Misc

关注 DK 盾谢谢喵

关注微信公众号, 发送 **0xGame 2024** 获取
flag: `0xGame{W31c0m3_70_0x64m3_2024_5p0n50r3d_8y_dkdun}`

0xGame2048

提示：通过~~一点也不~~可靠的途径，我们提前截获了0xGame2048的题目，据说这就是那时候的base编码（附件：0xGame2048.txt）

读取附件得到`ЖꝎ൜ഀ൭ᄀ੫ےپЖ˥ى꠵tՖɞɯ९ρҥe⑱`

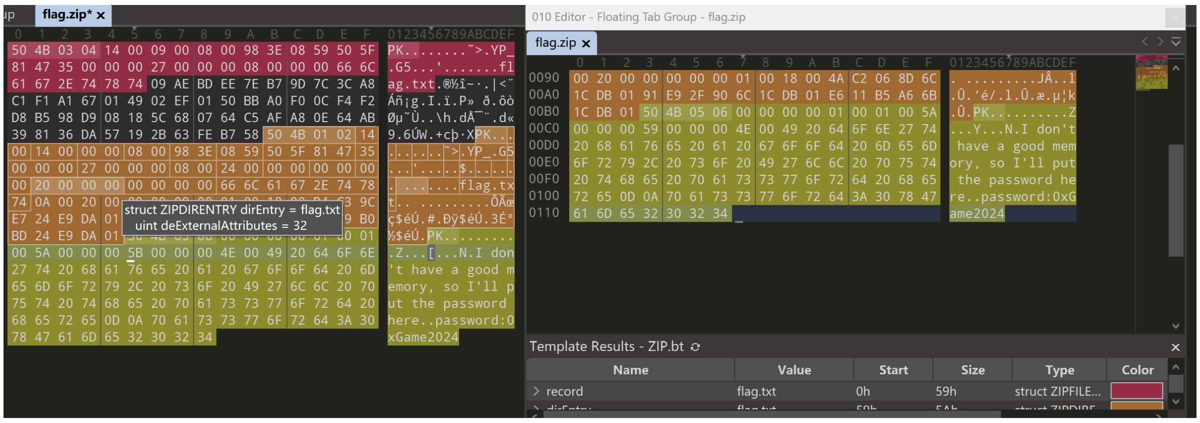
根据提示搜索base2048，发现网站[Base2048 Encoder And Decoder (nerdmosis.com)]()

对其解码后得到`0xGame{W3lc0me_t0_0xG4me!!!}`

加密的压缩包?

我其实也不会修压缩包，我自己新建了一个压缩包，也是只包含一个flag文件，39字节的，并用0xGame加密

然后用010Editor进行分析比较，发现末尾部分，有一处不同，一个是5B，一个是59，因此将5B改为59发现压缩包竟然可以解压了


flag.zip*

```
 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
50 4B 03 04 14 00 09 00 08 00 98 3E 08 59 50 5F   PK........˜>.YP_
81 47 35 00 00 00 27 00 00 00 08 00 00 00 66 6C   .G5...'.......fl
61 67 2E 74 78 74 09 AE BD EE 7E B7 9D 7C 3C A8   ag.txt.®½î~·.|<¨
C1 F1 A1 67 01 49 02 EF 01 50 BB A0 F0 0C F4 F2   Áñ¡g.I.ï.P» ð.ôò
D8 B5 98 D9 08 18 5C 68 07 64 C5 AF A8 0E 64 AB   Øµ˜Ù..\h.dÅ¯¨.d«
39 81 36 DA 57 19 2B 63 FE B7 58 50 4B 01 02 14   9.6ÚW.+cþ·XPK...
00 14 00 00 00 08 00 98 3E 08 59 50 5F 81 47 35   .......˜>.YP_.G5
00 00 00 27 00 00 00 08 00 24 00 00 00 00 00 00   ...'.....$......
00 20 00 00 00 00 00 00 00 66 6C 61 67 2E 74 78   . .......flag.tx
74 0A 00 20 00                            3 9C   t...........ÔÃœ
E7 24 E9 DA 01                            9 B0   ç$éÚ.#.Ðÿ$éÚ.3É°
BD 24 E9 DA 01                            00 01   ½$éÚ.PK.........
00 5A 00 00 00 5B 00 00 00 4E 00 49 20 64 6F 6E   .Z...[...N.I don
27 74 20 68 61 76 65 20 61 20 67 6F 6F 64 20 6D   't have a good m
65 6D 6F 72 79 2C 20 73 6F 20 49 27 6C 6C 20 70   emory, so I'll p
75 74 20 74 68 65 20 70 61 73 73 77 6F 72 64 20   ut the password 
68 65 72 65 0D 0A 70 61 73 73 77 6F 72 64 3A 30   here..password:0
78 47 61 6D 65 32 30 32 34                        xGame2024
```

struct ZIPDIRENTRY dirEntry = flag.txt
uint deExternalAttributes = 32

010 Editor - Floating Tab Group - flag.zip

flag.zip

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0090 00 20 00 00 00 00 00 01 00 18 00 4A C2 06 8D 6C   . .........JÂ..l
00A0 1C DB 01 91 E9 2F 90 6C 1C DB 01 E6 11 B5 A6 6B   .Û.'é/.l.Û.æ.µ¦k
00B0 1C DB 01 50 4B 05 06 00 00 00 00 01 00 01 00 5A   .Û.PK..........Z
00C0 00 00 00 59 00 00 00 4E 00 49 20 64 6F 6E 27 74   ...Y...N.I don't
00D0 20 68 61 76 65 20 61 20 67 6F 6F 64 20 6D 65 6D    have a good mem
00E0 6F 72 79 2C 20 73 6F 20 49 27 6C 6C 20 70 75 74   ory, so I'll put
00F0 20 74 68 65 20 70 61 73 73 77 6F 72 64 20 68 65    the password he
0100 72 65 0D 0A 70 61 73 73 77 6F 72 64 3A 30 78 47   re..password:0xG
0110 61 6D 65 32 30 32 34                              ame2024
```

Template Results - ZIP.bt

Name	Value	Start	Size	Type	Color
record	flag.txt	0h	59h	struct ZIPFILE...	🟥

查阅发现这个位置是中央目录开始位置相对位移

总之，解压后得到flag：`0xGame{M@ybe_y0u_ar2_t4e_mAsTer_0f_Z1p}`

Crypto

Caesar Cipher

密文：0yHbnf{Uif_Cfhjoojoh_Pg_Dszqup}

提示：凯撒加密。

尝试解密，发现key=1，获得flag：`0xgame{the_beginning_of_crypto}`

Code

获取附件打开得到:

