

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=good" -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>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.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>请从 www.mihooyo.com 访问本页面，否则你的原石崩这些全都别想要了
C:\Users\xjxua>
```

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)

Yakit && BurpSuite && HackBar 你自己选一个玩吧

或者你也可以选择其他的方法

Tech Otakus Save The World

0XGame{Congratuation_You_Are_Http_God!!!}

HTTP协议的真理,你已解明!

Flag: 0XGame{Congratuation_You_Are_Http_God!!!}

Misc

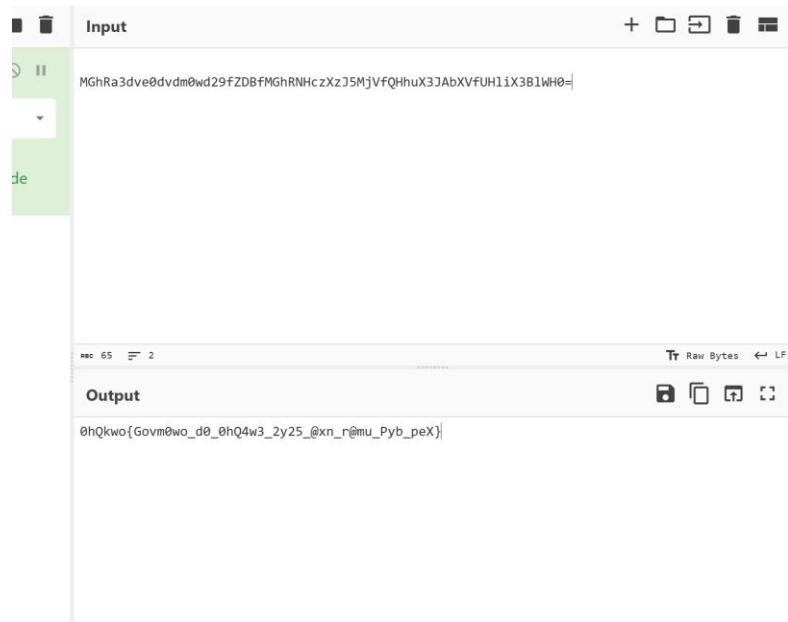
1. 签到-0xGame

扫码关注一波，然后获得 flag

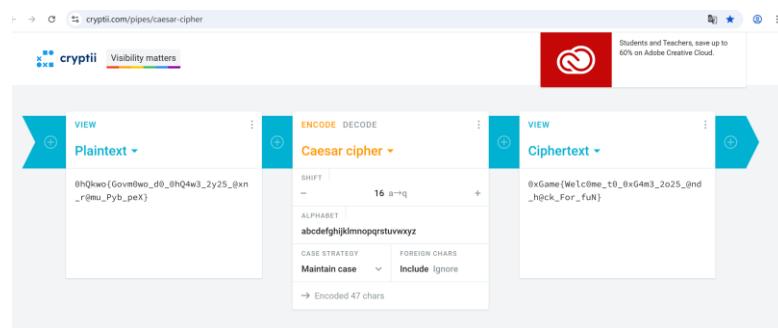
Flag: 0xGame{🎉👋🎮 2 0 2 5 0 ✖️ 🎮🎯🎉🎨⚽️ 😊 }

2.Sign_in

Base64 转换一下，发现格式正确，内容不对



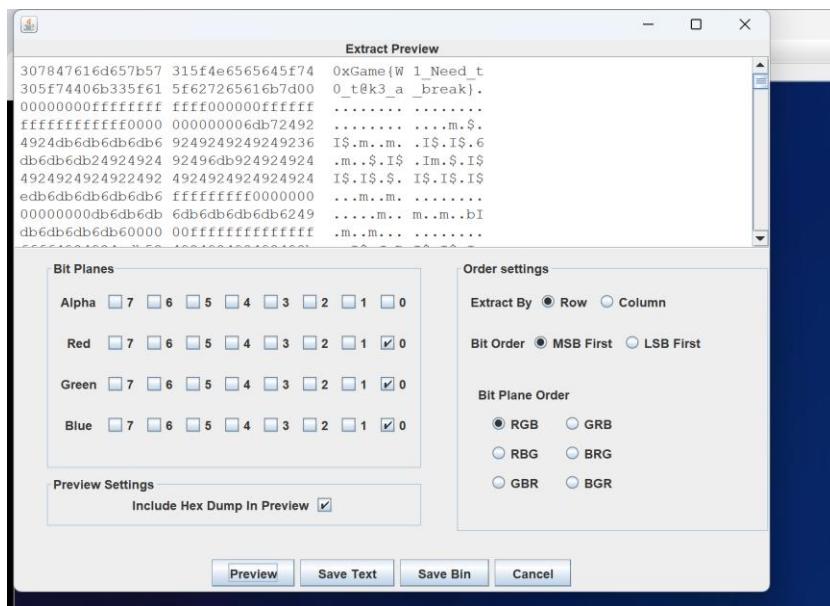
内容不对，就凑个对的，按照 A-Q 浅浅移动一下



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

公众号.zip 2025/10/1 20:19 压缩(zipped)文件夹

翻找 gift ing~

gift.xml SLBrowser HTML Document 1 KB 否

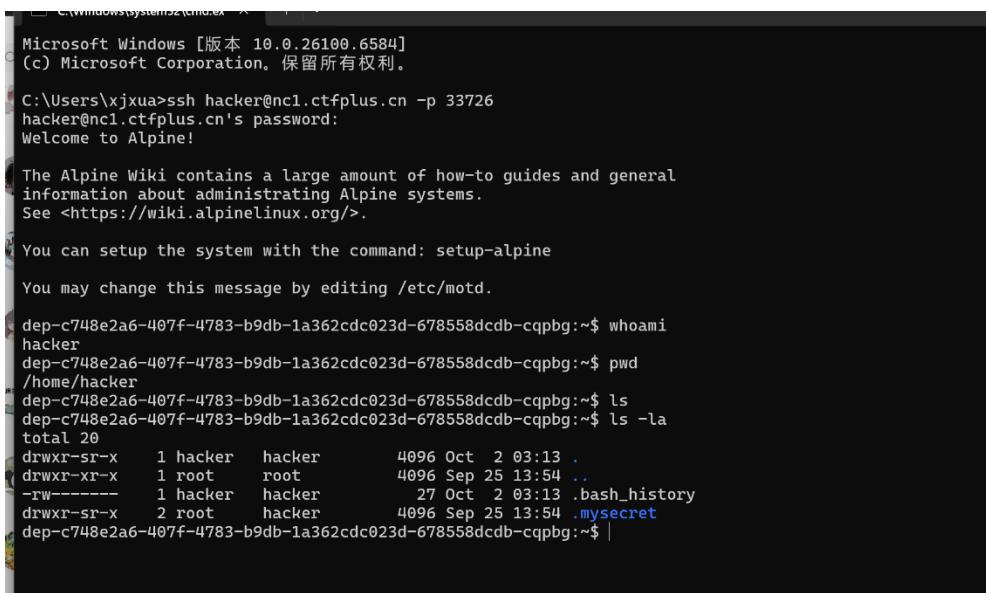
打开 gift OvO，出现网页，浅浅查看一下源代码



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

5.ez_Shell

没啥说法，跟着步骤走



