

# XCTF logmein

原创

YenKoc 于 2020-01-13 20:54:21 发布 563 收藏

分类专栏: [XCTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/YenKoc/article/details/103963881>

版权

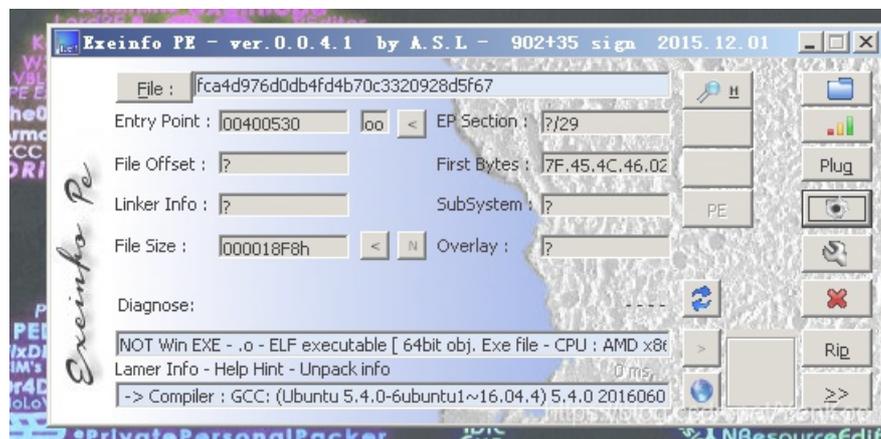


[XCTF 专栏收录该内容](#)

26 篇文章 2 订阅

订阅专栏

一.查壳



发现是64位的Linux文件 (ELF可以看出是linux的文件)

二.拖入ida64, 静态分析

Address	Length	Type	String
.rodata:0000002D	0000002D	C	Welcome to the RC3 secure password guesser.\n
.rodata:00000033	00000033	C	To continue, you must enter the correct password.\n
.rodata:00000013	00000013	C	Enter your guess:\n
.rodata:00000005	00000005	C	%32s
.rodata:00000015	00000015	C	Incorrect password!\n
.rodata:0000002E	0000002E	C	You entered the correct password!\nGreat job!\n
.eh_frame:00000006	00000006	C	\x01\x1B\x03:@
.eh_frame:00000006	00000006	C	,
.eh_frame:00000006	00000006	C	滯
.eh_frame:00000006	00000006	C	渾
.eh_frame:00000006	00000006	C	,
.eh_frame:00000006	00000006	C	\\
.eh_frame:00000007	00000007	C	咀
.eh_frame:00000007	00000007	C	,
.eh_frame:0000000E	0000000E	C	\x01\x10\x01\x1B\F\a\b
.eh_frame:00000006	00000006	C	8
.eh_frame:0000000B	0000000B	C	\x01\x10\x01\x1B\F\a\b
.eh_frame:00000006	00000006	C	檄
.eh_frame:0000000C	0000000C	C	\x0E\x10F\x0E\x18J\x0F\vw\b€
.eh_frame:00000008	00000008	C	?\x1A;*3\$\n
.eh_frame:00000009	00000009	C	A\x0E\x10
.eh_frame:00000006	00000006	C	P
.eh_frame:00000009	00000009	C	A\x0E\x10
.eh_frame:00000006	00000006	C	.
.eh_frame:00000009	00000009	C	A\x0E\x10
.eh_frame:00000006	00000006	C	p
.eh_frame:00000037	00000037	C	B\x0E\x10

```

ext:0000000000000000      push    rbp
ext:0000000000000031      mov     rbp, rsp
ext:0000000000000034      sub     rsp, 90h
ext:0000000000000038      mov     rdi, offset format ; "Welcome to the RC3 secure password guess"...
ext:0000000000000045      mov     [rbp+var_4], 0
ext:000000000000004C      mov     rax, ds:qword_4008B0
ext:0000000000000054      mov     qword ptr [rbp+var_20], rax
ext:0000000000000058      mov     rax, ds:qword_4008B8
ext:0000000000000060      mov     [rbp+var_18], rax
ext:0000000000000064      mov     cx, ds:word_4008C0
ext:000000000000006C      mov     [rbp+var_10], cx
ext:0000000000000070      mov     rax, ds:qword_4008D0
ext:0000000000000078      mov     [rbp+var_28], rax
ext:000000000000007C      mov     [rbp+var_2C], 7
ext:0000000000000083      mov     al, 0
ext:0000000000000085      call   _printf
ext:000000000000008A      mov     rdi, offset aToContinueYouM ; "To continue, you must enter the correct"...
ext:0000000000000094      mov     [rbp+var_5C], eax
ext:0000000000000097      mov     al, 0
ext:0000000000000099      call   _printf
ext:000000000000009E      mov     rdi, offset aEnterYourGuess ; "Enter your guess: "
ext:00000000000000A8      mov     [rbp+var_60], eax
ext:00000000000000AB      mov     al, 0
ext:00000000000000AD      call   _printf
ext:00000000000000B2      mov     rdi, offset a32s ; "%32s"
ext:00000000000000B8      lea    rsi, [rbp+s]
ext:00000000000000C0      mov     [rbp+var_64], eax
ext:00000000000000C3      mov     al, 0