```python
#How to use mathematics to represent information?
from Crypto.Util.number import bytes_to_long
from base64 import b64encode
from secret import flag

msg = flag.encode()
length = len(msg)

assert length%4 == 0
block = length//4
m = [ msg[ block*(i) : block*(i+1) ] for i in range(4) ]

m0 = m[0]
m1 = bytes_to_long(m[1])
m2 = m[2].hex()
m3 = b64encode(m[3])

print(f'm0 = {m0}\nm1 = {m1}\nm2 = {m2}\nm3 = {m3}')
'''
m0 = b'0xGame{73d7'
m1 = 60928972245886112747629873
m2 = 3165662d393339332d3034
m3 = b'N2YwZTdjNGRlMX0='
'''
```

读代码：将flag拆分为相等的4部分`m0`，`m1`，`m2`，`m3`，4部分分别：

1. `m0` 不处理。
2. `m1` 先转为16进制编码，再转换为10进制。只需要先10进制转16进制，再16进制解码即可。
3. `m2` 转16进制编码。
4. `m3` 用base64编码。

解码后拼接得到：

```
""""
m0 = 0xGame{73d7
m1 = 2f64-7656-1
m2 = 1ef-9393-04
m3 = 7f0e7c4de1}
flag = 0xGame{73d72f64-7656-11ef-9393-047f0e7c4de1}
""""
```

Code-Vigenere

获取附件打开得到：

```python
from secret import flag
from os import urandom
from base64 import b64encode

def Encrypt(msg, key):
    Lenth = len(key)
    result = ''
    upper_base = ord('A')
    lower_base = ord('a')
    KEY = [ord(key.upper()[_]) - upper_base for _ in range(Lenth)]

    index = 0
    for m in msg:
        tmp_key = KEY[index%Lenth]
        if not m.isalpha():
            result += m
            continue

        if m.isupper():
            result += chr(upper_base + (ord(m) - upper_base + tmp_key) % 26)
        else:
            result += chr(lower_base + (ord(m) - lower_base + tmp_key) % 26)
        index += 1
    return result

key = b64encode(urandom(6))[:5].decode()
print(Encrypt(flag,key))

#0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}
```

读代码：生成5位的base64编码密钥，对每个字符轮流使用密钥进行凯撒加密

由于已知flag前6位为`0xGame`且加密密钥只有5位，只需要分别找出密钥各个位使得前几位正常解密即可

可以用python枚举：

```
for i in "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/":
    # print("0" + i + "000")
    # print(Encrypt("0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}", "0" + i +
"000"))
    if (
        Encrypt("0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}", "" + i + "0000")
[1]
        == "x"
    ):
        print("Key1:", i)
    if (
        Encrypt("0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}", "0" + i + "000")
[2]
        == "G"
    ):
        print("Key2:", i)
    if (
        Encrypt("0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}", "00" + i + "00")
[3]
        == "a"
    ):
        print("Key3:", i)
    if (
        Encrypt("0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}", "000" + i + "0")
[4]
        == "m"
    ):
        print("Key4:", i)
    if (
        Encrypt("0lCcop{oyd94092-g8mq-4963-88b6-4helrxdhm6q7}", "0000" + i + "")
[5]
        == "e"
    ):
        print("Key5:", i)
```

由于字母只有26个，key不唯一，运行结果为：

```
Key2: E
Key1: M
Key5: P
Key3: Y
Key4: Y
Key2: e
Key1: m
Key5: p
Key3: y
Key4: y
Key1: 3
Key5: 6
Key2: +
```

使用 `3+yyP` 作为解密密钥得到flag：`0xGame{acb94092-e8bc-4963-88f6-4fcadbbfb6c7}`

RSA-Baby

```python
from Crypto.Util.number import bytes_to_long, getPrime
from hashlib import md5
from random import randint
from gmpy2 import invert,gcd

#Hash Function:
def MD5(m):return md5(str(m).encode()).hexdigest()

#RSA AlgorithmParameter Generation Function:
def KeyGen():
    Factor_BitLength = 30
    q = getPrime(Factor_BitLength)
    p = getPrime(Factor_BitLength)
    N = p * q
    #Euler's totient function:
    phi = (p-1) * (q-1)

    #Generate Keys:
    while True:
        e = randint(1,phi)
        if gcd(e,phi) == 1:
            d = int(invert(e,phi))
            break

    #Generate Result:
```

```
    Pub_Key = (N,e)
    Prv_Key = (N,d)
    return Pub_Key,Prv_Key

Pub,Prv = KeyGen()

N = Pub[0]
e = Pub[1]
d = Prv[1]

#RSA Encrypt:
m = randint(1,N)
c = pow(m,e,N)

print(f'Pub_Key = {Pub}')
print(f'Prv_Key = {Prv}')
print(f'Encrypt_msg = {c}')

'''
Pub_Key = (547938466798424179, 80644065229241095)
Prv_Key = (547938466798424179, 488474228706714247)
Encrypt_msg = 344136655393256706
'''

flag = '0xGame{'+ MD5(m) +'}'
```

既然已知私钥直接计算即可

```
c = 344136655393256706
d = 488474228706714247
N = 547938466798424179
m = pow(c, d, N)
flag = "0xGame{" + MD5(m) + "}"

print(f"Flag = {flag}")
```

得到flag: `0xGame{6e5719c54cdde25ce7124e280803f938}`

RSA-Easy

```
from Crypto.Util.number import bytes_to_long, getPrime
from hashlib import md5
from random import randint
from gmpy2 import invert,gcd

#Hash Function:
def MD5(m):return md5(str(m).encode()).hexdigest()

#RSA AlgorithmParameter Generation Function:
def KeyGen():
    Factor_BitLength = 30
    q = getPrime(Factor_BitLength)
    p = getPrime(Factor_BitLength)
    N = p * q
    #Euler's totient function:
    phi = (p-1) * (q-1)

    #Generate Keys:
    while True:
        e = randint(1,phi)
        if gcd(e,phi) == 1:
            break

    #Generate Result:
    Pub_Key = (N,e)
    return Pub_Key

Pub = KeyGen()

N = Pub[0]
e = Pub[1]

#RSA Encrypt:
m = randint(1,N)
c = pow(m,e,N)

print(f'Pub_Key = {Pub}')
print(f'Encrypt_msg = {c}')

