buuctf————[ACTF新生赛2020]fungame

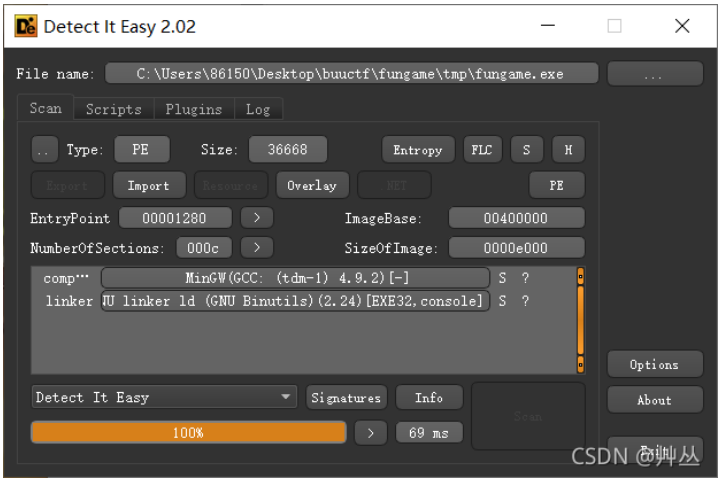
原创

re3sry 于 2021-10-11 06:19:36 发布 108 收藏 1
文章标签： buuctf reverse 栈

本文链接：https://blog.csdn.net/yhfgs/article/details/120695670

1.查壳。

无壳，32位。



2.IDA反编译。查看字符串。

可以看到两个输入（第六行，倒数第九行）

Address	Length	Type	String
.data:00403001	00000005	C	a>iTA
.data:00403007	00000009	C	Mn;eS0yE[
.rdata:004040...	0000000E	C	libgcj-13.dll
.rdata:004040...	00000014	C	_Jv_RegisterClasses
.rdata:004040...	0000003F	C	BCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
.rdata:004040...	0000000E	C	Please input:
.rdata:004042...	00000020	C	Enter the first (single) digit:
.rdata:004042...	00000022	C	Enter first group of five digits:
.rdata:004042...	00000010	C	%1d%1d%1d%1d%1d
.rdata:004042...	00000023	C	Enter second group of five digits:
.rdata:004042...	0000001F	C	Enter the last (single) digit:
.rdata:004043...	00000010	C	Check digit:%d\n
.rdata:004043...	00000006	C	VALID
.rdata:004043...	00000009	C	NO VALID
.rdata:004043...	00000013	C	Enter a sentence:
.rdata:004043...	00000016	C	Reversal of sentence:
.rdata:004043...	00000014	C	Please input again:
.rdata:004043...	00000018	C	Mingw runtime failure:\n
.rdata:004043...	00000031	C	VirtualQuery failed for %d bytes at address %p
.rdata:004043...	00000032	C	Unknown pseudo relocation protocol version %d.\n
.rdata:004044...	0000002A	C	Unknown pseudo relocation bit size %d.\n
.rdata:004044...	00000013	C	GCC: (tdm-1) 4.9.2
.rdata:004044...	00000013	C	GCC: (tdm-1) 4.9.2
.rdata:004044...	00000013	C	GCC: (tdm-1) 4.9.2
.rdata:004044...	00000013	C	GCC: (tdm-1) 4.9.2

分别跟进，

第一处：很简单的异或

```
3  char i; // [esp+1Fh] [ebp-9h]
4
5  printf("Please input:");
6  scanf("%s", a1);
7  for ( i = 0; i <= 15; ++i )
8  {
9    if ( (*(_BYTE *)(i + a1) ^ *((_BYTE *)y1 + i)) != y2[i] )
0      exit(0);
1  }
2  return 0;
3}
```


第二处：base64加密

```
7
8  printf("Please input again:");
9  strcpy(Str2, "YTFzMF9wV24=");
10 memset(Str, 0, sizeof(Str));
11 memset(Str1, 0, sizeof(Str1));
12 scanf("%s", Str);
13 v3 = strlen(Str);
14 sub_402421(Str, v3, Str1);
15 if ( !strcmp(Str1, Str2) )
16 {
17   printf("%s%s", x, Str);
18   exit(0);
19 }
20 exit(0);
21}
```


分别解密：拼接提交（卧槽），不对（还是天真）

第一处：Re_1s_So0_funny!

第二处：a1s0_pWn

3.看了看大佬的wp才知道是栈溢出。

```
5  __main();
6  v4 = malloc(0x14u);
7  memset(v4, 0, 0x14u);
8  memset(x, 0, 0x18u);
9  sub_401340((int)v4);
10 sub_4013BA((char *)v4);
11 return 0;
12}
```


在第一处加密后的函数sub_4013ba()中：在第一个strcpy()中把16位的copy到12位的引起栈溢出。

(原来是通过输入的flag产生溢出，然后用flag中的隐藏函数的地址（地址为40233D）覆盖，促使程序调用隐藏函数sub_40233D。)

```
int __cdecl sub_4013BA(char *Source)
{
  char Destination[12]; // [esp+1Ch] [ebp-Ch] BYREF

  strcpy(Destination, Source);
  strcpy(x, Source);
  return 0;
}
```


所以在两个字符串要加上chr（0x3D）chr（0x23）chr（0x40）

4.get flag

flag{Re_1s_So0_funny!=#@a1s0_pwn}