JBU CTF

-Misc & Reversing-

이다영

[Misc]

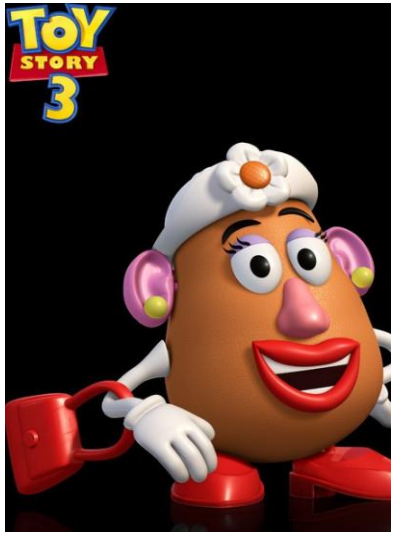
- 누군가 정답을 알고 있어

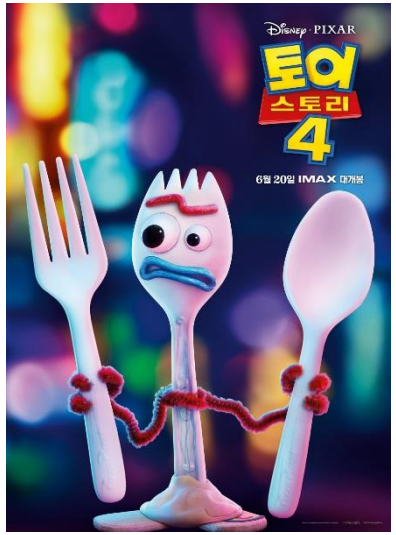
[Reversing]

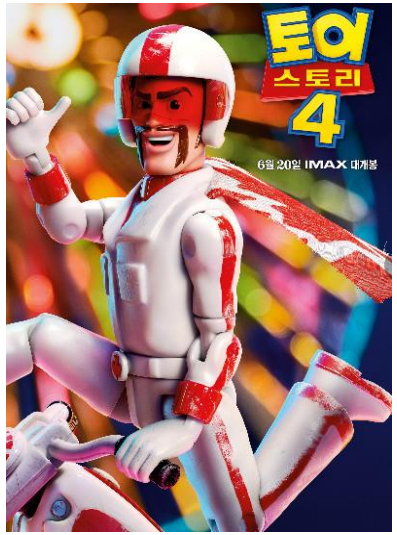
- Assembly
- ID & PW

누군가 정답을 알고 있어

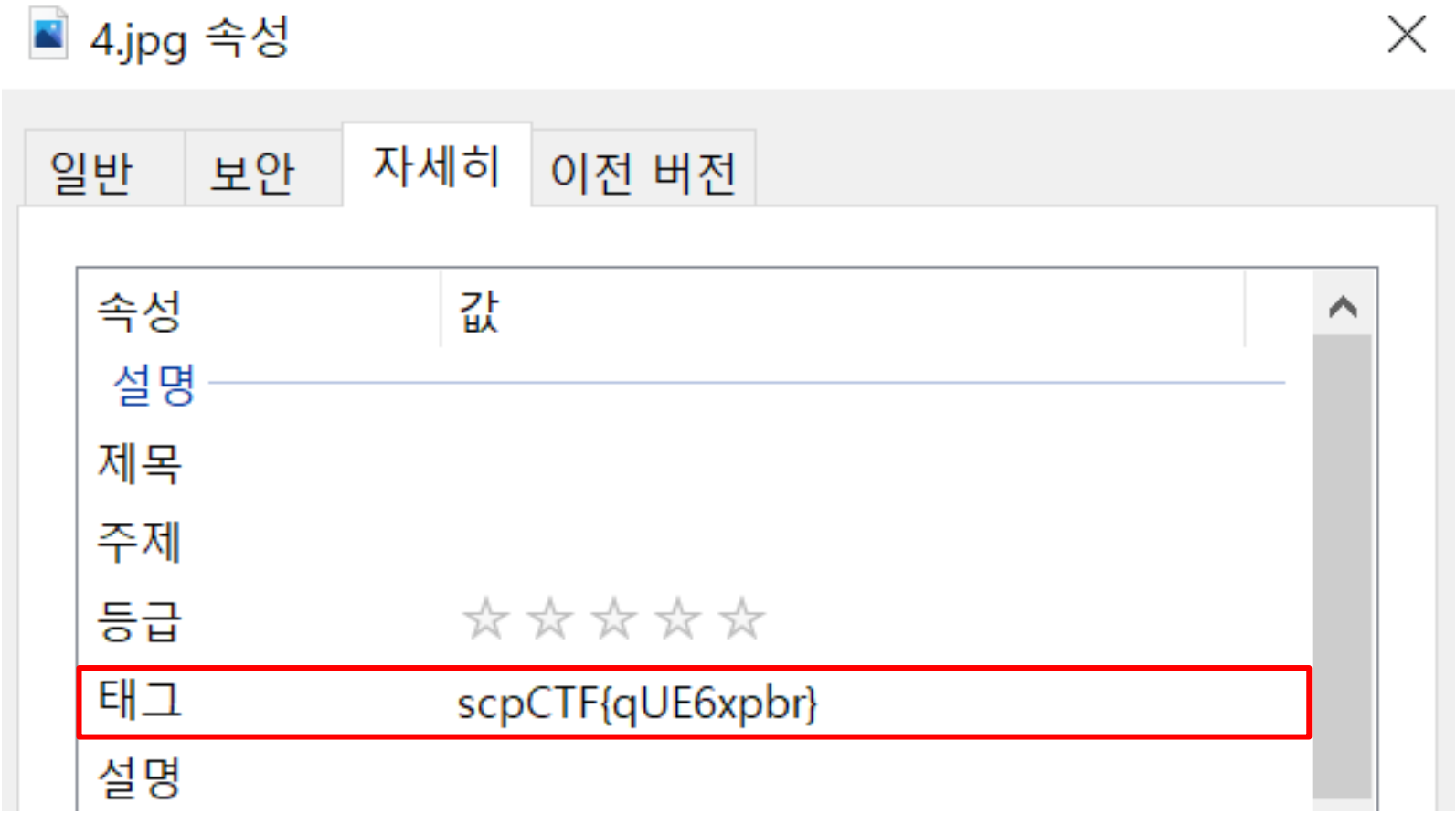
▷ 토이스토리 6명의 친구들 중 한 친구만
이 플래그를 가지고 있습니다! 찾아봅시다!