'''
Pub_Key = (689802261604270193, 620245111658678815)
Encrypt_msg = 289281498571087475
'''

flag = '0xGame{'+ MD5(m) +'}'
```

尝试对N进行因式分解：

```
from sympy.ntheory import factorint

number = 689802261604270193
factors = factorint(number)
print(f"The prime factors of {number} are: {factors}")
```

得到 `689802261604270193=823642439*837502087`

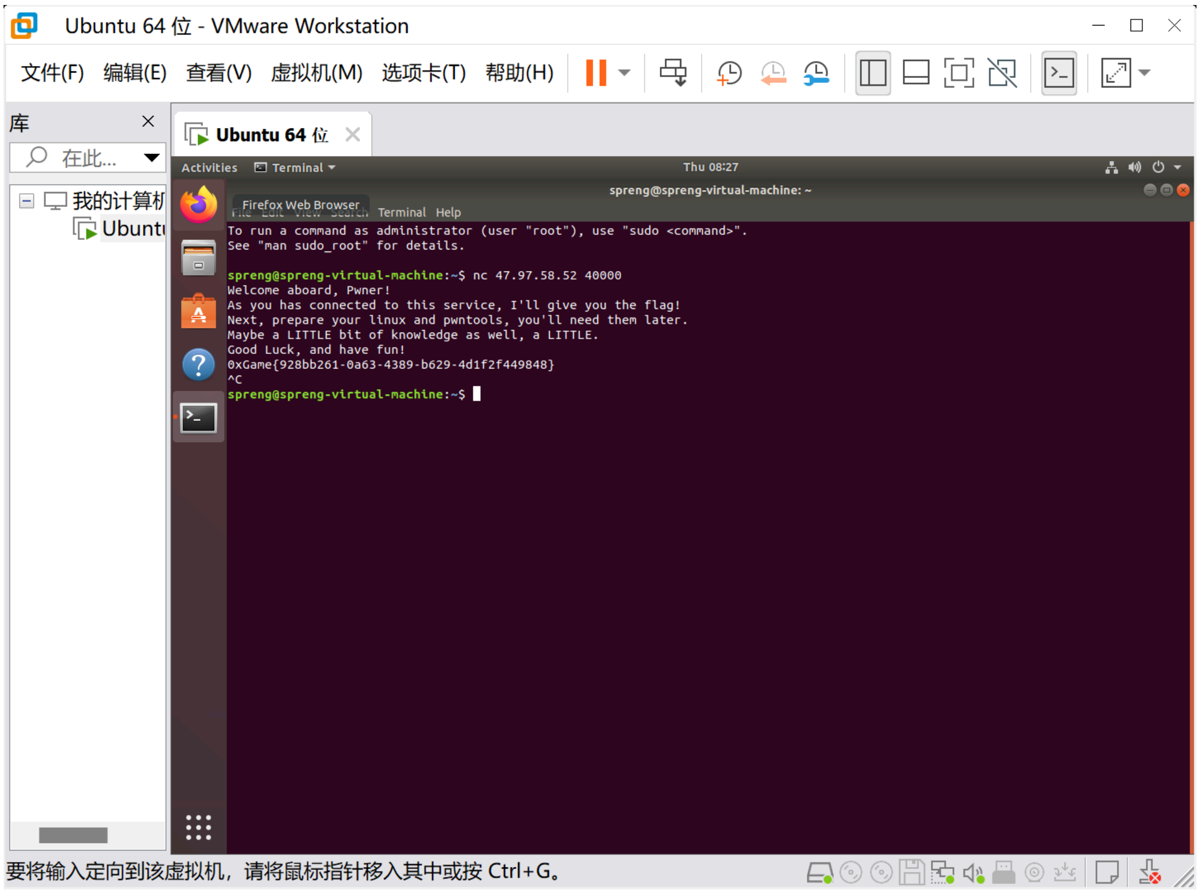
```python
N, e = 689802261604270193, 620245111658678815
c = 289281498571087475

# RSA Decrypt:
phi = (823642439 - 1) * (837502087 - 1)

d = invert(e, phi)

m = pow(c, d, N)  # 密文

flag = "0xGame{" + MD5(m) + "}"
print(f"Decrypt_msg = {m}")
print(f"Flag = {flag}")
```

计算出私钥`d:1807144943227680 91`，密文`m:302808065155328433`，密文用MD5处理得到flag：`0xGame{5aa4603855d01ffdc5dcf92e0e604f31}`。

Pwn

0. test your nc

在虚拟机中启动linux在终端输入 `nc 47.97.58.52 40000` 成功连接。


Ubuntu 64 位 - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
库
在此...
我的计算机
Ubuntu
Ubuntu 64 位
Activities
Terminal
Thu 08:27
spreng@spreng-virtual-machine: ~
Firefox Web Browser
Terminal Help
要将输入定向到该虚拟机，请将鼠标指针移入其中或按 Ctrl+G。

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

spreng@spreng-virtual-machine:~$ nc 47.97.58.52 40000
Welcome aboard, Pwner!
As you has connected to this service, I'll give you the flag!
Next, prepare your linux and pwntools, you'll need them later.
Maybe a LITTLE bit of knowledge as well, a LITTLE.
Good Luck, and have fun!
0xGame{928bb261-0a63-4389-b629-4d1f2f449848}
^C
spreng@spreng-virtual-machine:~$ 
```

