JBU CTF

-Misc & Reversing-

이다영

[Misc]

- 누군가 정답을 알고 있어

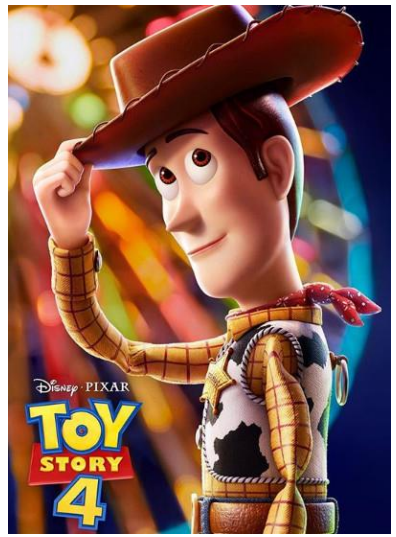
[Reversing]

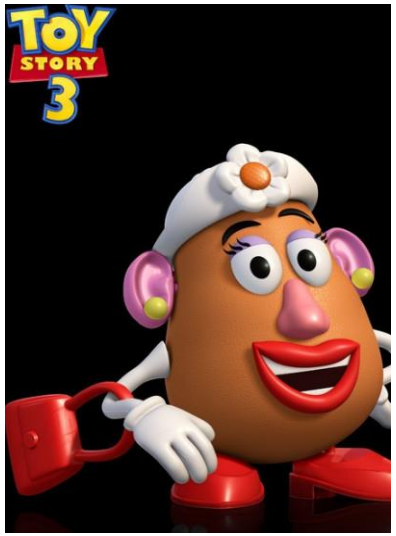
- Assembly

- ID & PW

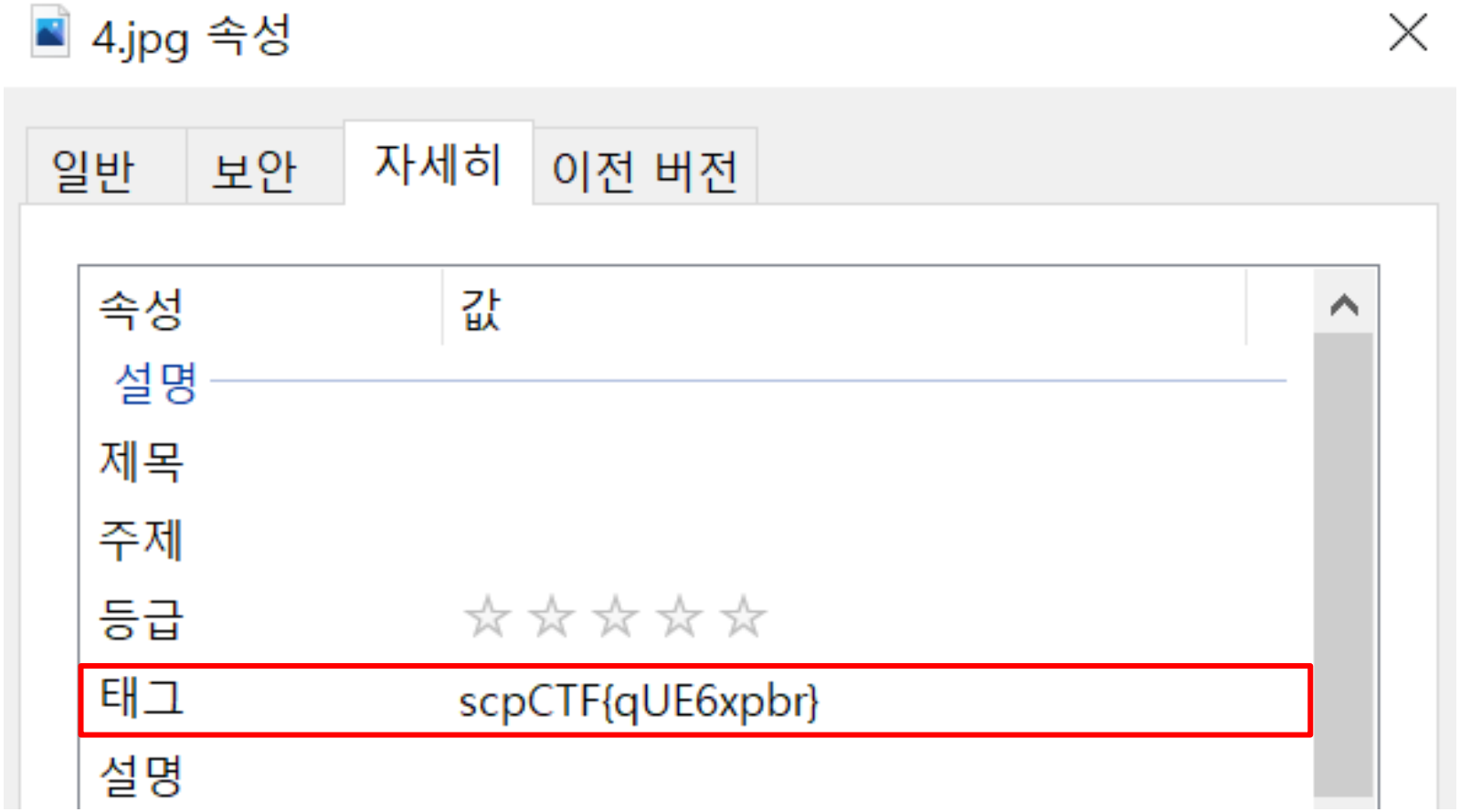
누군가 정답을 알고 있어

▷ 토이스토리 6명의 친구들 중 한 친구만
이 플래그를 가지고 있습니다! 찾아봅시다!





