

[hspace]rev-serial writeup

Find out what 'name' is, and what 'serial' number the name identifier has.

: name이 무엇인지, name 식별자가 갖는 serial 번호는 무엇인지 찾아보세요.

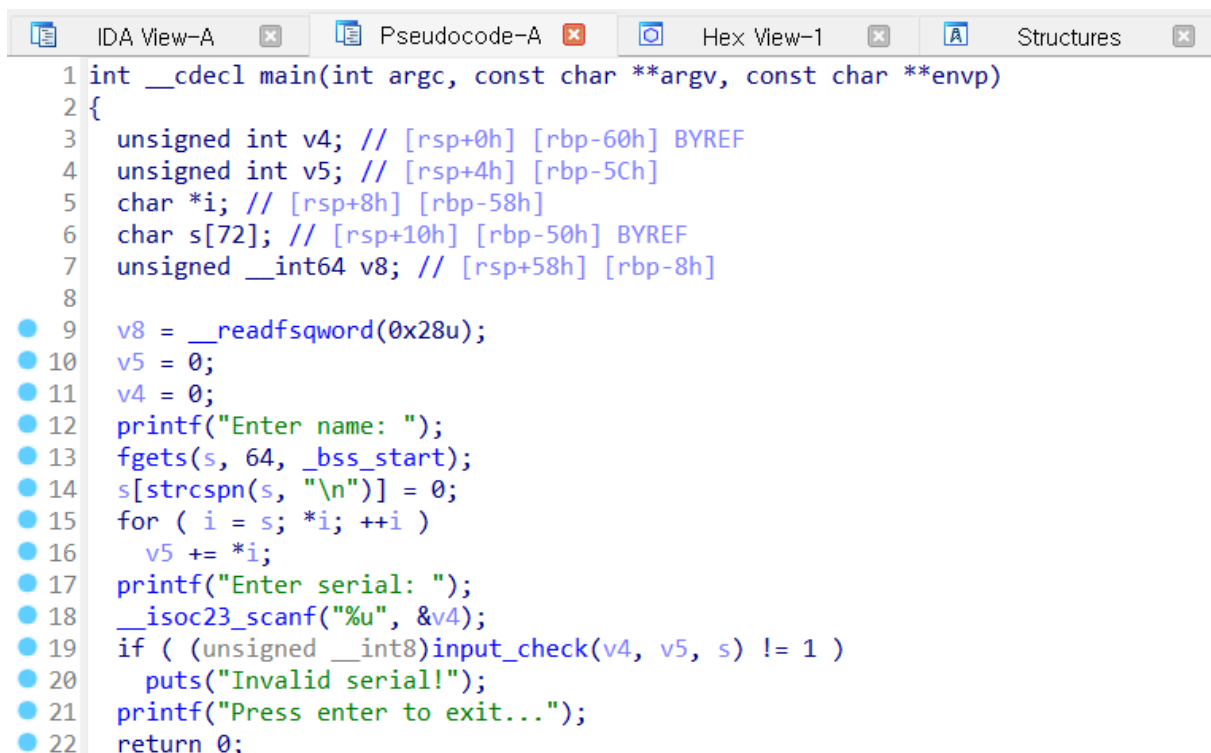
check_flag 파일

```
Enter name:
Enter serial:
|
```

사용자로부터 정해진 이름(name)과 시리얼(serial) 값을 입력받아, 조건을 만족하면 플래그를 출력한다.