获得flag： `0xGame{928bb261-0a63-4389-b629-4d1f2f449848}`

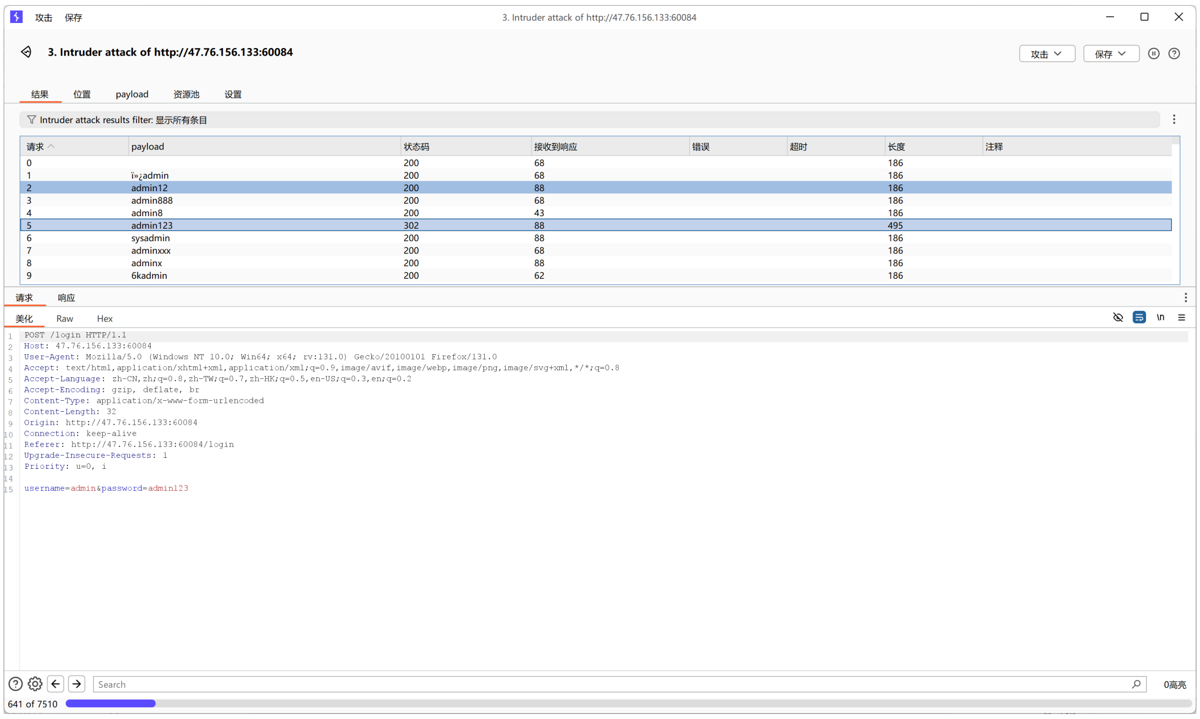
Web

ez_login

先输入账号密码用BP抓包，发送到Intruder

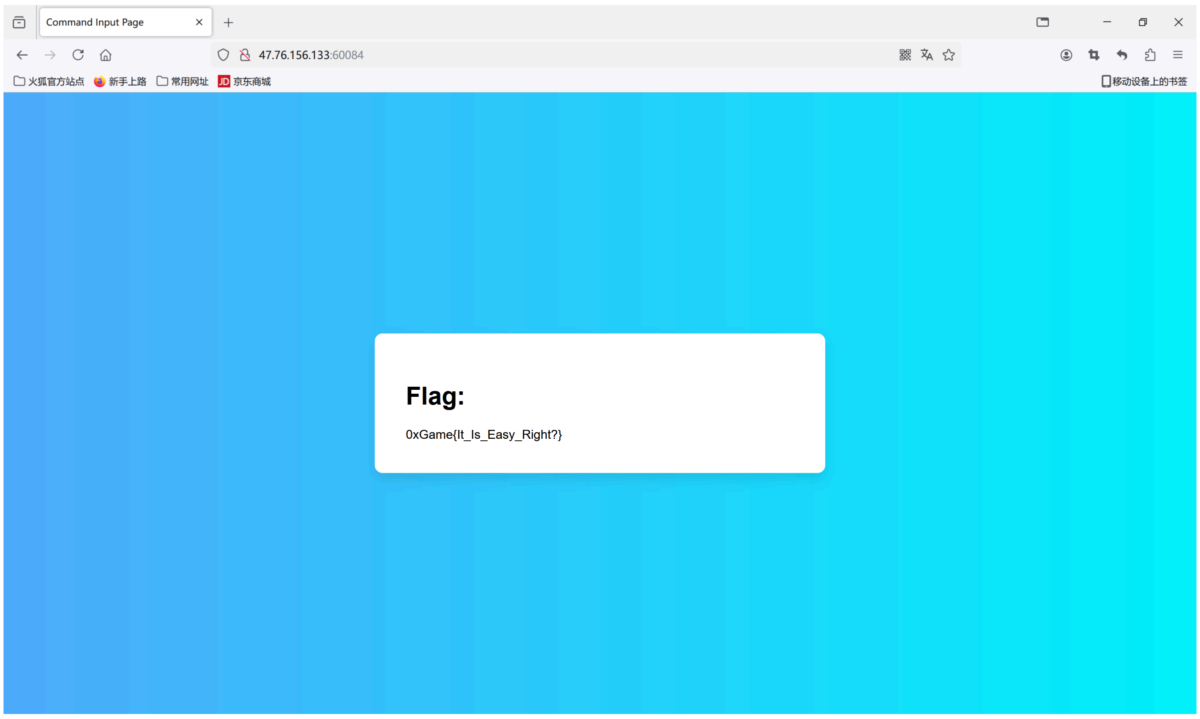
```
POST /login HTTP/1.1
Host: 47.76.156.133:60084
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101
Firefox/131.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/png,image/svg+xml,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://47.76.156.133:60084
Connection: keep-alive
Referer: http://47.76.156.133:60084/login
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=admin&password=admin
```

爆破发现，密码为admin123。


攻击
保存
3. Intruder attack of http://47.76.156.133:60084
结果
位置
payload
资源池
设置
Intruder attack results filter: 显示所有条目
请求
响应
美化
Raw
Hex
Search
0高亮
641 of 7510

请求	payload	状态码	接收到响应	错误	超时	长度	注释
0		200	68			186	
1	ï»¿admin	200	68			186	
2	admin12	200	88			186	
3	admin888	200	68			186	
4	admin8	200	43			186	
5	admin123	302	88			495	
6	sysadmin	200	88			186	
7	adminxxx	200	68			186	
8	adminx	200	88			186	
9	6kadmin	200	62			186	

```
POST /login HTTP/1.1
Host: 47.76.156.133:60084
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Origin: http://47.76.156.133:60084
Connection: keep-alive
Referer: http://47.76.156.133:60084/login
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=admin&password=admin123
```



Flag:

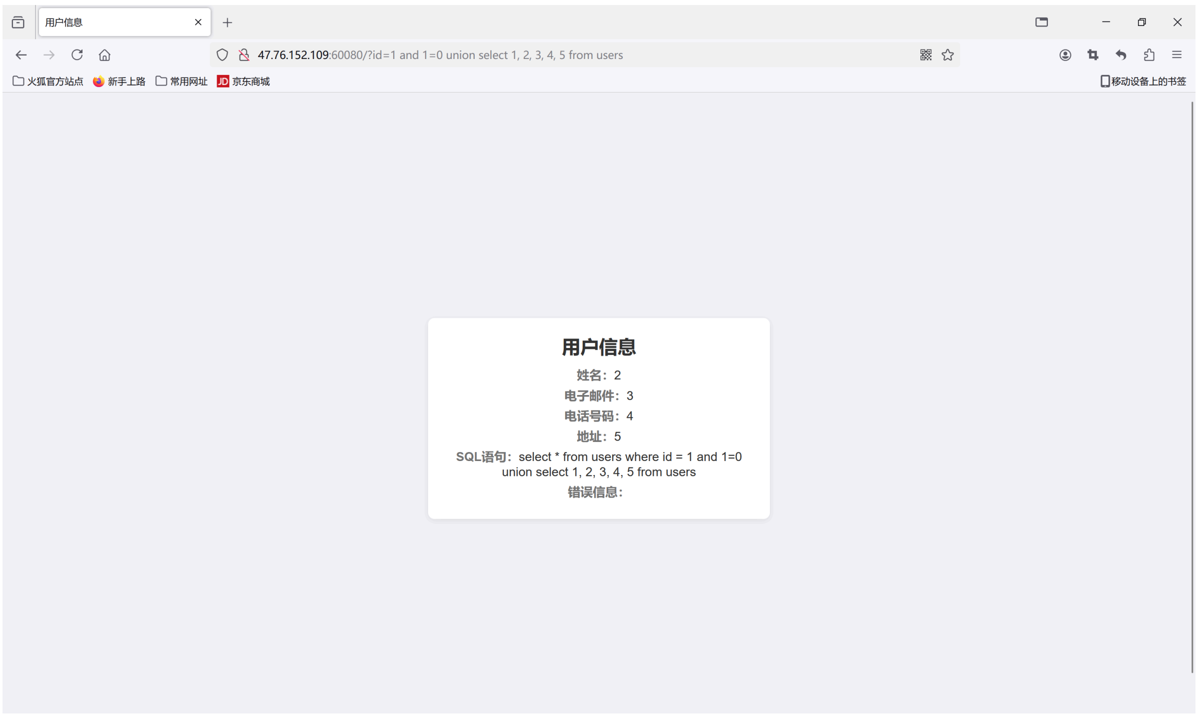
0xGame{It_Is_Easy_Right?}

获得flag: `0xGame{It_Is_Easy_Right?}`

ez_sql

输入 `http://47.76.152.109:60080/?id=1 order by 5#` 不报错，但输入 `http://47.76.152.109:60080/?id=1 order by 6#` 报错，说明该表格有五列。

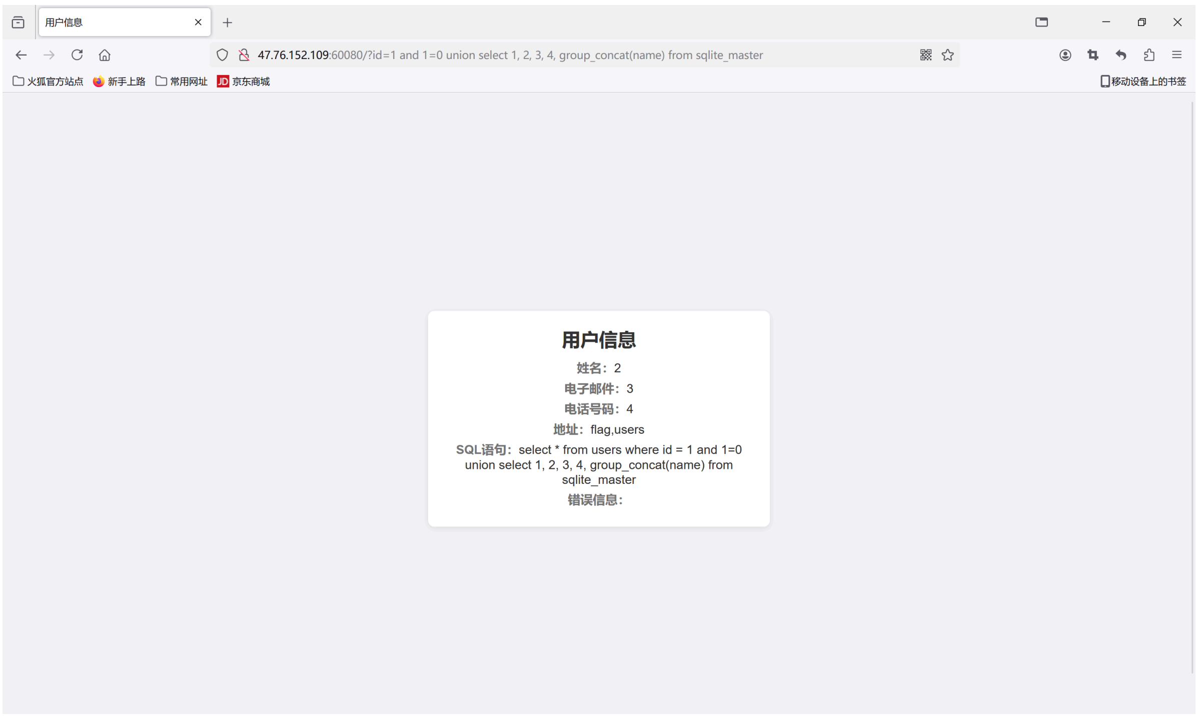
接下来就可以使用union了，先将前面的语句出错，就可以查自定义的语句了，`http://47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, 5 from users`，测试发现可以回显。


用户信息
47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, 5 from users
火狐官方站点
新手上路
常用网址
京东商城
移动设备上的书签

用户信息

姓名: 2
电子邮件: 3
电话号码: 4
地址: 5
SQL语句: select * from users where id = 1 and 1=0 union select 1, 2, 3, 4, 5 from users
错误信息:

先跑表 `http://47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(name) from sqlite_master`，获取表名`flag,users`。