누군가 정답을 알고 있어



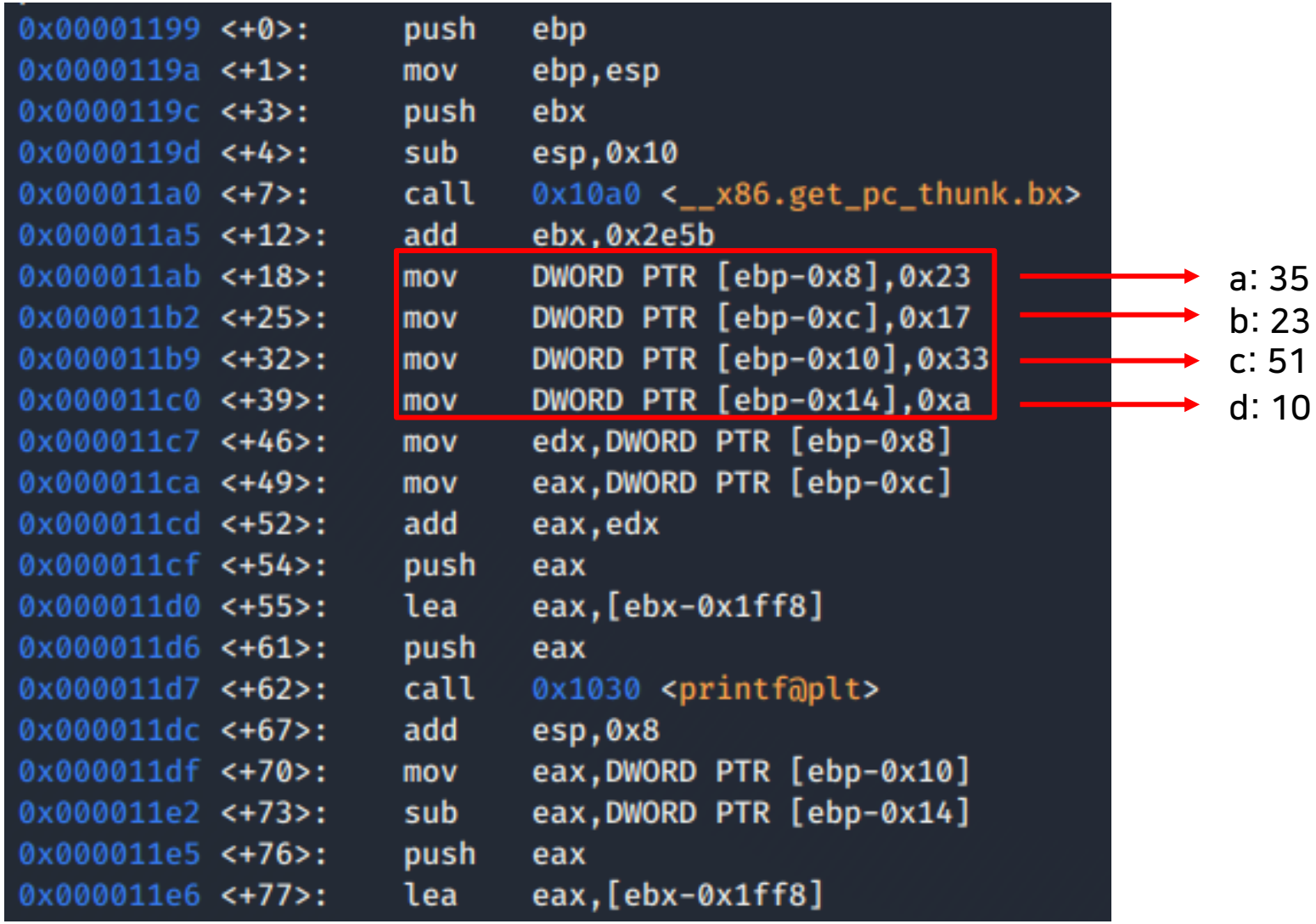
Assembly

▷ 플래그는 다음 어셈블리 코드에서 출력되는 두 값입니다!

Ex) 25, 10 → scpCTF{2510}

```
0x00001199 <+0>:     push   ebp
0x0000119a <+1>:     mov    ebp,esp
0x0000119c <+3>:     push   ebx
0x0000119d <+4>:     sub    esp,0x10
0x000011a0 <+7>:     call   0x10a0 <__x86.get_pc_thunk.bx>
0x000011a5 <+12>:    add    ebx,0x2e5b
0x000011ab <+18>:    mov    DWORD PTR [ebp-0x8],0x23
0x000011b2 <+25>:    mov    DWORD PTR [ebp-0xc],0x17
0x000011b9 <+32>:    mov    DWORD PTR [ebp-0x10],0x33
0x000011c0 <+39>:    mov    DWORD PTR [ebp-0x14],0xa
0x000011c7 <+46>:    mov    edx,DWORD PTR [ebp-0x8]
0x000011ca <+49>:    mov    eax,DWORD PTR [ebp-0xc]
0x000011cd <+52>:    add    eax,edx
0x000011cf <+54>:    push   eax
0x000011d0 <+55>:    lea    eax,[ebx-0x1ff8]
0x000011d6 <+61>:    push   eax
0x000011d7 <+62>:    call   0x1030 <printf@plt>
0x000011dc <+67>:    add    esp,0x8
0x000011df <+70>:    mov    eax,DWORD PTR [ebp-0x10]
