Week1_萱_B25041708_wp

Web

1.Lemon

右键和 F12 都不管用

那只能请出 ctrlU 大人

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>lemon</title>
    <script>
        document.addEventListener('contextmenu', function(e) {
            alert('如果右键跟酸与涩,都只是梦一重');
            e.preventDefault();
        });
        document.addEventListener('keydown', function(e) {
            if (e.key === 'F12') {
                alert('也许我,仍会选择,某个FLAG,再跟你重逢');
                e.preventDefault();
            }
        });
    </script>
</head>
<body>
<p>
    那一天的源神 启动起来~
</p>
</body>
</html>

<!-- 0xGame{Welc0me_t0_0xG@me_2025_Web!!!} -->
```

Flag：0xGame{Welc0me_t0_0xG@me_2025_Web!!!}

2. Http 的真理，我已解明

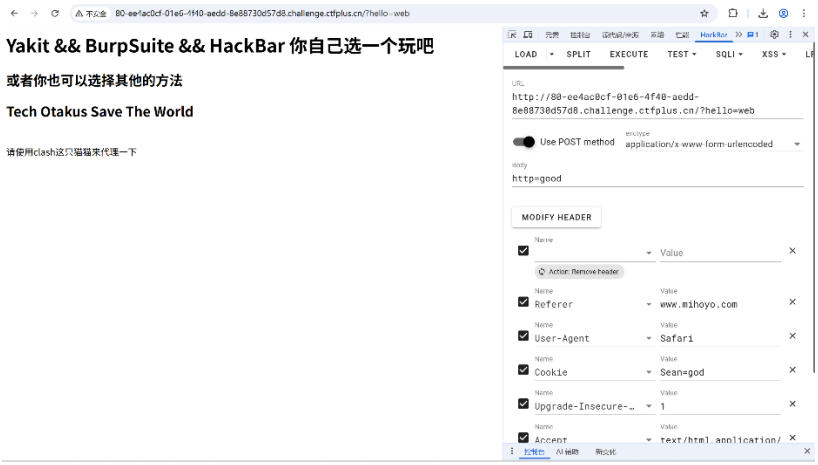
浅浅 get 一下，发现要求 post

那好吧，curl 命令 post 一下，加个 Sean 神的曲奇（？）和苹果浏览器的头

```
C:\Users\xjxua>curl -X POST http://80-ee4ac0cf-01e6-4f40-aedd-8e88730d57d8.challenge.ctfplus.cn/?hello=web -d "http=good
"
<h1>Yakit && BurpSuite && HackBar 你自己选一个玩吧</h1><h2>或者你也可以选择其他的方法</h2><h2>Tech Otakus Save The World
</h2><br>设置cookie Sean=god
C:\Users\xjxua>curl -b "Sean=god" "http://80-ee4ac0cf-01e6-4f40-aedd-8e88730d57d8.challenge.ctfplus.cn/?hello=web"
<h1>Yakit && BurpSuite && HackBar 你自己选一个玩吧</h1><h2>或者你也可以选择其他的方法</h2><h2>Tech Otakus Save The World
</h2><br>用POST传递 http=good
C:\Users\xjxua>curl -b "Sean=god" -X POST http://80-ee4ac0cf-01e6-4f40-aedd-8e88730d57d8.challenge.ctfplus.cn/?hello=web
 -d "http=good"
<h1>Yakit && BurpSuite && HackBar 你自己选一个玩吧</h1><h2>或者你也可以选择其他的方法</h2><h2>Tech Otakus Save The World
</h2>请使用Safari浏览器访问
C:\Users\xjxua>
```