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
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]
```

scpCTF{5841}

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//1adZuudtysflOrZaANLUp6qCZUJiTzo=
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 IDPW::.text

Address	Disassembly	Text string
00401040	PUSH OFFSET IDPW.??_C@_0DE@EMOIDMKC@?6?!	ASCII "▯ ID: i+dG+bhR15//1adZuudtysflOrZaAMLUp6qCZUJiTzo=▯"
0040104A	PUSH OFFSET IDPW.??_C@_0DE@MPMEHBEJ@?5P	ASCII " PW: E2h49+snqtk3WTuU9+huRYjttm99QJAs/XYLsOPtb7U=▯▯"
00401054	PUSH OFFSET IDPW.??_C@_0CH@LHFIOOPM@?5T	ASCII " This is not a valid ID and passuord.▯"
004012D0	JMP IDPW.__scrt_common_main_seh	(Initial CPU selection)
00401342	ASCII "J$1@",0	

ID & PW

Decryption

Encrypted Text

E2h49+snatk3WTwU9+huRVittm99QJAs/XYLsOPtb7U=

Decrypt

Decrypted Text

juice

ID & PW

main 함수 시작 주소 찾기 : 참조되는 문자열 확인

Text strings referenced in IDPW:.text

Address	Disassembly	Text string
00401040	PUSH OFFSET IDPW.??_C@_0DE@EMOIDMKC@?6?!	ASCII "[redacted] ID: i+dG+bhR15//1adZuudtysflOrZaAMLUp6qCZUJiTzo=[redacted]"
0040104A	PUSH OFFSET IDPW.??_C@_0DE@MPMEHBEJ@?5PI	ASCII " PW: E2h49+sngtk3WTuU9+huRYjttn99QJAs/XYLsOPtb7U=[redacted]"
00401054	PUSH OFFSET IDPW.??_C@_0CH@LHFIOOPM@?5TI	ASCII " This is not a valid ID and password.[redacted]"
004012D0	JMP IDPW.__scrt_common_main_seh	(Initial CPU selection)
00401342	ASCII "J$1@",0	

ID & PW

```
00401040  $  68 08214000   PUSH OFFSET IDPW.??_C@_0DE@EMOIDMKC@?6?5   format = "■ ID: i+dG+bhR15//1adZuudtysflOrZaAMLUp6qCZUJi
00401045  .  E8 C6FFFFFF   CALL IDPW.printf                          printf
0040104A  .  68 3C214000   PUSH OFFSET IDPW.??_C@_0DE@MPMEHBEJ@?5PI   format = " PW: E2h49+sngtk3WTuU9+huRVjttn99QJAs/XYLsOPtb
0040104F  .  E8 BCFFFFFF   CALL IDPW.printf                          printf
00401054  .  68 70214000   PUSH OFFSET IDPW.??_C@_0CH@LHFIOOPM@?5TI   format = " This is not a valid ID and passuord.■"
00401059  .  E8 B2FFFFFF   CALL IDPW.printf                          printf
0040105E  .  83C4 0C       ADD ESP,0C
00401061  .  FF15 00204001 CALL DWORD PTR DS:[<&KERNEL32.GetConsole  KERNEL32.GetConsoleWindow
00401067  .  6A 00         PUSH 0                                    ShowState = SW_HIDE
00401069  .  50            PUSH EAX                                  hWnd
0040106A  .  FF15 38204001 CALL DWORD PTR DS:[<&USER32.ShowWindow>]  ShowWindow
0040107D  .  33C0          XOR EAX,EAX
00401072  .  C3            RETN
```

scpCTF{coconutjuice00401040}

감사합니다