0x000011e2 <+73>:    sub    eax,DWORD PTR [ebp-0x14]
0x000011e5 <+76>:    push   eax
0x000011e6 <+77>:    lea    eax,[ebx-0x1ff8]
```

Assembly


0x00001199 <+0>: push ebp
0x0000119a <+1>: mov ebp,esp
0x0000119c <+3>: push ebx
0x0000119d <+4>: sub esp,0x10
0x000011a0 <+7>: call 0x10a0 <__x86.get_pc_thunk.bx>
0x000011a5 <+12>: add ebx,0x2e5b
0x000011ab <+18>: mov DWORD PTR [ebp-0x8],0x23
0x000011b2 <+25>: mov DWORD PTR [ebp-0xc],0x17
0x000011b9 <+32>: mov DWORD PTR [ebp-0x10],0x33
0x000011c0 <+39>: mov DWORD PTR [ebp-0x14],0xa
0x000011c7 <+46>: mov edx,DWORD PTR [ebp-0x8]
0x000011ca <+49>: mov eax,DWORD PTR [ebp-0xc]
0x000011cd <+52>: add eax,edx
0x000011cf <+54>: push eax
0x000011d0 <+55>: lea eax,[ebx-0x1ff8]
0x000011d6 <+61>: push eax
0x000011d7 <+62>: call 0x1030 <printf@plt>
0x000011dc <+67>: add esp,0x8
0x000011df <+70>: mov eax,DWORD PTR [ebp-0x10]
0x000011e2 <+73>: sub eax,DWORD PTR [ebp-0x14]
0x000011e5 <+76>: push eax
0x000011e6 <+77>: lea eax,[ebx-0x1ff8]
a: 35
b: 23
c: 51
d: 10