```
C:\Users\xjxua>curl  -X POST -b "Sean=god" -A "Safari"  -d "http=good" http://80-ee4ac0cf-01e6-4f40-aedd-8e88730d57d8.ch
allenge.ctfplus.cn/?hello=web
<h1>Yakit && BurpSuite && HackBar 你自己选一个玩吧</h1><h2>或者你也可以选择其他的方法</h2><h2>Tech Otakus Save The World
</h2><br>请从www.mihoyo.com访问本页面，否则你的原石啊这些会都别想要了
```

玩（）玩的，进入 HackBar 保护眼睛



X-Forwarded-For 不行，只能试试 via，发现可以

玩（）（）（）（）玩的（bushi）

Flag: 0XGame{Congratuation_You_Are_Http_God!!!}

Misc

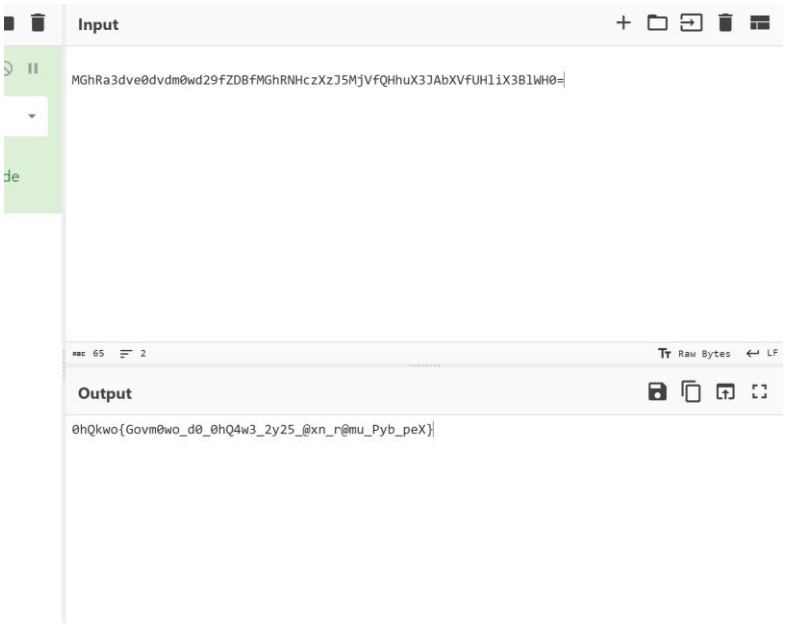
1.签到-0xGame

扫码关注一波，然后获得 flag

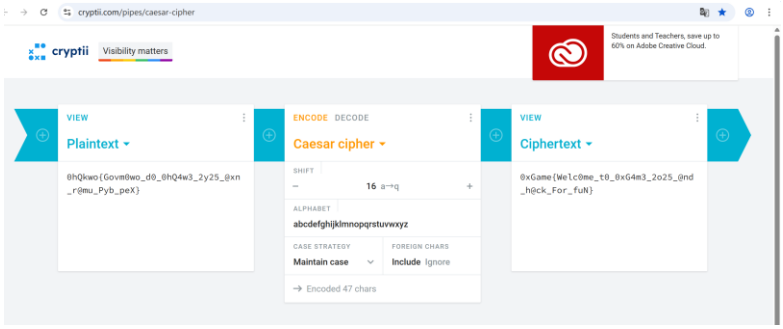
Flag: 0xGame{🎉👋🕹️2️⃣0️⃣2️⃣5️⃣0️⃣❎🎮🎯🏟️🥳🎊⚽😄}

2.Sign_in

Base64 转换一下，发现格式正确，内容不对



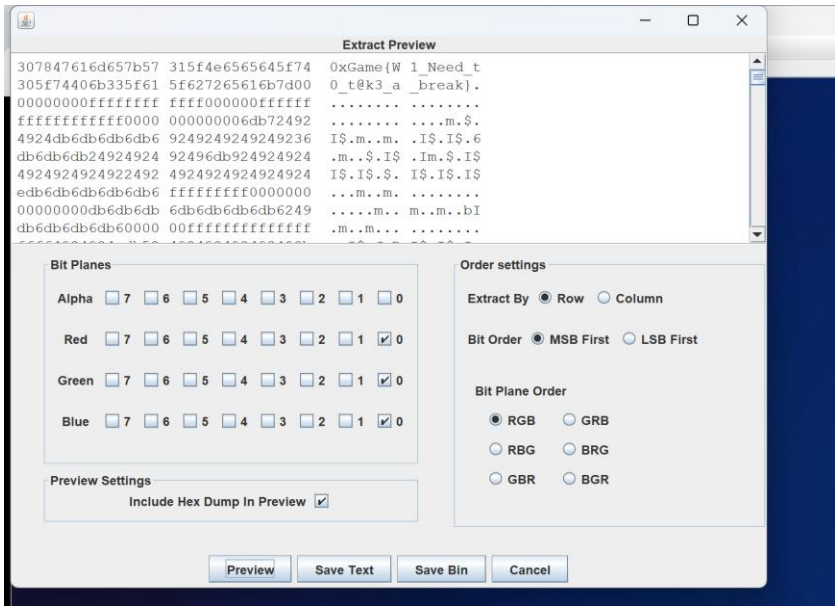
内容不对，就凑个对的，按照 A-Q 浅浅移动一下



Flag：0xGame{Welc0me_t0_0xG4m3_2o25_@nd_h@ck_For_fuN}

3.Zootopia

打开 stegsolve，选上 red0,green0,blue0 分析出这个

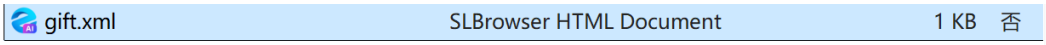


Flag：0xGame{W 1_Need_t0_t@k3_a _break}

4.公众号原稿

压缩一下

翻找 gift ing~

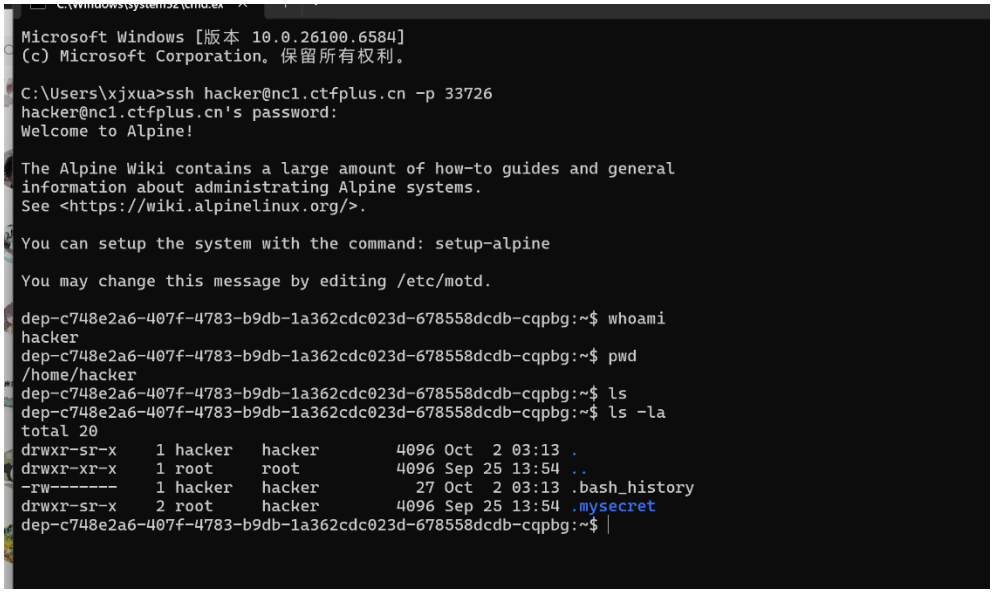


打开 gift OvO，出现网页，浅浅查看一下源代码

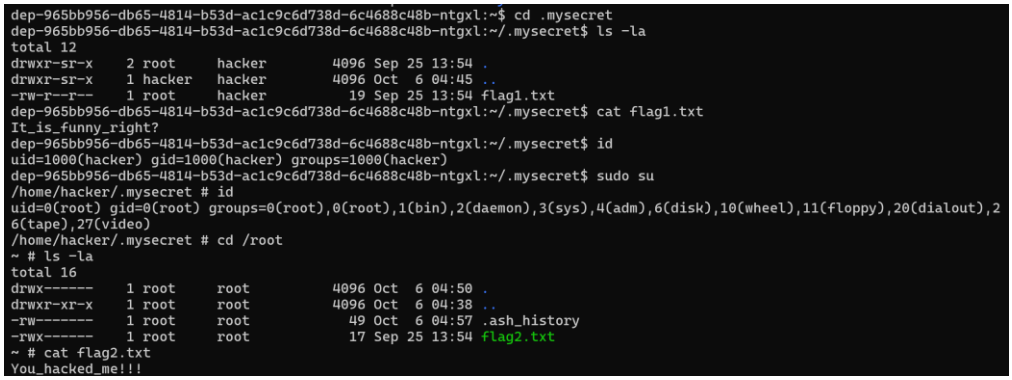
Flag：0xGame{omg!Y0u_f0und_m3!_C0ngr4tul4t10ns!}

5.ez_Shell

没啥说法，跟着步骤走



很详细，孩子很喜欢，下次还会回购



Flag:

0xGame{hacker_/home/hacker_.mysecret_It_is_funny_right?_You_hacked_me!!!}

6.ezShell_PLUS

如图，发现加密文本