serial_prob 파일

(1) main



```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     unsigned int v4; // [rsp+0h] [rbp-60h] BYREF
4     unsigned int v5; // [rsp+4h] [rbp-5Ch]
5     char *i; // [rsp+8h] [rbp-58h]
6     char s[72]; // [rsp+10h] [rbp-50h] BYREF
7     unsigned __int64 v8; // [rsp+58h] [rbp-8h]
8
9     v8 = __readfsqword(0x28u);
10    v5 = 0;
11    v4 = 0;
12    printf("Enter name: ");
13    fgets(s, 64, _bss_start);
14    s[strlen(s)] = 0;
15    for ( i = s; *i; ++i )
16        v5 += *i;
17    printf("Enter serial: ");
18    __isoc23_scanf("%u", &v4);
19    if ( (unsigned __int8)input_check(v4, v5, s) != 1 )
20        puts("Invalid serial!");
21    printf("Press enter to exit...");
22    return 0;
```

```

printf("Enter name: ");
fgets(s, 64, _bss_start);
s[strcspn(s, "\n")] = 0; // 개행 문자 제거
for ( i = s; *i; ++i ) //각 문자의 아스키 코드를 모두 더한다
    v5 += *i;

```

이름(name)을 입력받아 s에 저장하고 개행 문자를 제거한다. 이후 이름을 구성하는 각 문자의 아스키 코드 값을 모두 더해 v5에 저장한다. 시리얼(serial) 값은 입력받아 v4에 저장하고, input_check 함수로 입력값인 이름(name)과 시리얼(serial)을 검증한다.

(2) input_check 함수

```

1  __int64 __fastcall input_check(int a1, unsigned int a2, const char *a3)
2  {
3      const char *i; // [rsp+10h] [rbp-C0h]
4      char src[4]; // [rsp+1Ch] [rbp-B4h] BYREF
5      char v7[32]; // [rsp+20h] [rbp-B0h] BYREF
6      char s[136]; // [rsp+40h] [rbp-90h] BYREF
7      unsigned __int64 v9; // [rsp+C8h] [rbp-8h]
8
9      v9 = __readfsqword(0x28u);
10     if ( a1 != a2 || strcmp(a3, "hello") )
11         return 0LL;
12     encode_serial(a2, v7);
13     snprintf(s, 0x80uLL, "hspace{%s", v7);
14     for ( i = a3; *i; ++i )
15     {
16         snprintf(src, 4uLL, "%hhu", *(unsigned __int8 *)i);
17         strcat(s, src);
18     }
19     *(_WORD *)&s[strlen(s)] = 125;
20     puts(s);
21     return 1LL;
22 }

```

시리얼 값(v4)와 이름 문자의 아스키 합(v5)이 같아야 하며, 입력받은 name(s)이 hello여야 한다는 것을 확인할 수 있다.

이름이 hello라는 것을 확인했으니, 각 문자의 아스키 코드 값을 모두 더하면 시리얼 값을 찾을 수 있다.

ASCII 코드 변환

입력된 ASCII 코드를 쉽게 변환할 수 있습니다.

2진수, 10진수, 16진수에서 문자로 변환하거나 문자를 2진수, 10진수, 16진수로 변환할 수 있습니다.

변환 전

텍스트 ▼

→

변환 후

10진수 ▼

구분자:

공간 ▼

변환 전:

hello

변환하다

변환 후:

104 101 108 108 111

위로 복사

- 'h' = 104
- 'e' = 101
- 'l' = 108
- 'l' = 108
- 'o' = 111

이므로 $104 + 101 + 108 + 108 + 111 = 532$ 가 된다. 시리얼 값은 532라는 것을 알 수 있다.

함께 주어진 check_flag.exe 파일에 올바른 name과 serial 값을 넣으면 flag가 바로 출력된다.

+ check_flag.exe 없이 풀이하는 write-up [난이도 조정]