```
Microsoft Windows [版本 10.0.26100.6584]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\xjxua>ssh hacker@nc1.ctfplus.cn -p 33726
hacker@nc1.ctfplus.cn's password:
Welcome to Alpine!

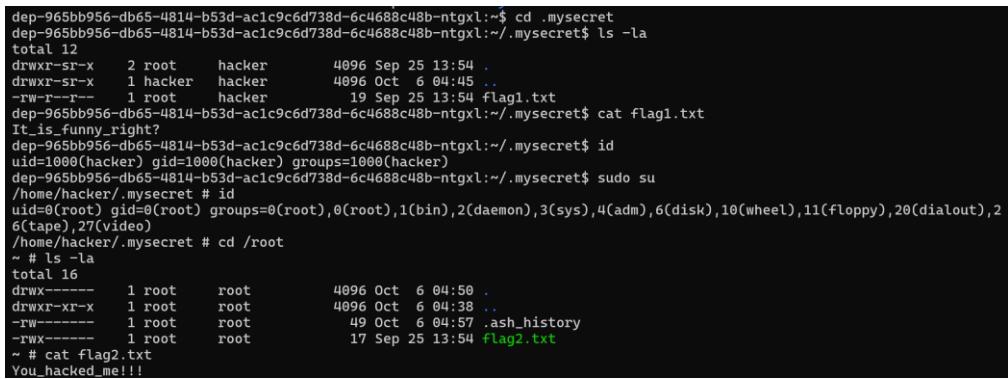
The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