```
C:\Users\xjxua>ssh welcome@nc1.ctfplus.cn -p 39132
welcome@nc1.ctfplus.cn's password:
Last login: Mon Oct  6 08:58:29 2025 from 10.10.0.19
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~$ ls -la
total 28
drwxr-x--- 1 welcome welcome 4096 Oct  6 08:57 .
drwxr-xr-x 1 root    root    4096 Sep 30 12:24 ..
-rw-r--r-- 1 welcome welcome  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 welcome welcome 3771 Jan  6  2022 .bashrc
-rw-r--r-- 1 welcome welcome  807 Jan  6  2022 .profile
drwxr-x--- 3 root    welcome 4096 Oct  6 08:57 challenge
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~$ cd challenge
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$ ls -la
total 20
drwxr-x--- 3 root    welcome 4096 Oct  6 08:57 .
drwxr-x--- 1 welcome welcome 4096 Oct  6 08:57 ..
-rwxr-x--- 1 root    welcome  271 Oct  6 08:57 decrypt.sh
drwxr-x--- 2 root    welcome 4096 Oct  6 08:57 files
-rw-r----- 1 root    welcome   65 Oct  6 08:57 hash_value
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$ cat hash_value
1cac12fd8640ef96caf3890fac8821dd235272f3100a0be8b22492f0f4fc15fd
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$
```

找到加密文本路径（眼睛差点看瞎 orz）