Assembly

```
0x00001199 <+0>:     push   ebp
0x0000119a <+1>:     mov    ebp,esp
0x0000119c <+3>:     push   ebx
0x0000119d <+4>:     sub    esp,0x10
0x000011a0 <+7>:     call   0x10a0 <__x86.get_pc_thunk.bx>
0x000011a5 <+12>:    add    ebx,0x2e5b
0x000011ab <+18>:    mov    DWORD PTR [ebp-0x8],0x23
0x000011b2 <+25>:    mov    DWORD PTR [ebp-0xc],0x17
0x000011b9 <+32>:    mov    DWORD PTR [ebp-0x10],0x33
0x000011c0 <+39>:    mov    DWORD PTR [ebp-0x14],0xa
0x000011c7 <+46>:    mov    edx,DWORD PTR [ebp-0x8]
0x000011ca <+49>:    mov    eax,DWORD PTR [ebp-0xc]
0x000011cd <+52>:    add    eax,edx
0x000011cf <+54>:    push   eax
0x000011d0 <+55>:    lea    eax,[ebx-0x1ff8]
0x000011d6 <+61>:    push   eax
0x000011d7 <+62>:    call   0x1030 <printf@plt>
0x000011dc <+67>:    add    esp,0x8
0x000011df <+70>:    mov    eax,DWORD PTR [ebp-0x10]
0x000011e2 <+73>:    sub    eax,DWORD PTR [ebp-0x14]
0x000011e5 <+76>:    push   eax
0x000011e6 <+77>:    lea    eax,[ebx-0x1ff8]
```

35 + 23 = 58

Assembly

```
0x00001199 <+0>:     push   ebp
0x0000119a <+1>:     mov    ebp,esp
0x0000119c <+3>:     push   ebx
0x0000119d <+4>:     sub    esp,0x10
0x000011a0 <+7>:     call   0x10a0 <__x86.get_pc_thunk.bx>
0x000011a5 <+12>:    add    ebx,0x2e5b
0x000011ab <+18>:    mov    DWORD PTR [ebp-0x8],0x23
0x000011b2 <+25>:    mov    DWORD PTR [ebp-0xc],0x17
0x000011b9 <+32>:    mov    DWORD PTR [ebp-0x10],0x33
0x000011c0 <+39>:    mov    DWORD PTR [ebp-0x14],0xa
0x000011c7 <+46>:    mov    edx,DWORD PTR [ebp-0x8]
0x000011ca <+49>:    mov    eax,DWORD PTR [ebp-0xc]
0x000011cd <+52>:    add    eax,edx
0x000011cf <+54>:    push   eax
0x000011d0 <+55>:    lea    eax,[ebx-0x1ff8]
0x000011d6 <+61>:    push   eax
0x000011d7 <+62>:    call   0x1030 <printf@plt>
0x000011dc <+67>:    add    esp,0x8
0x000011df <+70>:    mov    eax,DWORD PTR [ebp-0x10]
0x000011e2 <+73>:    sub    eax,DWORD PTR [ebp-0x14]
0x000011e5 <+76>:    push   eax
0x000011e6 <+77>:    lea    eax,[ebx-0x1ff8]
```

51 - 10 = 41

Assembly


0x00001199 <+0>: push ebp
0x0000119a <+1>: mov ebp,esp
0x0000119c <+3>: push ebx
0x0000119d <+4>: sub esp,0x10
0x000011a0 <+7>: call 0x10a0 <__x86.get_pc_thunk.bx>
0x000011a5 <+12>: add ebx,0x2e5b
0x000011ab <+18>: mov DWORD PTR [ebp-0x8],0x23
0x000011b2 <+25>: mov DWORD PTR [ebp-0xc],0x17
scpCTF{5841}
0x000011d0 <+55>: lea eax,[ebx-0x1ff8]
0x000011d6 <+61>: push eax
0x000011d7 <+62>: call 0x1030 <printf@plt>
0x000011dc <+67>: add esp,0x8
0x000011df <+70>: mov eax,DWORD PTR [ebp-0x10]
0x000011e2 <+73>: sub eax,DWORD PTR [ebp-0x14]
0x000011e5 <+76>: push eax
0x000011e6 <+77>: lea eax,[ebx-0x1ff8]

ID & PW

▷ 아이디와 패스워드가 암호화되어 프로그램이 실행되지 않습니다. 프로그램을 분석하여 암호화된 값을 확인하고, 정상적인 아이디와 패스워드를 알아내세요! main 함수의 시작 주소도 함께 찾아봅시다!