dep-c748e2a6-407f-4783-b9db-1a362cdc023d-678558dcdb-cqpb:~$ whoami
hacker
dep-c748e2a6-407f-4783-b9db-1a362cdc023d-678558dcdb-cqpb:~$ pwd
/home/hacker
dep-c748e2a6-407f-4783-b9db-1a362cdc023d-678558dcdb-cqpb:~$ ls
dep-c748e2a6-407f-4783-b9db-1a362cdc023d-678558dcdb-cqpb:~$ ls -la
total 20
drwxr-sr-x  1 hacker  hacker        4096 Oct  2  03:13 .
drwxr-sr-x  1 root    root        4096 Sep 25 13:54 ..
-rw-----  1 hacker  hacker        27 Oct  2  03:13 .bash_history
drwxr-sr-x  2 root    hacker        4096 Sep 25 13:54 .mysecret
dep-c748e2a6-407f-4783-b9db-1a362cdc023d-678558dcdb-cqpb:~$ |
```

很详细，孩子很喜欢，下次还会回购



```
dep-965bb956-db65-4814-b53d-ac1c9c6d738d-6c4688c48b-ntgx1:~$ cd .mysecret
dep-965bb956-db65-4814-b53d-ac1c9c6d738d-6c4688c48b-ntgx1:~/./mysecret$ ls -la
total 12
drwxr-sr-x  2 root    hacker        4096 Sep 25 13:54 .
drwxr-sr-x  1 hacker  hacker        4096 Oct  6  04:45 ..
-rw-r--r--  1 root    hacker        19 Sep 25 13:54 flag1.txt
dep-965bb956-db65-4814-b53d-ac1c9c6d738d-6c4688c48b-ntgx1:~/./mysecret$ cat flag1.txt
It_is_funny_right?
dep-965bb956-db65-4814-b53d-ac1c9c6d738d-6c4688c48b-ntgx1:~/./mysecret$ id
uid=1000(hacker) gid=1000(hacker) groups=1000(hacker)
dep-965bb956-db65-4814-b53d-ac1c9c6d738d-6c4688c48b-ntgx1:~/./mysecret$ sudo su
/home/hacker/.mysecret # id
uid=0(root) gid=0(root) groups=0(root),0(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),2
6(tape),27(video)
/home/hacker/.mysecret # cd /root
~ # ls -la
total 16
drwx-----  1 root    root        4096 Oct  6  04:50 .
drwxr-sr-x  1 root    root        4096 Oct  6  04:38 ..
-rw-----  1 root    root        49 Oct  6  04:57 .ash_history
-rw-----  1 root    root        17 Sep 25 13:54 flag2.txt
~ # cat flag2.txt
You_hacked_me!!!
```

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 ..
-rw-r--r--  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-r----- 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 ..
-rw-r----- 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/*
528d581dd531e5df8f6fe7fafefb7d2f5295240de311f9763d6633d4075d4# files/05118ba3f39170b62.dat
cd802c832da670f9c3aa5c77bbcd2c24bf87ce750027570d95a0334e628fc18# files/05794785809d581.dat
e438f11716c0d14d68c07cf53294fffc54931844034e80127f46ea432b5847# files/077c89fcfc337b067.dat
8bf7e25920edd292a4a0bf41536835ebfb1a80007b6c149dd7a2d57c84c6# files/0c766f00bc610517.dat
1cac12fd8640ef96caf3890fac8821dd235272f3100a0be8b22492f0f4fc15fd files/0ff92d4b56263c81.dat
8f6329fb523b094cc4aaed9674bf3a61dfe0430bda5c5b33f9d78f2a61dbbe files/12aa0bf9b237f570.dat
593cb19879a5acd4582d35491a1a61c97cf7607aa56fc29b1419b7028132b0d2 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



Line 32 of 99

00000950 main:1 (401550)

00000950 main:1 (401550)

Output

Analysing the input file...

You may start to explore the input file right now.

Using FLIRT signature: SEH for vc64 7.14

Propagating type information...

Function argument information has been propagated

The initial autoanalysis has been finished.

401640: using guessed type `_int64 __fastcall _main_QWORD, _QWORD, _QWORD`;

ICD

双击 0xGame 一行，就有 flag

IDA View-A Pseudocode-A Hex View-1 Structures Enums Imports Exports