```
C:\Users\xjxua>ssh welcome@nc1.ctfplus.cn -p 39132
welcome@nc1.ctfplus.cn's password:
Last login: Mon Oct  6 09:51:42 2025 from 10.10.0.19
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~$ ls -la
total 32
drwxr-x--- 1 welcome welcome 4096 Oct  6 10:26 .
drwxr-xr-x 1 root    root    4096 Sep 30 12:24 ..
-rw------- 1 welcome welcome   72 Oct  6 10:26 .bash_history
-rw-r--r-- 1 welcome welcome  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 welcome welcome 3771 Jan  6  2022 .bashrc
-rw-r--r-- 1 welcome welcome  807 Jan  6  2022 .profile
drwxr-x--- 3 root    welcome 4096 Oct  6 08:57 challenge
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~$ cd challenge
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$ ls -la
total 20
drwxr-x--- 3 root    welcome 4096 Oct  6 08:57 .
drwxr-x--- 1 welcome welcome 4096 Oct  6 10:26 ..
-rwxr-x--- 1 root    welcome  271 Oct  6 08:57 decrypt.sh
drwxr-x--- 2 root    welcome 4096 Oct  6 08:57 files
-rw-r----- 1 root    welcome   65 Oct  6 08:57 hash_value
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$ cat hash_value
1cac12fd8640ef96caf3890fac8821dd235272f3100a0be8b22492f0f4fc15fd
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$ sha256sum files/*
528d581dd531e5df8f6fe7ffabefb7b2f5295240de311f9763d6638d4075df4f  files/05118baf39170b62.dat
cdb02c832da670f9c3aa5c77bbbcd2c4bf87ce750027570d95a0334e628fc18d  files/0579a7858b09d581.dat
e438f11716c0d14d68c07cfe53294ffc54931844c034e80127f446ea432b5847  files/077c89cfc337b067.dat
8bf7e25020eddd292a4a0bf41536835ebfbf1a800b7b6c149d8d7a2d57c84c60  files/0c766f00bc610517.dat
1cac12fd8640ef96caf3890fac8821dd235272f3100a0be8b22492f0f4fc15fd  files/0ff92d4b56263c81.dat
8f6329fb523b094cc4aae0d9674bf3a61dfe0430bda5c5b33f9d678f2a61dbbe  files/12aa0bf9b237f570.dat
593cb19879a5acd4582d354a11a61c9f7cf7607aa56fc29b1419b7028132b0d2  files/137f68078d1a581e.dat
```

跑一下脚本

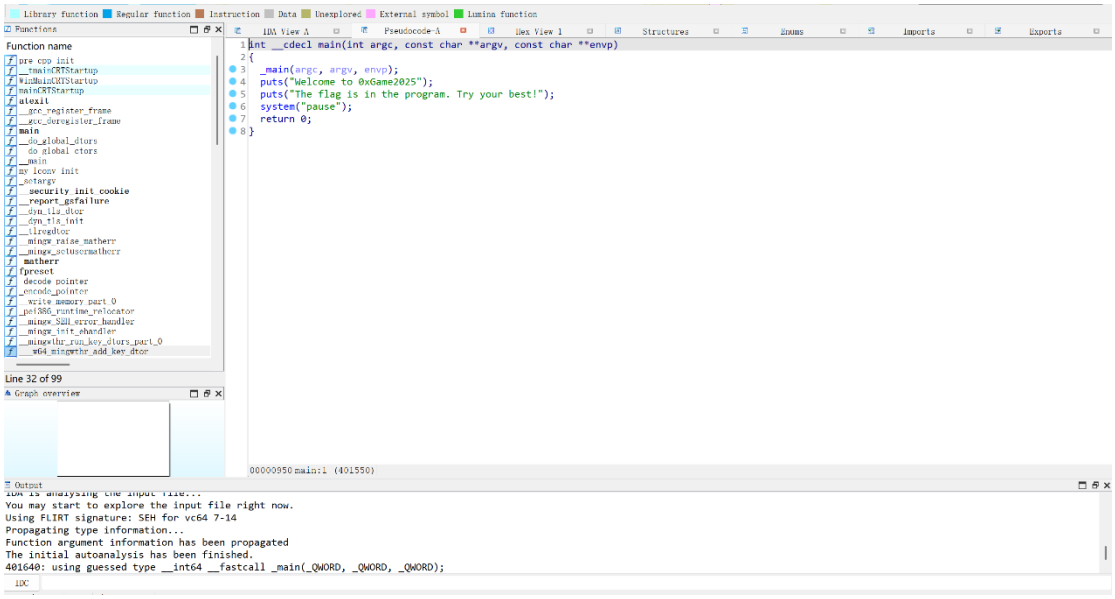
```
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$ ./decrypt.sh files/0ff92d4b56263c81.dat
0xGame{Welc0me_to_H@ckers_w0rld}
welcome@dep-0d51c320-01b1-4e31-a0e1-8de19f3cb03c-78549d8c47-xz77t:~/challenge$
```

Flag：0xGame{Welc0me_to_H@ckers_w0rld}

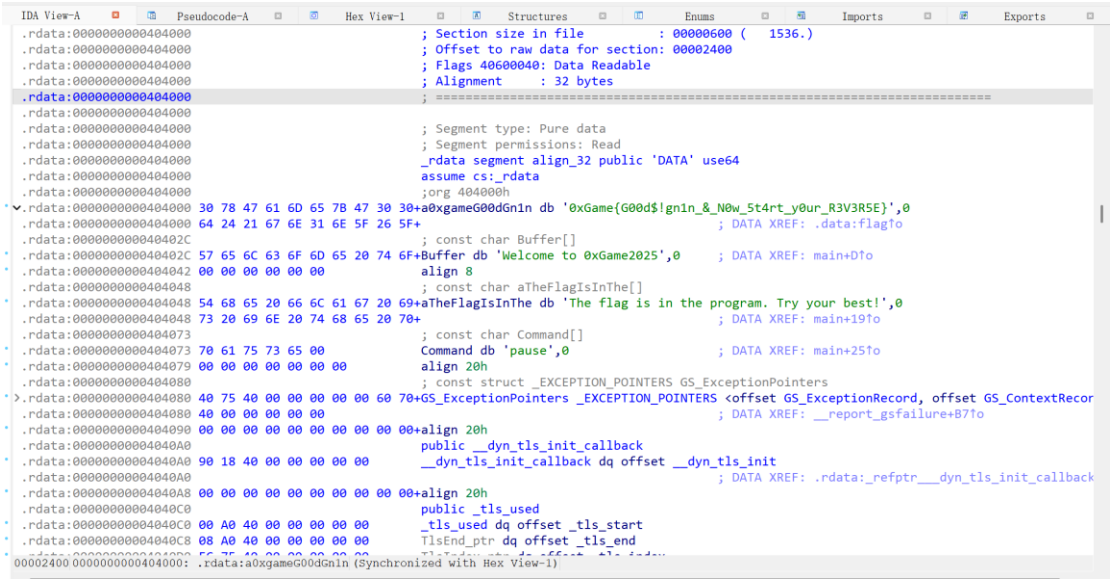
Reverse

1.SignIn

打开 IDA，导入文件 F5 一下



双击 0xGame 一行，就有 flag



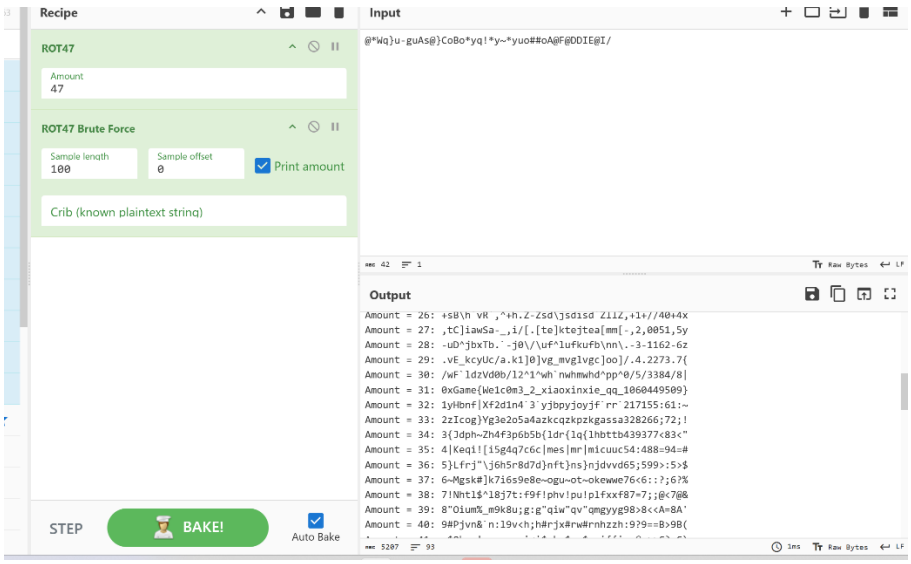
Flag：0xGame{G00d$!gn1n_&_N0w_5t4rt_y0ur_R3V3R5E}

2.SignIn2

打开程序，显示需要 ROT47 解密

