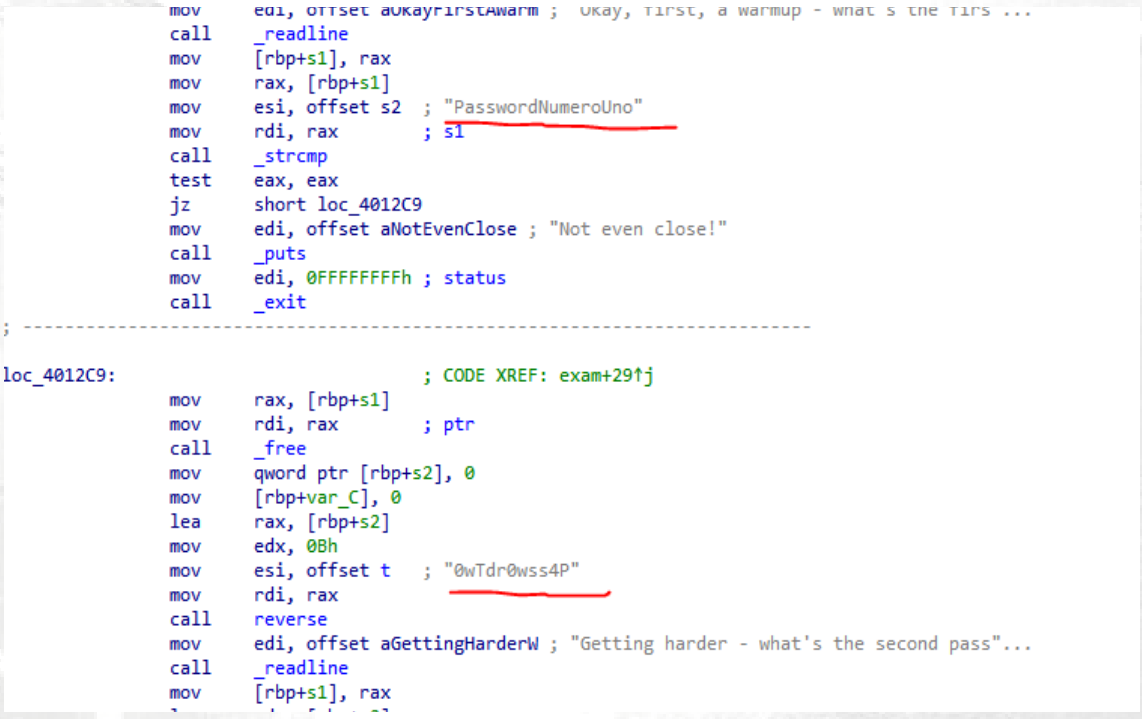
STOP! Adventurer, have you got an up to date relic hunting license? If you don't, you'll need to take the exam again before you'll be allowed passage into the spacelanes!

Context :

- Analyze a Compiled File, to get three passwords hidden within itself, one is in the open, one is in reverse and one has XOR encryption.
- After gaining the password you need to answer a small questionnaire with the password requested at the end to gain the flag.

Flag :

- First Install the files and start decompiling the File. The first two passwords should be really easy to get as it's really noticeable. Depending on the tool you use.



```
                mov     edi, offset aOkayFirstAwarm ; "Okay, first, a warmup - what's the firs ...
                call    _readline
                mov     [rbp+s1], rax
                mov     rax, [rbp+s1]
                mov     esi, offset s2  ; "PasswordNumeroUno"
                mov     rdi, rax        ; s1
                call    _strcmp
                test    eax, eax
                jz      short loc_4012C9
                mov     edi, offset aNotEvenClose ; "Not even close!"
                call    _puts
                mov     edi, 0FFFFFFFFh ; status
                call    _exit
; ---------------------------------------------------------------------------

loc_4012C9:                             ; CODE XREF: exam+29↑j
                mov     rax, [rbp+s1]
                mov     rdi, rax        ; ptr
                call    _free
                mov     qword ptr [rbp+s2], 0
                mov     [rbp+var_C], 0
                lea     rax, [rbp+s2]
                mov     edx, 0Bh
                mov     esi, offset t   ; "0wTdr0wss4P"
                mov     rdi, rax
                call    reverse
                mov     edi, offset aGettingHarderW ; "Getting harder - what's the second pass"...
                call    _readline
                mov     [rbp+s1], rax
```

- The first flag is correct by default, for the second one you will need to reverse the string so it is spelt correctly.

  [PasswordNumeroUno]

  [P4ssw0rdTw0]

- The Third flag is a bit harder to get, its under a name called "T2"

```
                public t
t               db '0wTdr0wss4P',0      ; DATA XREF: exam+63↑o
                align 10h
                public t2
t2              db  47h ; G             ; DATA XREF: exam+D7↑o
                db  7Bh ; {
                db  7Ah ; z
                db  61h ; a
                db  77h ; w
                db  52h ; R
                db  7Dh ; }
                db  77h ; w
                db  55h ; U
                db  7Ah ; z
                db  7Dh ; }
                db  72h ; r
                db  7Fh ;
                db  32h ; 2
                db  32h ; 2
                db  32h ; 2
                db  13h
_data           ends
```

- Reading some of the Decompiled code and its binary suggested that XOR is being used to encrypting the string in "T2" with the Key of just "19"

[0x47 0x7b 0x7a 0x61 0x77 0x52 0x7d 0x77 0x55 0x7a 0x7d 0x72 0x7f 0x32 0x32 0x32 0x13]

- Deciphering this will give the password needed to complete the questionnaire. By applying the XOR operation with the key 19 to each number in the list, we translate them into a different set of numbers. The transformed numbers represent letters of the alphabet.

- We convert these numbers back into letters. After decoding each number using the key 19, the outcome should reveal the password "ThirdandFinal!!!".

- Here is the python code I used to decode this.

```
Make sure to Add in the Right Array to decode the actual Password

array = [ 0x7a ,0x7d ,0x72, 0x7f ,0x32, 0x32 ,0x32, 0x13]

size_array = 17

actual_password = bytearray(17)

for i in range (17):

        actual_password[i] = array[i] ^ 19

      print(actual_password)
```

- To get the Flag for this one you will need to start up the Instance and Netcat into the IP:PORT, A TUI should pop-up as a questionnaire, asking questions about the file, these should be straight-foward.

```
What is the file format of the executable?
> ELF
[+] Correct!

What is the CPU architecture of the executable?
> X86-64
[+] Correct!

What library is used to read lines for user answers? (`ldd` may help)
> libreadline.so.8
[+] Correct!

What is the address of the `main` function?
> 0x401172
[+] Correct!

How many calls to `puts` are there in `main`? (using a decompiler may help)
> 5
[+] Correct!

What is the first password?
> PasswordNumberUno
[-] Wrong Answer.
What is the first password?
> PasswordNumeroUno
[+] Correct!

What is the reversed form of the second password?
> 0wTdr0wss4P
[+] Correct!

What is the real second password?
> P4ssw0rdTw0
[+] Correct!

What is the XOR key used to encode the third password?
> 19
[+] Correct!

What is the third password?
> ThirdAndFinal!!!
[+] Correct!

[+] Here is the flag: `HTB{l1c3ns3_4cquir3d-hunt1ng_t1m3!}`
```

- Finishing the Questionnaire should provide you with a flag.
    - HTB{l1c3ns3_4cquir3d-hunt1ng_t1m3!}