```

<https://blog.csdn.net/YenKoc>

```

1 void __fastcall __noreturn main(__int64 a1, char **a2, char **a3)
2 {
3     size_t v3; // rsi@1
4     int i; // [sp+3Ch] [bp-54h]@3
5     char s[36]; // [sp+40h] [bp-50h]@1
6     int v6; // [sp+64h] [bp-2Ch]@1
7     __int64 v7; // [sp+68h] [bp-28h]@1
8     char v8[8]; // [sp+70h] [bp-20h]@1
9     int v9; // [sp+8Ch] [bp-4h]@1
10
11     v9 = 0;
12     strcpy(v8, "\\\"AL_RT^L*.*?+6/46");
13     v7 = 28537194573619560LL;
14     v6 = 7;
15     printf("Welcome to the RC3 secure password guesser.\n", a2, a3);
16     printf("To continue, you must enter the correct password.\n");
17     printf("Enter your guess: ");
18     __isoc99_scanf("%32s", s);
19     v3 = strlen(s);
20     if ( v3 < strlen(v8) )
21         sub_4007C0();
22     for ( i = 0; i < strlen(s); ++i )
23     {
24         if ( i >= strlen(v8) )
25             sub_4007C0();
26         if ( s[i] != (*(&v7 + i % v6) ^ v8[i]) )
27             sub_4007C0();
28     }
29     sub_4007F0();
30 }

```

<https://blog.csdn.net/YenKoc>

```

12 strcpy(v8, "\\\"AL_RT^L*.*?+6/46");
13 v7 = 28537194573619560LL;

```

注意这里两个坑：

1. `strcpy`是复制字符串的意思，前面定义的 `v8` 数组只有 8 个，但是后面的字符串是超过 8 个的，所以有可能这个 `v8` 定义是反编译错误的（算了一遍，确实是错的，不需要管这个数组，直接用原字符串就好了）

2. `_int 64 v7`

```

11 v9 = 0;
12 strcpy(v8, "\\\"AL_RT^L*.*?+6/46");
13 v7 = 'ebmarah'; |
14 v6 = 7;
15 printf("Welcome to the RC3 secure password guesser.\n", a2, a3);
16 printf("To continue, you must enter the correct password.\n");

```

```

16 | printf( "to continue, you must enter the correct password.\n" );
17 | printf("Enter your guess: ");
18 | __isoc99_scanf("%32s", s);
19 | v3 = strlen(s);
20 | if ( v3 < strlen(v8) )
21 |     sub_4007C0();
22 | for ( i = 0; i < strlen(s); ++i )
23 | {
24 |     if ( i >= strlen(v8) )
25 |         sub_4007C0();
26 |     if ( s[i] != (char)((_BYTE *)&v7 + i % v6) ^ v8[i] )
27 |         sub_4007C0();
28 | }
29 | sub_4007E0();

```

<https://blog.csdn.net/YenKoc>

先将v7转换成字符串，这步个人理解是BYTE是字节型的指针，刚好对应的是一个字符，同时这个字符需要倒过来，因为是小端存储。

三。算法分析，写出脚本

```

v7="harambe"
v6=7
v8=":\\"AL_RT^L*.?+6/46"
str=""
for i in range(0,len(v8)):
    str+=chr(ord((v7[i%7]))^ord(v8[i]))
print(str)

```

得到flag

RC3-2016-XORISGUD