```
It is recommended that you switch the console encoding to UTF-8.

刚刚我把你的控制台编码换成utf-8了，现在你应该可以正常看utf-8编码的中文字符了。
其实是我怕你看不到我打的广告了，哈哈

这是加密后的flag:
@*Wq}u-guAs@}CoBo*yq!*y~*yuo##oA@F@DDIE@I/
请输入一个整数作为key来解密：
6
解密后的flag: :$Qkwo'ao;m:w=i<i$sky$sx$soi{{i;:@:>>C?:C)
好像不太对捏，给你一点提示吧

ROT47 Brust Force
```

那还说啥了，解吧解吧，look at 31

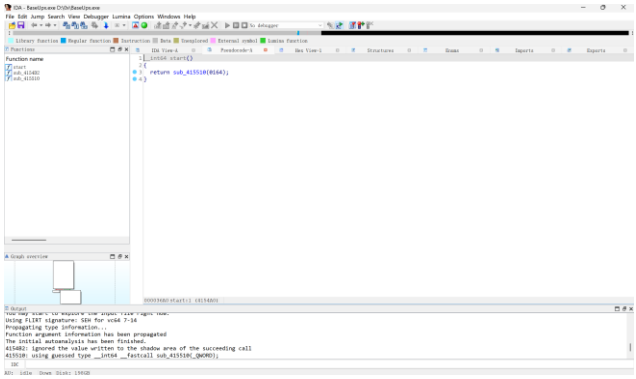


Flag：0xGame{We1c0m3_2_xiaoxinxie_qq_1060449509}

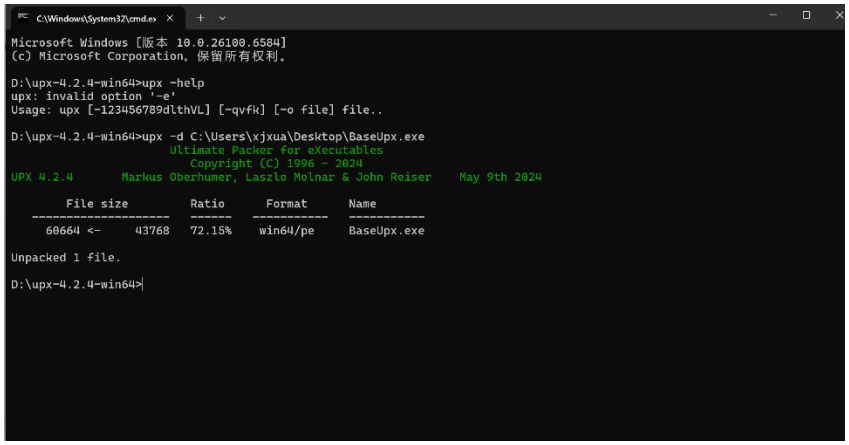
（这个 wp 是后面写的，但怎么感觉和当时解出来的不太一样……？）

3.BaseUpx

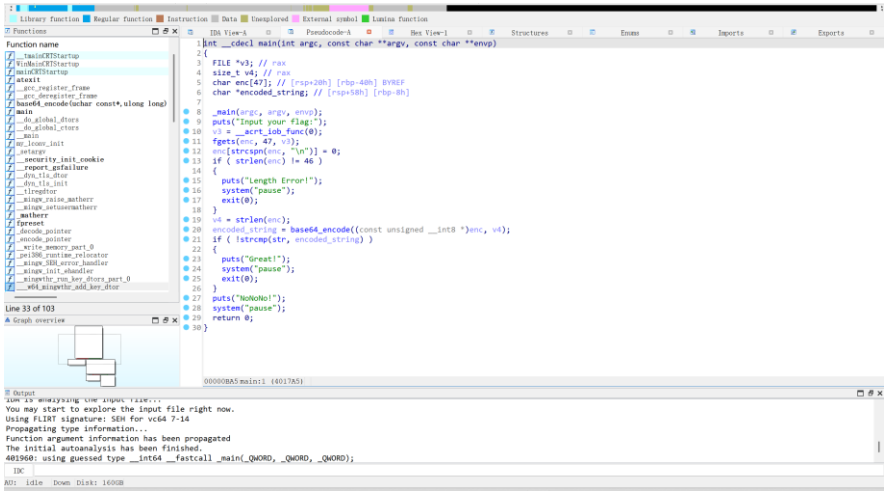
直接 IDA 打开的话什么都没有



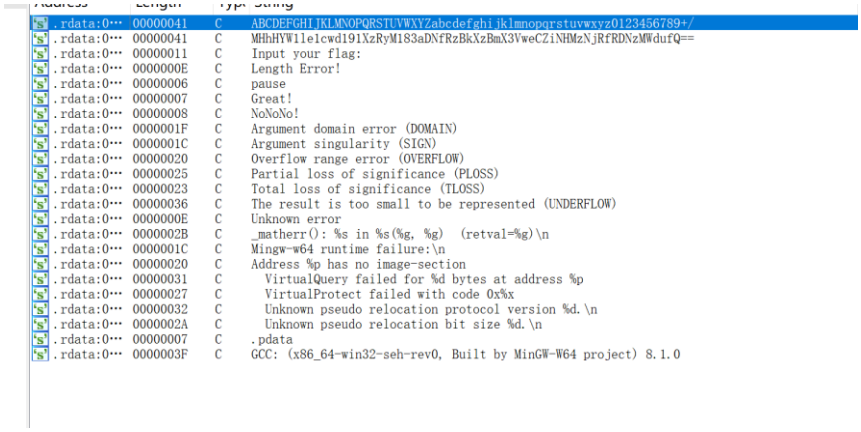
根据题目来看是有壳的，那还说啥，脱了呗

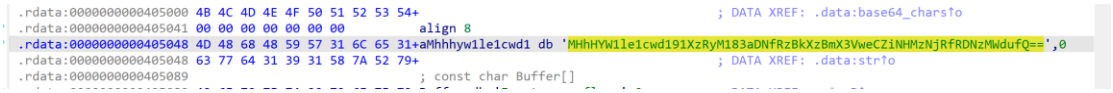


脱壳后 IDA 打开就有东西了

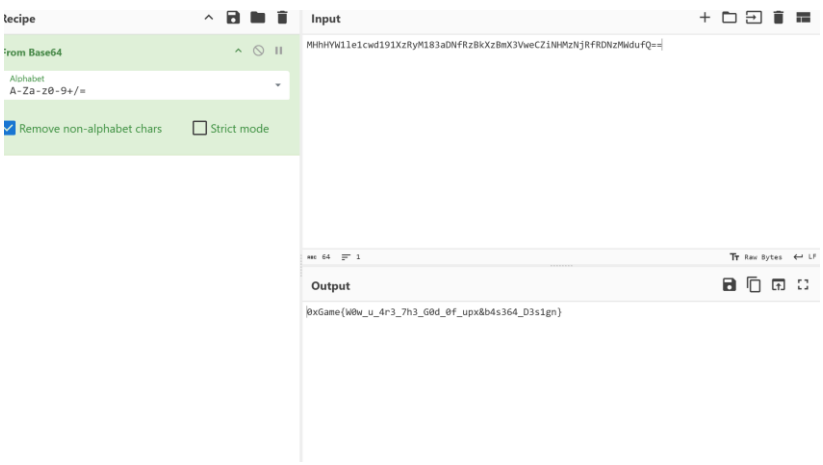


ShiftF12 一下，出现一个 base64 编码





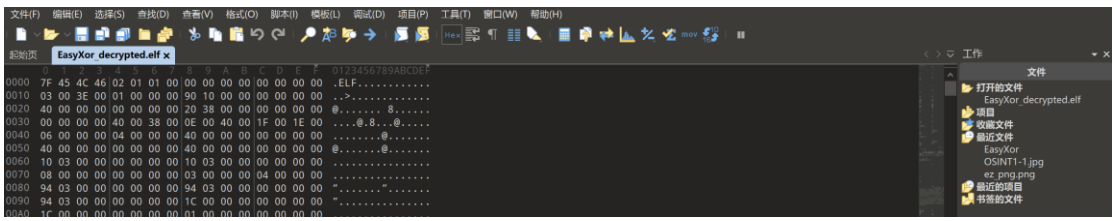
丢进去解码一下



Flag：0xGame{W0w_u_4r3_7h3_G0d_0f_upx&b4s364_D3s1gn}

4.EasyXor

010editor 打开发现文件对应的是 LinuxELF



中间大量无意义片段，有一段明文