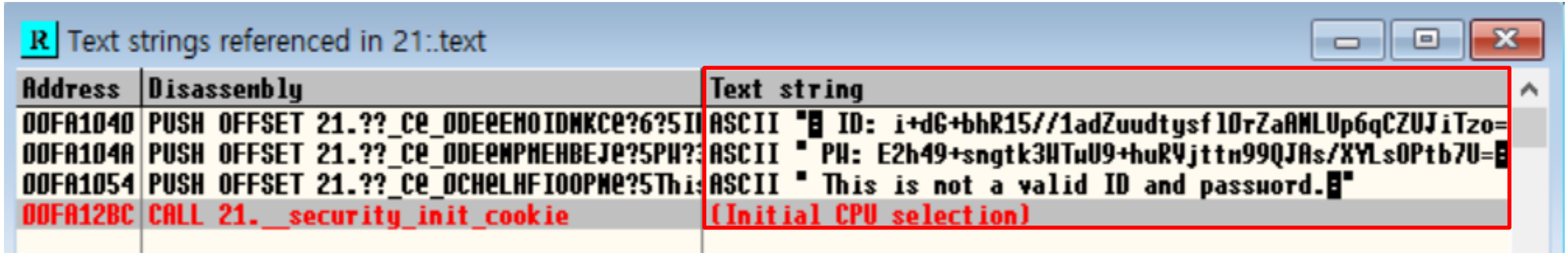
Ex) ID: moonlight, PW: Garden, main 함수 시작 주소: 00771081
→ scpCTF{moonlightGarden00771081}

```
ID: i+dG+bhR15//1adZuudtysfIOrZaANLUp6qCZUJiTzo=
PW: E2h49+sngtk3WTwU9+huRVjttm99QJAs/XYLsOPtb7U=

This is not a valid ID and password.
```

```
HWND hWndConsole = GetConsoleWindow();
ShowWindow(hWndConsole, SW_HIDE);
```

ID & PW


Text strings referenced in 21:.text
Address
Disassembly
Text string
00FA1040 PUSH OFFSET 21.??_C@_0DE@EMOIDMKC@?6?5I ASCII "▯ ID: i+dG+bhR15//1adZuudtysf1OrZaAMLUp6qCZUJiTzo=
00FA104A PUSH OFFSET 21.??_C@_0DE@MPMEHBEJ@?5PW?3 ASCII " PW: E2h49+sngtk3WTuU9+huRVjttn99QJAs/XYLsOPtb7U=▯
00FA1054 PUSH OFFSET 21.??_C@_0CH@LHFIOOPM@?5This ASCII " This is not a valid ID and passuord.▯"
00FA12BC CALL 21.__security_init_cookie (Initial CPU selection)

