CTF Writeup - Monty Hall Game (Reverse Engineering)

Challenge: Reverse Engineering – Monty Hall Game

Category: Reverse Engineering

What I Did: Binary patching to win the game faster

Challenge Summary:

The binary was a simple Monty Hall-style game:
- You start with 100 points.
- Each game round costs 5 points.
- Winning a round gives you 15 points.
- When your points reach 10000, the game calls the reward() function to reveal the flag.

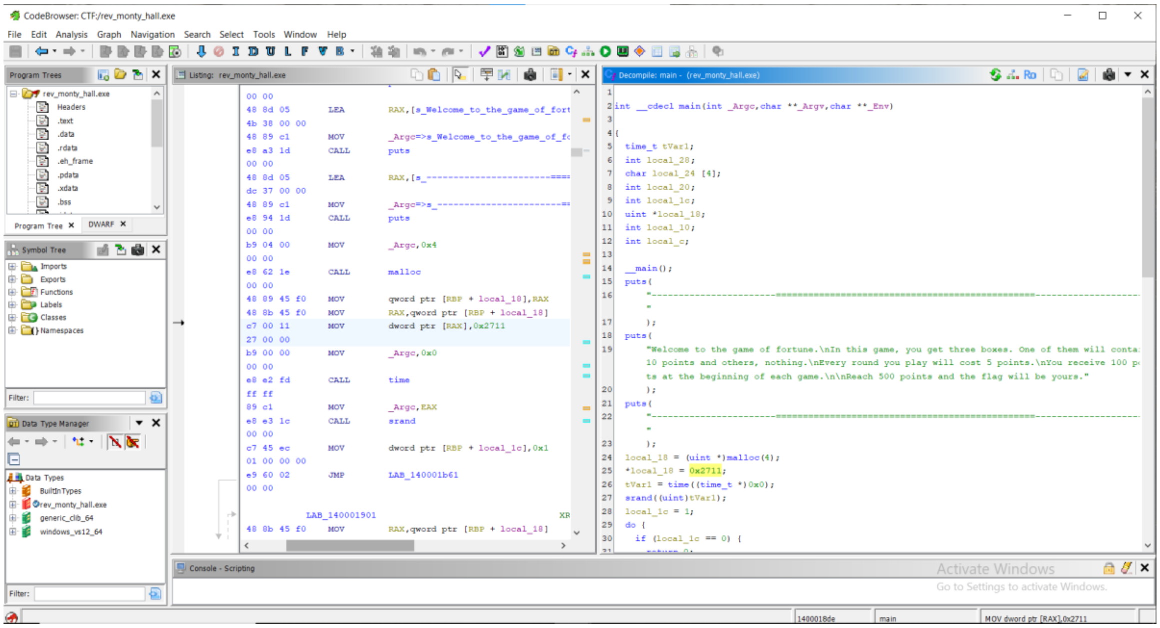
My Approach:

1. I opened the binary in Ghidra and found that the initial points (*local_18) were set to 100:

```
local_18 = malloc(4);
*local_18 = 100;
```

2. I patched the value from 100 to 10001 in the assembly (hex editor or Ghidra patcher).
3. After saving the binary, I ran it. Because the starting points were already higher than the required 10000, I just played a round...