```
0  ................
0  ................
0  ................
8  ........raputa0x
0  Game2025........
5  Do you know abou
4  t bitwise operat
F  ions? They're co
C  mmon in reverse,
E   especially XOR.
5  .Now please give
A   me your flag:..
9  .Please try agai
0  n!.Good job!....
F  ...;,.......pïÿÿ
F  x...Ðïÿÿ ...àïÿÿ
0  H...Éðÿÿ¸.......
1  .........zR..x..
0  ................
```

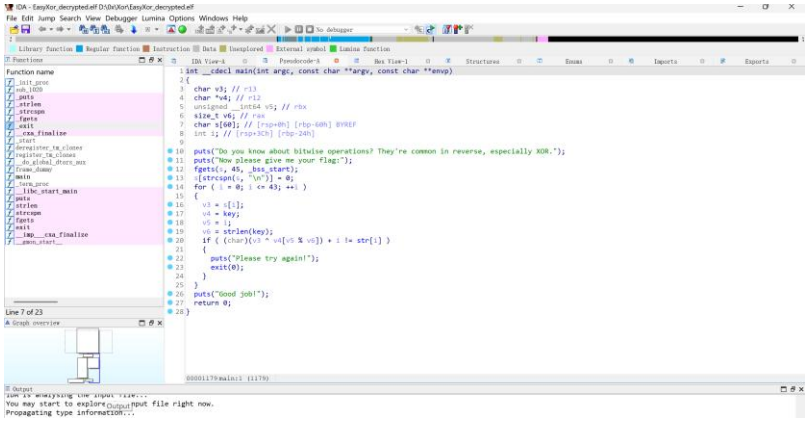
上一个脚本找到 elf 文件

```python
import os

def analyze_and_decrypt(filename):
    # 读取文件
    with open(filename, 'rb') as f:
        data = f.read()

    print(f"文件大小: {len(data)} 字节")
    print(f"文件头: {data[:16].hex()}")