```
.rdata:0000000000404000 ; Section size in file : 00000600 ( 1536.)
.rdata:0000000000404000 ; Offset to raw data for section: 00002400
.rdata:0000000000404000 ; Flags 40600040: Data Readable
.rdata:0000000000404000 ; Alignment : 32 bytes
.rdata:0000000000404000 ; =====
.rdata:0000000000404000 ; Segment type: Pure data
.rdata:0000000000404000 ; Segment permissions: Read
.rdata:0000000000404000 _rdata segment align_32 public 'DATA' use64
.rdata:0000000000404000 assume cs:_rdata
.rdata:0000000000404000 ; orng 404000h
.``rdata:0000000000404000 30 78 47 61 6D 65 7B 47 30 30+0xGameG00dgnIn db '0xGame{G00d$!gnIn & N0w_5t4rt_y0ur_R3V3R5E}',0
.rdata:0000000000404000 64 24 21 67 6E 31 6E 5F 26 5F+ ; DATA XREF: .rdata:flag0
.rdata:000000000040402C ; const char Buffer[]
.rdata:000000000040402C 57 65 6C 63 6F 6D 65 20 74 6F+Buffer db 'Welcome to 0xGame2025',0 ; DATA XREF: main+D10
.rdata:0000000000404042 00 00 00 00 00 00 align 8
.rdata:0000000000404048 ; const char aTheFlagIsInThe[]
.rdata:0000000000404048 54 68 65 20 66 6C 61 67 20 69+TheFlagIsInThe db 'The flag is in the program. Try your best!',0
.rdata:0000000000404048 73 20 69 6E 28 74 68 65 20 70+ ; DATA XREF: main+190
.rdata:0000000000404073 ; const char Command[]
.rdata:0000000000404073 70 61 75 73 65 00 ; DATA XREF: main+250
.rdata:0000000000404079 00 00 00 00 00 00 00 align 20h
.rdata:0000000000404088 ; const struct _EXCEPTION_POINTERS GS_ExceptionPointers
.``rdata:0000000000404088 40 75 40 00 00 00 00 00 00 60 70+GS_ExceptionPointers _EXCEPTION_POINTERS <offset GS_ExceptionRecord, offset GS_ContextRecord>
.rdata:0000000000404088 40 00 00 00 00 00 00 00 00 00 00 00 ; DATA XREF: __report_gsfailure+B70
.rdata:0000000000404090 00 00 00 00 00 00 00 00 00 00 00 00 align 20h
.rdata:00000000004040A0 public __dyn_tls_init_callback
.rdata:00000000004040A0 90 18 40 00 00 00 00 00 00 00 00 00 __dyn_tls_init_callback dq offset __dyn_tls_init
.rdata:00000000004040A0 ; DATA XREF: .rdata:_refptr__dyn_tls_init_callback
.rdata:00000000004040A8 00 00 00 00 00 00 00 00 00 00 00 00+align 20h
.rdata:00000000004040A8 public __tls_used
.rdata:00000000004040C0 ; DATA XREF: __tls_used
.rdata:00000000004040C0 00 A0 40 00 00 00 00 00 00 00 00 00 __tls_start
.rdata:00000000004040C0 08 A0 40 00 00 00 00 00 00 00 00 00 TlsEnd_ptr dq offset __tls_end
.rdata:00000000004040C0 00 00 00 00 00 00 00 00 00 00 00 00 ; DATA XREF: __tls_end
.000002400 0000000000404000: .rdata:0000000000404000 (Synchronized with Hex View-1)
```

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

2.SignIn2

打开程序，显示需要 ROT47 解密

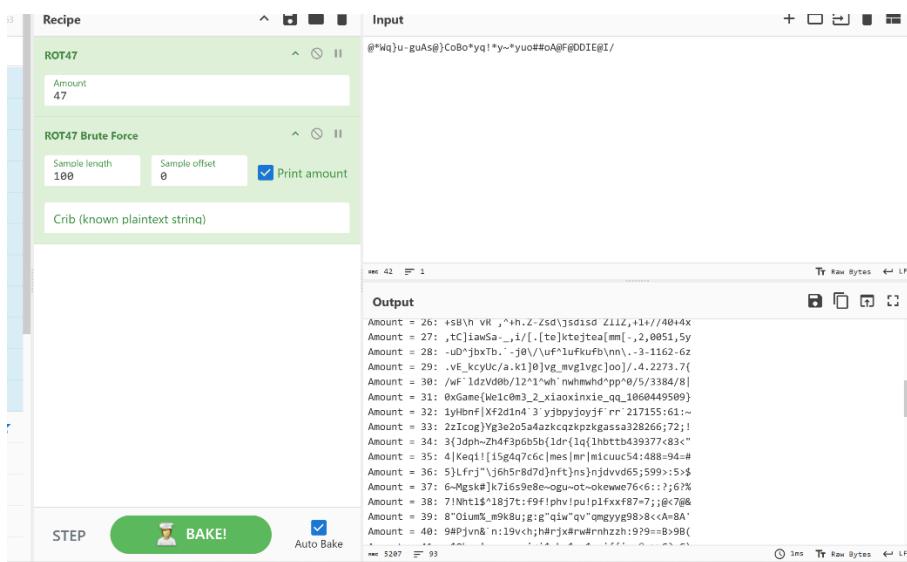
```
It is recommended that you switch the console encoding to UTF-8.