Screenshot

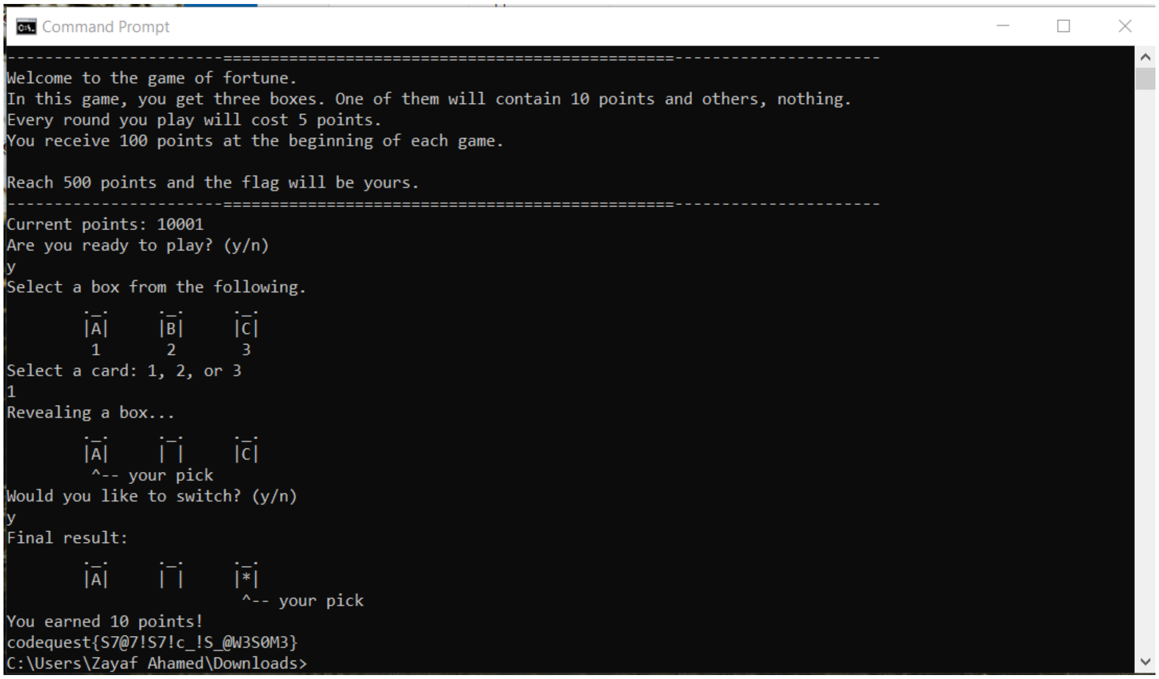


```
00 00
48 8d 05       LEA     RAX,[s_Welcome_to_the_game_of_fort
4b 38 00 00
48 89 c1       MOV     _Argc=>s_Welcome_to_the_game_of_fo
e8 a3 1d       CALL    puts
00 00
48 8d 05       LEA     RAX,[s_-----------------------====
dc 37 00 00
48 89 c1       MOV     _Argc=>s_---------------------==
e8 94 1d       CALL    puts
00 00
b9 04 00       MOV     _Argc,0x4
00 00
e8 62 1e       CALL    malloc
00 00
48 89 45 f0    MOV     qword ptr [RBP + local_18],RAX
48 8b 45 f0    MOV     RAX,qword ptr [RBP + local_18]
c7 00 11       MOV     dword ptr [RAX],0x2711
27 00 00
b9 00 00       MOV     _Argc,0x0
00 00
e8 e2 fd       CALL    time
ff ff
89 c1          MOV     _Argc,EAX
e8 e3 1c       CALL    srand
00 00
c7 45 ec       MOV     dword ptr [RBP + local_1c],0x1
01 00 00 00
e9 60 02       JMP     LAB_140001b61
00 00

          LAB_140001901                              XR
48 8b 45 f0    MOV     RAX,qword ptr [RBP + local_18]
```

```

int __cdecl main(int _Argc,char **_Argv,char **_Env)

{
  time_t tVar1;
  int local_28;
  char local_24 [4];
  int local_20;
  int local_1c;
  uint *local_18;
  int local_10;
  int local_c;

  __main();
  puts(
      "---------------------==================================================-------------------
      "
      );
  puts(
      "Welcome to the game of fortune.\nIn this game, you get three boxes. One of them will conta
      10 points and others, nothing.\nEvery round you play will cost 5 points.\nYou receive 100 po
      ts at the beginning of each game.\n\nReach 500 points and the flag will be yours."
      );
  puts(
      "---------------------==================================================-------------------
      "
      );
  local_18 = (uint *)malloc(4);
  *local_18 = 0x2711;
  tVar1 = time((time_t *)0x0);
  srand((uint)tVar1);
  local_1c = 1;
  do {
    if (local_1c == 0) {
```

Result:

After one round, the game reached the target point condition and called the reward() function, revealing the flag.



```
-----------------------================================================----------------------
Welcome to the game of fortune.
In this game, you get three boxes. One of them will contain 10 points and others, nothing.
Every round you play will cost 5 points.
You receive 100 points at the beginning of each game.

Reach 500 points and the flag will be yours.
-----------------------================================================----------------------
Current points: 10001
Are you ready to play? (y/n)
y
Select a box from the following.
        ._.     ._.     ._.
        |A|     |B|     |C|
         1       2       3
Select a card: 1, 2, or 3
1
Revealing a box...
        ._.     ._.     ._.
        |A|     | |     |C|
         ^-- your pick
Would you like to switch? (y/n)
y
Final result:
        ._.     ._.     ._.
        |A|     | |     |*|
                         ^-- your pick
You earned 10 points!
codequest{S7@7!S7!c_!S_@W3S0M3}
C:\Users\Zayaf Ahamed\Downloads>
```

FLAG: codequest{S7@7!S7!c_!S_@W3S0M3}