    # 标准ELF头应该是: 7f 45 4c 46
    elf_header = b'\x7fELF'

    # 推导可能的密钥
    potential_key_bytes = []
    for i in range(4):
        potential_key_bytes.append(data[i] ^ elf_header[i])

    print(f"推导的密钥前4字节: {[hex(b) for b in potential_key_bytes]}")

    # 尝试不同密钥长度
    for key_len in [1, 2, 4, 8, 16, len("raputa0xGame2025")]:
        if key_len == len("raputa0xGame2025"):
            key = "raputa0xGame2025".encode()
        else:
            key = bytes(potential_key_bytes[:key_len])
```

```python
        decrypted = bytearray()
        for i in range(len(data)):
            decrypted.append(data[i] ^ key[i % len(key)])

        decrypted_bytes = bytes(decrypted)

        # 检查是否为有效ELF
        if decrypted_bytes.startswith(b'\x7fELF'):
            print(f"成功解密！密钥长度: {key_len}")
            if key_len == len("raputa0xGame2025"):
                print(f"密钥: raputa0xGame2025")
            else:
                print(f"密钥: {key.hex()}")

            # 保存解密文件
            output_path = os.path.splitext(filename)[0] + "_decrypted.elf"
            with open(output_path, 'wb') as f:
                f.write(decrypted_bytes)
            print(f"已保存到: {output_path}")

            # 检查文件中是否包含flag提示
            if b'flag' in decrypted_bytes.lower():
                print("文件中包含flag相关字符串")

            return True
```

```
    return False

# 使用你的文件路径
file_path = r"D:\0x\Xor\EasyXor"
analyze_and_decrypt(file_path)
```

输出文件用 IDA 打开



那么 key 是 raputa0xGame2025

```
.data:0000000000004090                          public key
.data:0000000000004090                          ; char *key
.data:0000000000004090 08 20 00 00 00 00 00 00  key dq offset aRaputa0xgame20      ; DATA XREF: main+75↑r
.data:0000000000004090                                                            ; main+82↑r
.data:0000000000004090                          _data ends                        ; "raputa0xGame2025"
.data:0000000000004090
```

Str 数组

```
30000004060                             public str
30000004060                             ; unsigned __int8 str[48]
30000004060 42 1A 39 17 1D 09 51 55 2C 5F+str db 42h, 1Ah, 39h, 17h, 1Dh, 9, 51h, 55h, 2Ch, 5Fh, 63h, 0Ch, 0Dh, 16h, 62h, 27h, 55h, 64h, 55h, 26h
30000004060 63 0C 0D 16 62 27 55 64 55 26+                                ; DATA XREF: main+BC↑o
30000004060 6D 6A 18 34 88 65 6E 1C 21 6E+db 6Dh, 6Ah, 18h, 34h, 88h, 65h, 6Eh, 1Ch, 21h, 6Eh, 3Dh, 23h, 6Ah, 25h, 6Bh, 63h, 68h, 7Eh, 77h, 75h
30000004060 3D 23 6A 25 6B 63 68 7E 77 75+db 9Ah, 7Dh, 39h, 43h, 4 dup(0)
30000004090                             public key
30000004090                             ; char *key
```

脚本再+1

```
def decrypt_flag():
    key = b"raputa0xGame2025"
    key_len = len(key)

    # str数组的44个字节（从IDA中提取）
    str_data = [
        0x42, 0x1A, 0x39, 0x17, 0x1D, 0x09, 0x51, 0x55, 0x2C, 0x5F,
        0x63, 0x0C, 0x0D, 0x16, 0x62, 0x27, 0x55, 0x64, 0x55, 0x26,
        0x6D, 0x6A, 0x18, 0x34, 0x88, 0x65, 0x6E, 0x1C, 0x21, 0x6E,
        0x3D, 0x23, 0x6A, 0x25, 0x6B, 0x63, 0x68, 0x7E, 0x77, 0x75,
        0x9A, 0x7D, 0x39, 0x43
    ]

    flag = ""
    for i in range(44):
        # 反向计算: flag[i] = (str[i] - i) XOR key[i % key_len]
        flag_char = (str_data[i] - i) ^ key[i % key_len]
        flag += chr(flag_char & 0xFF)  # 确保是有效字节

    return flag

# 计算并打印flag
flag = decrypt_flag()
print(f"Flag: {flag}")
```

输出

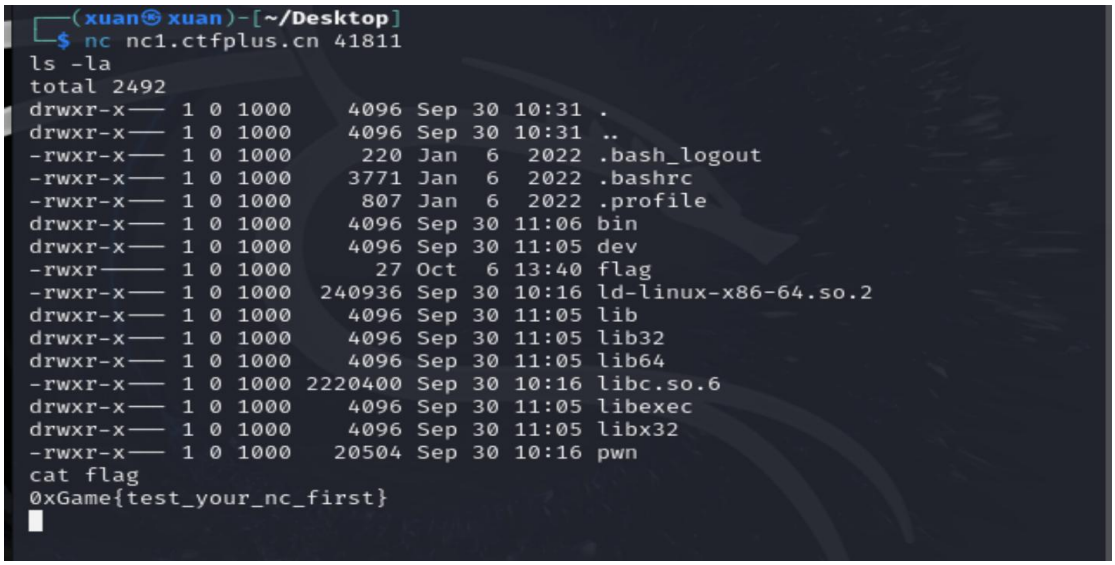
```
Flag: 0xGame{6c74d39f-723f-42e7-9d7a-18e9508a655b}
```

flag：0xGame{6c74d39f-723f-42e7-9d7a-18e9508a655b}

Pwn

1.test_your_nc

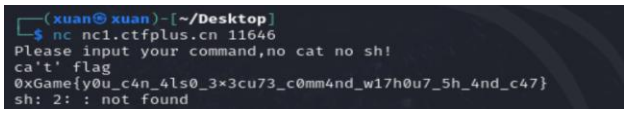
如图



Flag：0xGame{test_your_nc_first}

2.命令执行🤔

Cat 用不了，管我 ca’t’什么事 OvO



Flag：0xGame{y0u_c4n_4ls0_3x3cu73_c0mm4nd_w17h0u7_5h_4nd_c47}

Crypto

1.Vigenere

脚本一位~

