Writeup soal_rev0

```
soal_rev0: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 3.2.0, BuildI
D[sha1]=fd2aead0f553d0193cf89fee193083ae5b09a307, not stripped
```

```
gfff
gfff
UWVS
[^_]
AK8l%
A>Ct2?
DJRNTmXb"_]N7=?\SJb
W9M%X1\
FLAG???
WRONG FLAG!
CORRECT!
```

Pada soal ini tool strings tidak memberikan output yang berguna, sehingga analisa lebih lanjut perlu dilakukan.

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    signed int v3; // ST20_4
4    int v5; // [esp+0h] [ebp-50h]
5    int i; // [esp+4h] [ebp-4Ch]
6    char format; // [esp+16h] [ebp-3Ah]
7    char v8; // [esp+17h] [ebp-39h]
8    char v9; // [esp+18h] [ebp-38h]
9    char v10; // [esp+19h] [ebp-37h]
10   char v11; // [esp+1Ah] [ebp-36h]
11   char v12[41]; // [esp+1Bh] [ebp-35h]
12   unsigned int v13; // [esp+44h] [ebp-Ch]
13   int *v14; // [esp+48h] [ebp-8h]
14
15   v14 = &argc;
16   v13 = __readgsdword(0x14u);
17   v3 = strlen((const char *)WEW);
18   format = 37;
19   v8 = v3 / 10 + 48;
20   v9 = v3 % 10 + 48;
21   v10 = 115;
22   v11 = 0;
23   puts("FLAG???");
24   scanf(&format, v12);
25   v5 = 0;
26   for ( i = v3 - 1; i >= 0; --i )
27   {
28     if ( v12[HAH[v5]] - HEH[i] != *((char *)WEW + v5) )
29     {
30       puts("WRONG FLAG!");
31       return 0;
32     }
33     ++v5;
34   }
35   return puts("CORRECT!");
36 }
```

Program dibuka di tool IDA pro untuk analisa statis lebih lanjut. User input diterima menggunakan fungsi scanf, dengan format string yang disusun pada saat eksekusi sehingga tidak dapat ditemukan di section .data, format string tersebut menerima suatu string, dengan jumlah karakter sebanyak panjang dari string WEW yang ditampung di variabel v3. User input ditampung di variabel v12. Pengecekan input dilakukan di suatu loop, pada saat iterasi v12 indeks ke HAH[v5] diperiksa apakah sama dengan WEW[v5] jika dikurangi dengan HEH[i], dengan [i | i <- [v3-1, v3-2..0]] dan v3 ≥ v5 ≥ 0. Panjang WEW (dan panjang flag) belum diketahui.

```
.rodata:000007E0 unk_7E0       db  41h ; A          ; DATA XREF: .data:WEW↓o
.rodata:000007E1               db  4Bh ; K
.rodata:000007E2               db  38h ; 8
.rodata:000007E3               db  6Ch ; l
.rodata:000007E4               db  25h ; %
.rodata:000007E5               db  1Ch
.rodata:000007E6               db  41h ; A
.rodata:000007E7               db  3Eh ; >
.rodata:000007E8               db  43h ; C
.rodata:000007E9               db  74h ; t
.rodata:000007EA               db  32h ; 2
.rodata:000007EB               db  3Fh ; ?
.rodata:000007EC               db  1Fh
```

```
.data:00002020                 public HAH
.data:00002020 ; int HAH[]
.data:00002020 HAH             dd 1Ch            ; DATA XREF: main+D6↑r
.data:00002024                 dd offset dword_20
.data:00002028                 db   3
.data:00002029                 db   0
.data:0000202A                 db   0
.data:0000202B                 db   0
.data:0000202C                 db  10h
.data:0000202D                 db   0
.data:0000202E                 db   0
.data:0000202F                 db   0
.data:00002030                 db  0Ah
.data:00002031                 db   0
.data:00002032                 db   0
.data:00002033                 db   0
.data:00002034                 db   7
.data:00002035                 db   0
.data:00002036                 db   0
.data:00002037                 db   0
.data:00002038                 db   5
.data:00002039                 db   0
.data:0000203A                 db   0
.data:0000203B                 db   0
.data:0000203C                 db   1
.data:0000203D                 db   0
.data:0000203E                 db   0
.data:0000203F                 db   0
.data:00002040                 db  25h ; %
.data:00002041                 db   0
.data:00002042                 db   0
.data:00002043                 db   0
.data:00002044                 db  1Bh
.data:00002045                 db   0
.data:00002046                 db   0
.data:00002047                 db   0
.data:00002048                 db  27h ; '
.data:00002049                 db   0
.data:0000204A                 db   0
.data:0000204B                 db   0
.data:0000204C                 db   8
.data:0000204D                 db   0
.data:0000204E                 db   0
.data:0000204F                 db   0
.data:00002050                 db  12h
.data:00002051                 db   0
.data:00002052                 db   0
```

g adalah 40
ng HAH, HEH,

si integer,

dengan nilai
a tidak sama

si dari array

```
heh = [3, 31, 26, 15, 33, 38, 14, 30, 12, 28, 22, 9, 8, 36, 40, 23, 21, 19, 24, 4, 27, 7, 2, 17, 20, 5, 37, 18, 32, 34, 1, 35, 10, 11, 29, 16, 6,
39, 25, 13]
hah = [28, 32, 3, 16, 10, 7, 5, 1, 37, 27, 39, 8, 18, 38, 36, 26, 25, 14, 0, 17, 4, 11, 12, 24, 15, 33, 6, 20, 2, 9, 22, 31, 23, 13, 29, 19, 30,
35, 34, 21]
wew =
"\x41\x4b\x38\x6c\x25\x1c\x41\x3e\x43\x74\x32\x3f\x1f\x44\x4a\x52\x4e\x54\x6d\x58\x62\x22\x5f\x5d\x4e\x37\x3d\x3f\x5c\x53\x4a\x62\x12\x57\x39\x4d\x25
flag = ['=' for i in range(40)]
heh = heh[::-1]
for i in range(len(heh)):
        flag[hah[i]] = chr(ord(wew[i]) + heh[i])
print ''.join(flag)
```

HEH, HAH, dan string WEW disalin, kemudian string WEW dibalik menggunakan list comprehension, hasil ditampung pada variabel flag yang diberi nilai awal. Karakter flag indeks ke HAH[i] didapatkan dengan cara menambahkan nilai WEW[i] dengan HEH[i], dengan $0 \leq i < 40$. Kemudian hasil dari operasi terakhir di print.

```
mrcoffee@mrcoffee-SVF14218SGW:~/Documents$ python solver_soal_rev0.py
tHe_fLa9_i5:reVers1nG_f0r_fuN_4nd_PrOfiT
mrcoffee@mrcoffee-SVF14218SGW:~/Documents$ ./soal_rev0
FLAG???
tHe_fLa9_i5:reVers1nG_f0r_fuN_4nd_PrOfiT
CORRECT!
```

Flagnya adalah tHe_fLa9_i5:reVers1nG_f0r_fuN_4nd_PrOfiT.