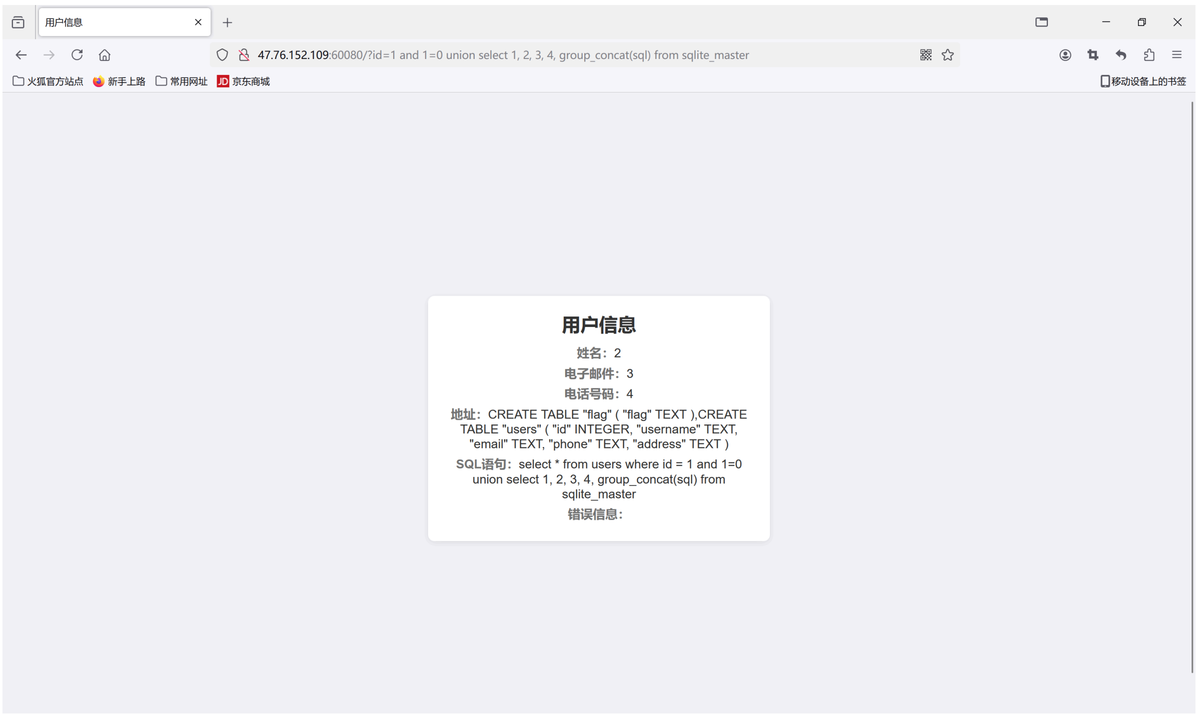

用户信息
47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(name) from sqlite_master

用户信息

姓名：2
电子邮件：3
电话号码：4
地址：flag,users
SQL语句：select * from users where id = 1 and 1=0 union select 1, 2, 3, 4, group_concat(name) from sqlite_master
错误信息：

再跑列`http://47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(sql) from sqlite_master where name = 'flag'`得到`hacker`，被过滤了，呜呜。

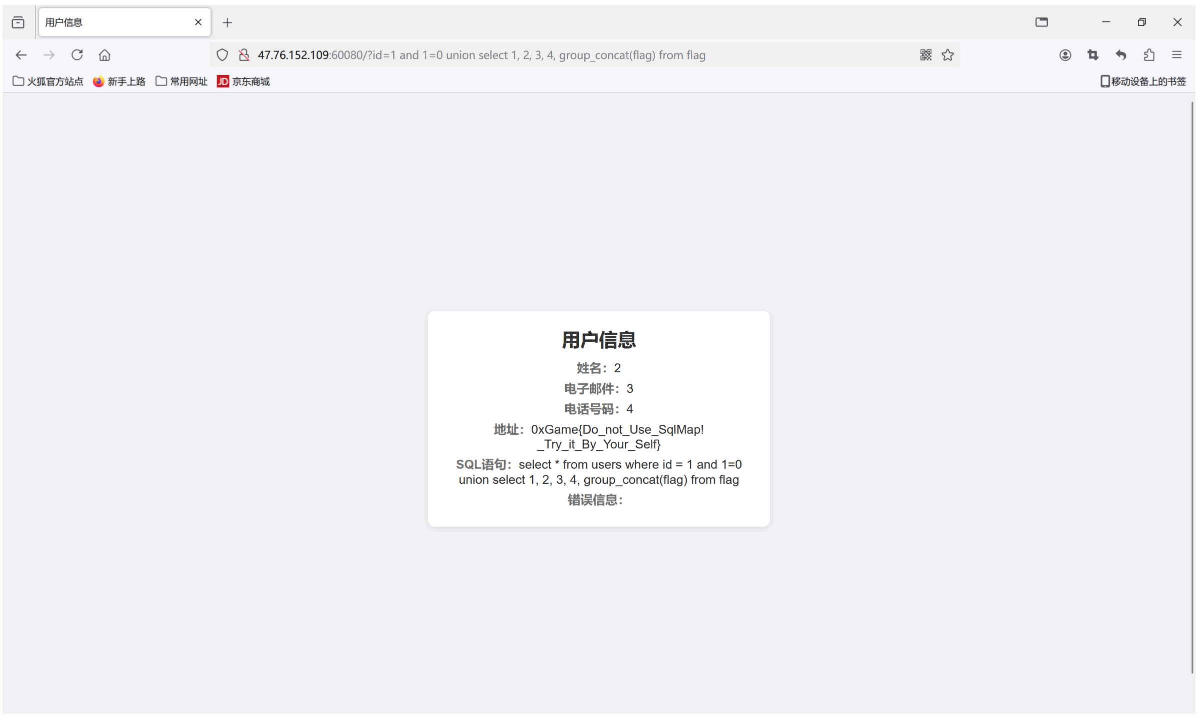
不加就是了，`http://47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(sql) from sqlite_master`，得到所有列，其中flag表只有flag列。


用户信息
47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(sql) from sqlite_master

用户信息

姓名：2
电子邮件：3
电话号码：4
地址：CREATE TABLE "flag" ("flag" TEXT),CREATE TABLE "users" ("id" INTEGER, "username" TEXT, "email" TEXT, "phone" TEXT, "address" TEXT)
SQL语句：select * from users where id = 1 and 1=0 union select 1, 2, 3, 4, group_concat(sql) from sqlite_master
错误信息：

继续跑值`http://47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(flag) from flag`


用户信息
47.76.152.109:60080/?id=1 and 1=0 union select 1, 2, 3, 4, group_concat(flag) from flag
火狐官方站点
新手上路
常用网址
京东商城
移动设备上的书签

用户信息

姓名：2

电子邮件：3

电话号码：4

地址：0xGame{Do_not_Use_SqlMap!_Try_it_By_Your_Self}

SQL语句：select * from users where id = 1 and 1=0 union select 1, 2, 3, 4, group_concat(flag) from flag