ID & PW

Decryption

Encrypted Text

E2h49+snqtk3WTwU9+huRVittm99QJAs/XYLsOPtb7U=

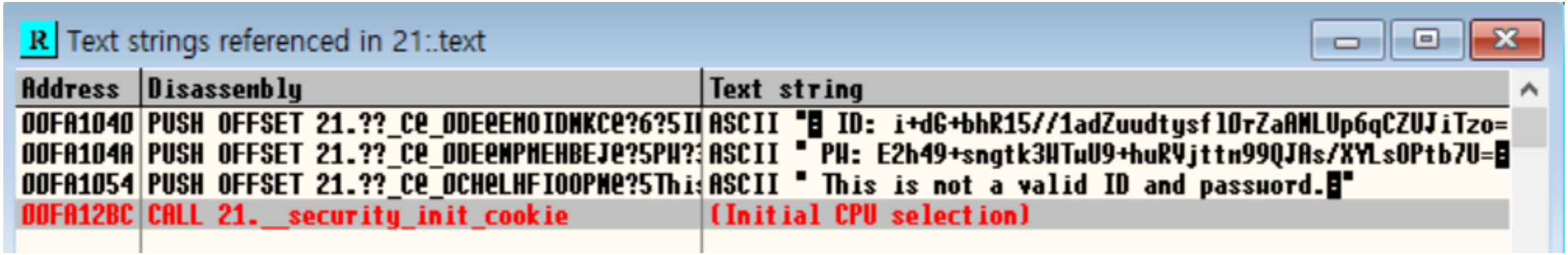
Decrypt

Decrypted Text

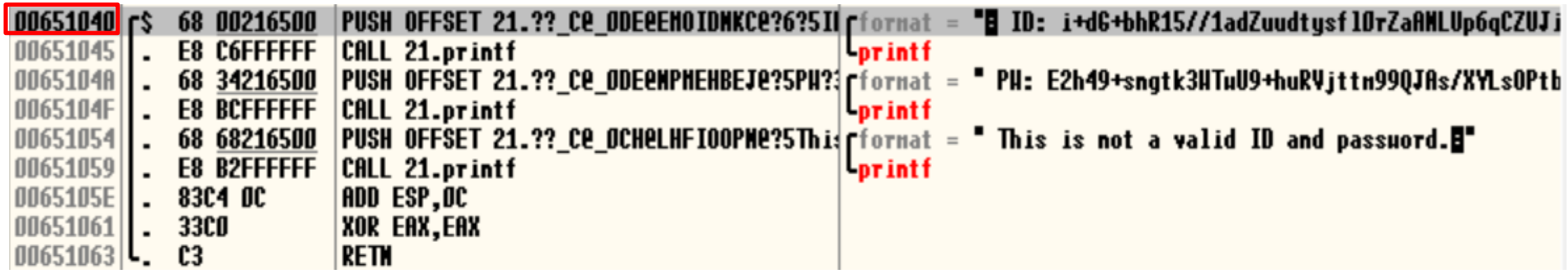
juice

ID & PW

main 함수 시작 주소 찾기 ①: 참조되는 문자열 확인

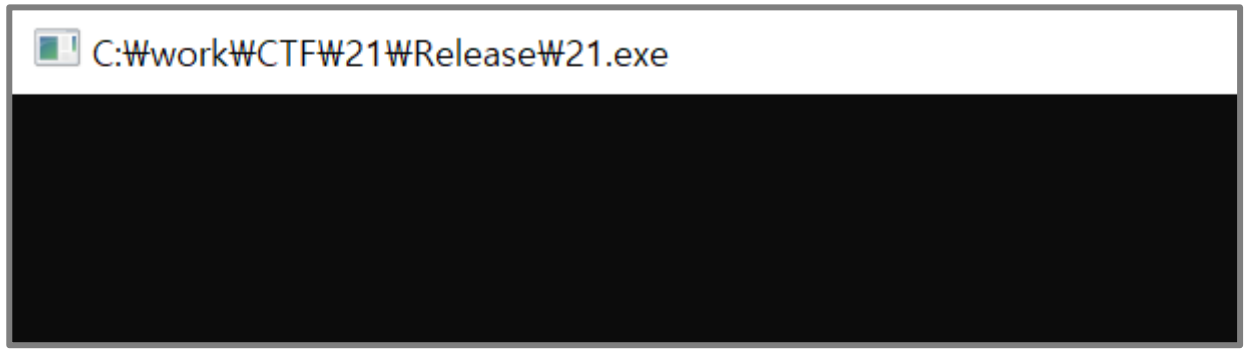


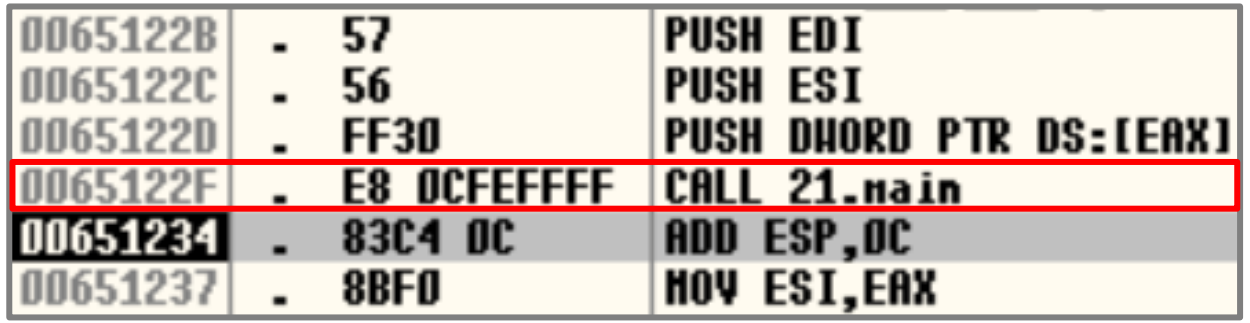
ID & PW

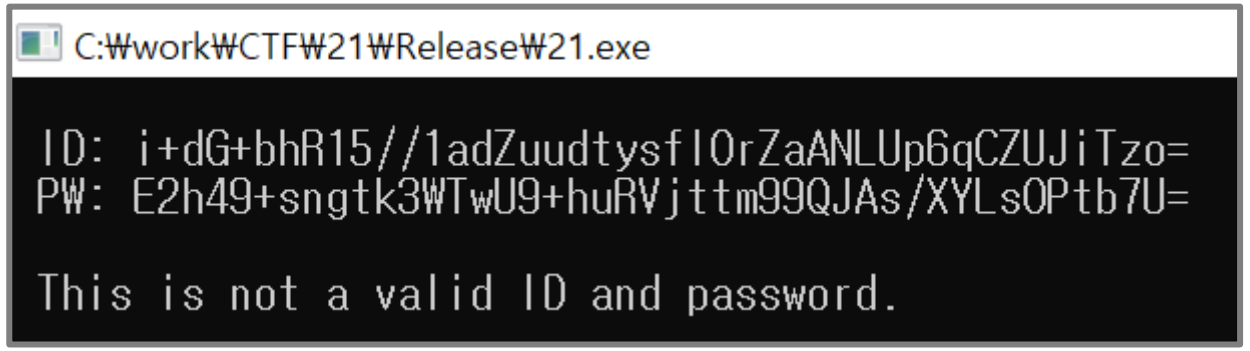

00651040 $ 68 00216500 PUSH OFFSET 21.??_C@_0DE@EMOIDMKC@?6?5II format = "▪ ID: i+dG+bhR15//1adZuudtysf1OrZaAMLUp6qCZUJi
00651045 . E8 C6FFFFFF CALL 21.printf printf
0065104A . 68 34216500 PUSH OFFSET 21.??_C@_0DE@MPMEHBEJ@?5PW?3 format = " PW: E2h49+sngtk3WTuU9+huRYjttn99QJAs/XYLsOPtb
0065104F . E8 BCFFFFFF CALL 21.printf printf