```python
from string import digits, ascii_letters, punctuation

key = "Welcome-2025-0xGame"
alphabet = digits + ascii_letters + punctuation

ciphertext = 'WL"mKAaequ{q_aY$oz8`wBqLAF_{cku|eYAczt!pmoqAh+'

def vigenere_decrypt(ciphertext, key):
    plaintext = ""
    key_index = 0
    for char in ciphertext:
        bias = alphabet.index(key[key_index])
        char_index = alphabet.index(char)
        new_index = (char_index - bias) % len(alphabet)
        plaintext += alphabet[new_index]
        key_index = (key_index + 1) % len(key)
    return plaintext

print(vigenere_decrypt(ciphertext, key))
```

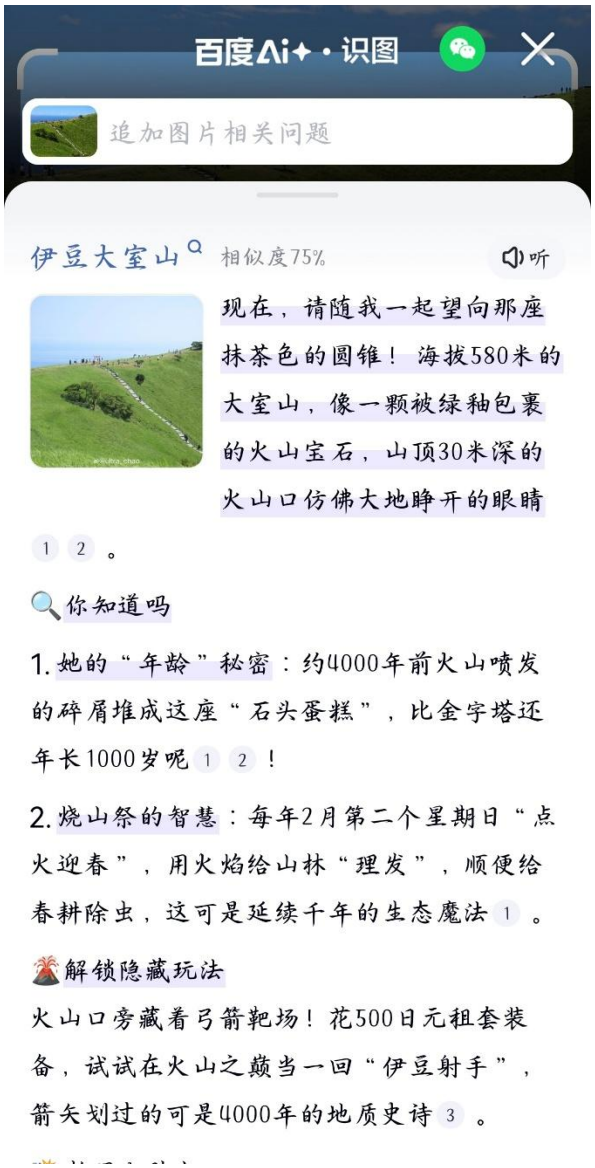
```
0xGame{you_learned_vigenere_cipher_2df4b1c2e3}
```

Flag：0xGame{you_learned_vigenere_cipher_2df4b1c2e3}

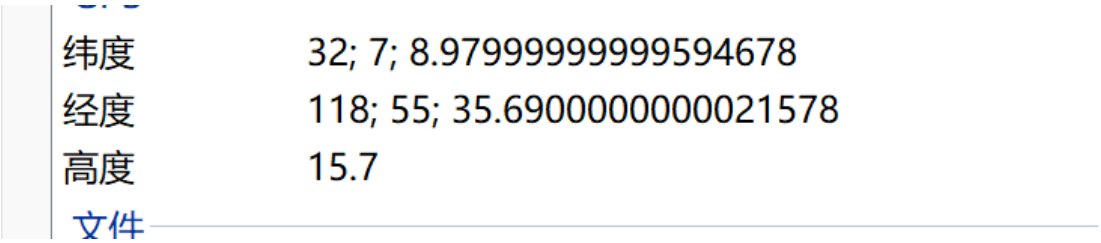
Osint

1.猜猜 background

打开百度识图，得到大室山（伊豆是地名!）



查看一下第二张图属性



换算一下可得经纬度

Flag：0xGame{大室山_32.1191_118.9265}