(3) encode_serial 함수

encode_serial 함수로 시리얼 값을 인코딩하여 플래그를 구할 수도 있다.

```
encode_serial(a2, v7);
snprintf(s, 0x80uLL, "hspace{%s", v7);
for ( i = a3; *i; ++i )
{
    snprintf(src, 4uLL, "%hhu", *(unsigned __int8 *)i);
    strcat(s, src);
}
```

input_check 함수를 보면 flag는 "hspace{<인코딩된 시리얼><이름 각 문자 아스키값>}" 형태로 출력된다.

```
unsigned __int64 __fastcall encode_serial(unsigned int a1, char *a2)
{
    unsigned __int8 v2; // a1
    unsigned __int8 v3; // a1
    int v4; // eax
    char v5; // dl
    int v6; // eax
    char v7; // dl
    int v8; // eax
    unsigned __int8 v10; // [rsp+16h] [rbp-3Ah]
    int v11; // [rsp+18h] [rbp-38h]
    int v12; // [rsp+18h] [rbp-38h]
    int v13; // [rsp+18h] [rbp-38h]
    unsigned int v14; // [rsp+1Ch] [rbp-34h]
    size_t i; // [rsp+20h] [rbp-30h]
    size_t v16; // [rsp+28h] [rbp-28h]
    char s[24]; // [rsp+30h] [rbp-20h] BYREF
    unsigned __int64 v18; // [rsp+48h] [rbp-8h]

    v18 = __readfsqword(0x28u);
    snprintf(s, 0x14uLL, "%u", a1);
    v16 = strlen(s);
    v11 = 0;
    for ( i = 0LL; i < v16; i += 3LL )
```

```

{
  if ( i + 1 >= v16 )
    v2 = 0;
  else
    v2 = s[i + 1] - 48;
  v10 = v2;
  if ( i + 2 >= v16 )
    v3 = 0;
  else
    v3 = s[i + 2] - 48;
  v14 = (16 * v10) | ((unsigned __int8)(s[i] - 48) << 8) | v3;
  a2[v11] = bb[(v14 >> 12) & 0x3F];
  v4 = v11 + 1;
  v12 = v11 + 2;
  a2[v4] = bb[(v14 >> 6) & 0x3F];
  if ( i + 1 >= v16 )
    v5 = 61;
  else
    v5 = bb[v14 & 0x3F];
  v6 = v12;
  v13 = v12 + 1;
  a2[v6] = v5;
  if ( i + 2 >= v16 )
    v7 = 61;
  else
    v7 = *bb;
  v8 = v13;
  v11 = v13 + 1;
  a2[v8] = v7;
}
a2[v11] = 0;
return v18 - __readfsqword(0x28u);
}

```

$a1 = 532$, $v16 = \text{strlen}(s) = 30$ 이다.

각 자리를 추출하면

- $b1 = s - '0' = '5' - '0' = 5$
- $b2 = s - '0' = '3' - '0' = 3$

- $b3 = s - '0' = '2' - '0' = 2$

v14를 계산하면

$$\begin{aligned} v14 &= (16 * b2) \mid (b1 << 8) \mid b3 \\ &= (16 * 3) \mid (5 << 8) \mid 2 \\ &= 48 \mid 1280 \mid 2 \\ &= (1280 \mid 48) = 1328 \mid 2 = 1330 \end{aligned}$$

인코딩하면

bb =

"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

- $(v14 >> 12) \& 0x3F = (1330 >> 12) \& 0x3F = (0) \& 0x3F = 0 \rightarrow 'A'$
- $(v14 >> 6) \& 0x3F = (1330 >> 6) \& 0x3F = (20) \& 0x3F = 20 \rightarrow 'U'$
- $v14 \& 0x3F = 1330 \& 0x3F = 58 \rightarrow 'y'$
- $(v14 << 6) \& 0x3F = (1330 << 6) \& 0x3F = (85120) \& 0x3F = 0 \rightarrow 'A'$

따라서

- $a2 = 'A'$
- $a2 = 'U'$
- $a2 = 'y'$
- $a2 = 'A'$
- $a2 = '\0'$

이 나오므로, <인코딩된 시리얼><이름 각 문자 아스키값>으로 **플래그를 조합하면**
hspace{AUyA104101108108111}이 나온다.

```
Enter name: hello
Enter serial: 532
hspace{AUyA104101108108111}
계속하려면 아무 키나 누르십시오 . . . |
```