错误信息：

获得flag：`0xGame{Do_not_Use_SqlMap!_Try_it_By_Your_Self}`，哈哈，我没用SqlMap。

hello_http

用bp抓包给重放器，改一下http，他有几个要求，每完成一项奖励一点flag：

1. 用x1cBrowser浏览器访问。User-Agent 改为 x1cBrowser
2. 提交hello=world。GET后写 `/?hello=world`
3. Post提交web=security。Get改为POST，加一句 `Content-Type: application/x-www-form-urlencoded` 后输入web=security
4. 从 `http://localhost:8080/` 访问。写 `Referer: http://localhost:8080/`
5. 从 `127.0.0.1` 访问。写 `X-Forwarded-For: 127.0.0.1`

最终http为：

```
POST /?hello=world HTTP/1.1
Host: 8.130.84.100:50002
User-Agent: x1cBrowser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
=0.8
Cookie:flag=secret
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Length: 12
Content-Type: application/x-www-form-urlencoded
Referer: http://localhost:8080/
X-Forwarded-For: 127.0.0.1

web=security
```

获得flag：`0xgame{1cd6a904-725f-11ef-aafb-d4d8533ec05c}`

helloz-web

虽然他说不许F12，但还是可以用的，得到提示：

```
<!-- 看看f14g.php -->
<!-- 此乃flag的第一段：0xGame{ee7f2040-1987-4e0a -->
```

然后看看f14g.php，访问`http://8.130.84.100:50001/f14g.php`，得到提示："你知道如何查看响应包吗？"
BP看响应包得到：`æ¤ä¹flagçç¬¬äºæ®µï¼-872d-68589c4ab3d3}`

拼接得到flag：`0xGame{ee7f2040-1987-4e0a-872d-68589c4ab3d3}`

Reverse

BabyBase

用IDA打开读伪代码

```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4; // [rsp+20h] [rbp-60h]
  char Str; // [rsp+60h] [rbp-20h]
  unsigned int v6; // [rsp+ACh] [rbp+2Ch]

  _main(argc, argv, envp);
  memset(&Str, 0, 0x40ui64);
  memset(&v4, 0, 0x40ui64);
  puts(&::Str);
  scanf("%s", &Str);
  puts(&byte_405052);
  v6 = strlen(&Str);
  encode(&Str, &v4, v6);
  if ( v6 != 42 || check_flag(&v4) )
  {
    printf("Invalid!");
    exit(0);
  }
  printf("Congratulation!!");
  return 0;
}
```

查看check_flag

```c
int __fastcall check_flag(const char *a1)
{
  return strcmp(a1, "MHhHYW1le04wd195MHVfa24wd19CNHNlNjRfRW5jMGQxbmdfdzNsbCF9");
}
```

对其进行Base64解码得到flag：`0xGame{N0w_y0u_kn0w_B4se64_Enc0d1ng_w3ll!}`

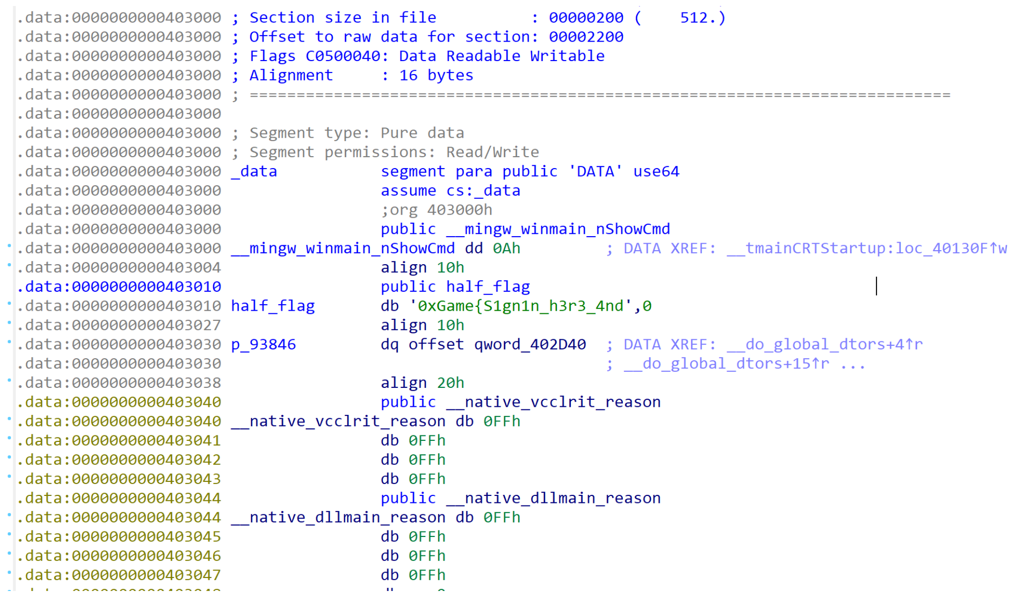
BinaryMaster

```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
  BOOL v3; // eax
  char Buffer; // [rsp+20h] [rbp-40h]
  BOOL v6; // [rsp+5Ch] [rbp-4h]

  _main(argc, argv, envp);
  puts("Welcome to the world of Binary!");
  printf("But, do you know \"Octal\" and \"Hexadecimal\"?");
  puts("\n");
  puts("This is an Oct number: 04242424");
  puts("Please convert it to Hex:");
  gets(&Buffer);
  v3 = strcmp("0x114514", &Buffer) && strcmp("114514", &Buffer);
  v6 = v3;
  if ( v3 )
  {
    puts("try again . . .");
    system("pause");
    exit(0);
  }
  puts(&byte_40409A);
  puts("You find it!");
  puts("OxGame{114514cc-a3a7-4e36-8db1-5f224b776271}");
  return 0;
}
```

emm，他想说114514转为8进制是04242424，但他已经给答案了，flag：`OxGame{114514cc-a3a7-4e36-8db1-5f224b776271}`