00651054 . 68 68216500 PUSH OFFSET 21.??_C@_0CH@LHFIOOPM@?5This format = " This is not a valid ID and passuord.▪"
00651059 . E8 B2FFFFFF CALL 21.printf printf
0065105E . 83C4 0C ADD ESP,0C
00651061 . 33C0 XOR EAX,EAX
00651063 . C3 RETN

ID & PW

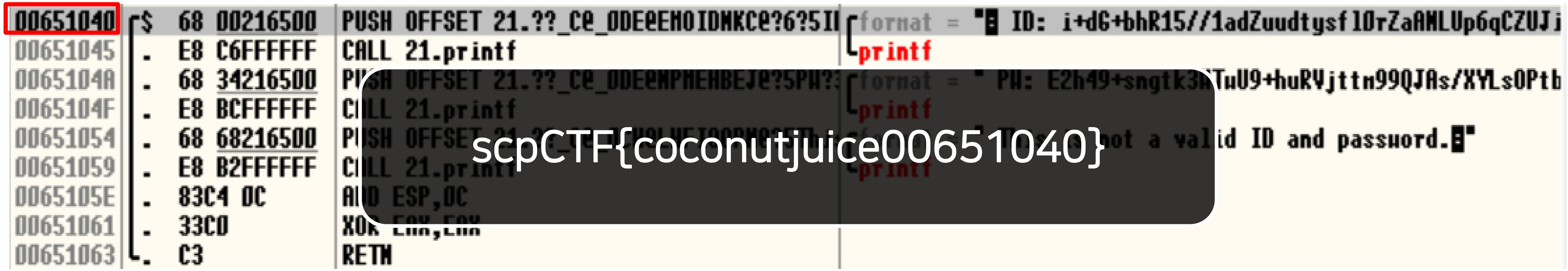
main 함수 시작 주소 찾기 ②
: 한 줄씩 실행







ID & PW


00651040 $ 68 00216500 PUSH OFFSET 21.??_C@_0DE@EHOIDNKC@?6?5I format = " ID: i+dG+bhR15//1adZuudtysf10rZaANLUp6qCZUJi
00651045 . E8 C6FFFFFF CALL 21.printf printf
0065104A . 68 34216500
0065104F . E8 BCFFFFFF
00651054 . 68 68216500
00651059 . E8 B2FFFFFF
0065105E . 83C4 0C
00651061 . 33C0
00651063 . C3 RETN
scpCTF{coconutjuice00651040}
id ID and password.

감사합니다 ☺