SignSign

先找到后半段`_b3g1n_Reversing_n0w}`，再往前翻翻发现，得到`OxGame{S1gn1n_h3r3_4nd`



```
.data:0000000000403000 ; Section size in file          : 00000200 (    512.)
.data:0000000000403000 ; Offset to raw data for section: 00002200
.data:0000000000403000 ; Flags C0500040: Data Readable Writable
.data:0000000000403000 ; Alignment     : 16 bytes
.data:0000000000403000 ; ===========================================================================
.data:0000000000403000
.data:0000000000403000 ; Segment type: Pure data
.data:0000000000403000 ; Segment permissions: Read/Write
.data:0000000000403000 _data           segment para public 'DATA' use64
.data:0000000000403000                 assume cs:_data
.data:0000000000403000                 ;org 403000h
.data:0000000000403000                 public __mingw_winmain_nShowCmd
.data:0000000000403000 __mingw_winmain_nShowCmd dd 0Ah         ; DATA XREF: __tmainCRTStartup:loc_40130F↑w
.data:0000000000403004                 align 10h
.data:0000000000403010                 public half_flag
.data:0000000000403010 half_flag       db '0xGame{S1gn1n_h3r3_4nd',0
.data:0000000000403027                 align 10h
.data:0000000000403030 p_93846         dq offset qword_402D40  ; DATA XREF: __do_global_dtors+4↑r
.data:0000000000403030                                         ; __do_global_dtors+15↑r ...
.data:0000000000403038                 align 20h
.data:0000000000403040                 public __native_vcclrit_reason
.data:0000000000403040 __native_vcclrit_reason db 0FFh
.data:0000000000403041                 db 0FFh
.data:0000000000403042                 db 0FFh
.data:0000000000403043                 db 0FFh
.data:0000000000403044                 public __native_dllmain_reason
.data:0000000000403044 __native_dllmain_reason db 0FFh
.data:0000000000403045                 db 0FFh
.data:0000000000403046                 db 0FFh
.data:0000000000403047                 db 0FFh
```

flag：`OxGame{S1gn1n_h3r3_4nd_b3g1n_Reversing_n0w}`

Xor-Beginning

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4[64]; // [rsp+20h] [rbp-70h]
  char v5; // [rsp+60h] [rbp-30h]
  char v6; // [rsp+61h] [rbp-2Fh]
  char v7; // [rsp+62h] [rbp-2Eh]
  char v8; // [rsp+63h] [rbp-2Dh]
  char v9; // [rsp+64h] [rbp-2Ch]
  char v10; // [rsp+65h] [rbp-2Bh]
  char v11; // [rsp+66h] [rbp-2Ah]
  char v12; // [rsp+67h] [rbp-29h]
  char v13; // [rsp+68h] [rbp-28h]
  char v14; // [rsp+69h] [rbp-27h]
  char v15; // [rsp+6Ah] [rbp-26h]
  char v16; // [rsp+6Bh] [rbp-25h]
  char v17; // [rsp+6Ch] [rbp-24h]
  char v18; // [rsp+6Dh] [rbp-23h]
  char v19; // [rsp+6Eh] [rbp-22h]
  char v20; // [rsp+6Fh] [rbp-21h]
  char v21; // [rsp+70h] [rbp-20h]
  char v22; // [rsp+71h] [rbp-1Fh]
  char v23; // [rsp+72h] [rbp-1Eh]
  char v24; // [rsp+73h] [rbp-1Dh]
  char v25; // [rsp+74h] [rbp-1Ch]
  char v26; // [rsp+75h] [rbp-1Bh]
  char v27; // [rsp+76h] [rbp-1Ah]
  char v28; // [rsp+77h] [rbp-19h]
  char v29; // [rsp+78h] [rbp-18h]
  char v30; // [rsp+79h] [rbp-17h]
  char v31; // [rsp+7Ah] [rbp-16h]
  char v32; // [rsp+7Bh] [rbp-15h]
  char v33; // [rsp+7Ch] [rbp-14h]
  char v34; // [rsp+7Dh] [rbp-13h]
  int v35; // [rsp+88h] [rbp-8h]
  int v36; // [rsp+8Ch] [rbp-4h]

  _main(argc, argv, envp);
  v36 = 0;
  v35 = 0;
  v5 = 126;
  v6 = 53;
  v7 = 11;
  v8 = 42;
  v9 = 39;
  v10 = 44;
  v11 = 51;
  v12 = 31;
  v13 = 118;
  v14 = 55;
  v15 = 27;
  v16 = 114;
  v17 = 49;
  v18 = 30;
  v19 = 54;
  v20 = 12;
  v21 = 76;
  v22 = 68;
  v23 = 99;
  v24 = 114;
  v25 = 87;
  v26 = 73;
  v27 = 8;
  v28 = 69;
  v29 = 66;
  v30 = 1;
  v31 = 90;
  v32 = 4;
  v33 = 19;
  v34 = 76;
  printf(&Format);
  scanf("%s", v4);
  while ( v4[v36] )
  {
    v4[v36] ^= 78 - (_BYTE)v36;
    ++v36;
  }
  while ( v35 < v36 )
  {
    if ( v4[v35] != (unsigned __int8)*(&v5 + v35) || v36 != 30 )
    {
      printf(asc_404022);
      system("pause");
      exit(0);
    }
    ++v35;
  }
  puts(Str);
  system("pause");
  return 0;
}
```

先读码，v5到v34应为长度为30的数组

代码大意是输入的30位字符串（即flag）分别与78、77、76……做异或操作与126、53…76比较

由于异或可逆，只要将126、53…76与78、77、76……异或就能得到flag的Ascii码

这里用python来算

```
// Xor
c = 126 ^ 78;
printf("%c", c);
c = 53 ^ 77;
printf("%c", c);
c = 11 ^ 76;
printf("%c", c);
c = 42 ^ 75;
printf("%c", c);
c = 39 ^ 74;
```

```
printf("%c", c);
c = 44 ^ 73;
printf("%c", c);
c = 51 ^ 72;
printf("%c", c);
c = 31 ^ 71;
printf("%c", c);
c = 118 ^ 70;
printf("%c", c);
c = 55 ^ 69;
printf("%c", c);
c = 27 ^ 68;
printf("%c", c);
c = 114 ^ 67;
printf("%c", c);
c = 49 ^ 66;
printf("%c", c);
c = 30 ^ 65;
printf("%c", c);
c = 54 ^ 64;
printf("%c", c);
c = 12 ^ 63;
printf("%c", c);
c = 76 ^ 62;
printf("%c", c);
c = 68 ^ 61;
printf("%c", c);
c = 99 ^ 60;
printf("%c", c);
c = 114 ^ 59;
printf("%c", c);
c = 87 ^ 58;
printf("%c", c);
c = 73 ^ 57;
printf("%c", c);
c = 8 ^ 56;
printf("%c", c);
c = 69 ^ 55;
printf("%c", c);
c = 66 ^ 54;
printf("%c", c);
c = 1 ^ 53;
printf("%c", c);
c = 90 ^ 52;
printf("%c", c);
c = 4 ^ 51;
printf("%c", c);
c = 19 ^ 50;
printf("%c", c);
c = 76 ^ 49;
printf("%c", c);
```

哈哈，最后10分钟做的比较急，写的繁琐一点

最后算出 `0xGame{X0r_1s_v3ry_Imp0rt4n7!}`