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

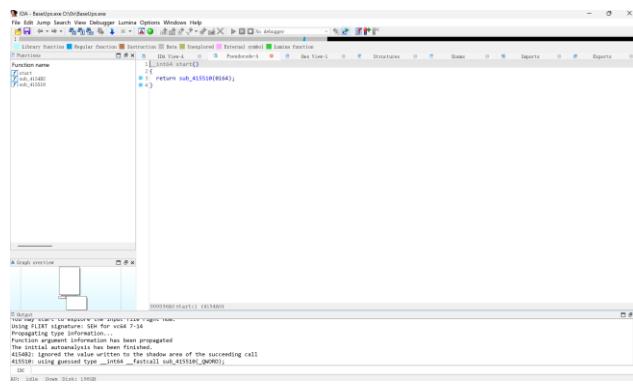


Flag: 0xGame{We1c0m3_2_xiaoxinxie_qq_1060449509}

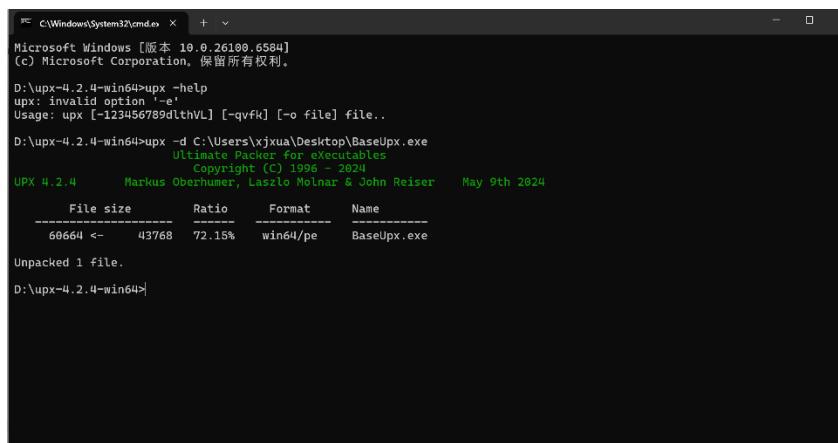
(这个 wp 是后面写的，但怎么感觉和当时解出来的不太一样………?)

3.BaseUpx

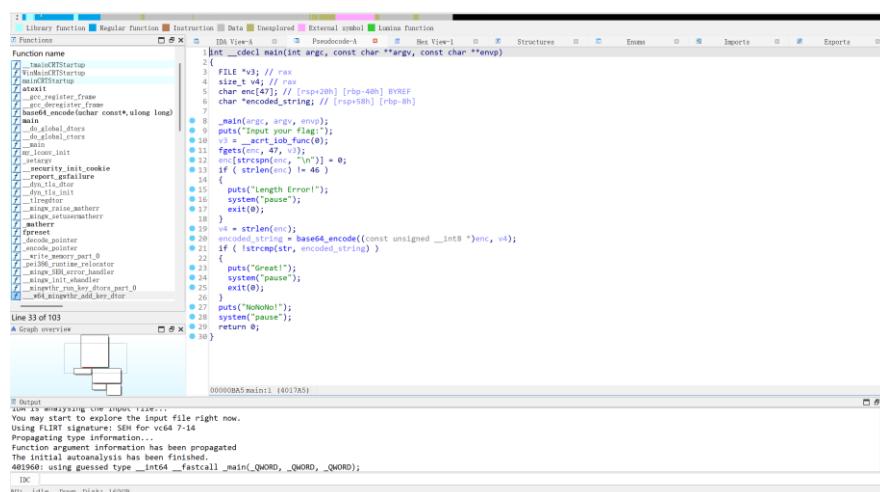
直接 IDA 打开的话什么都没有



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



Shift+F12 一下，出现一个 base64 编码

```

ADDRESS LENGTH TYPE: STRING
.0000000000000041 C ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzijklmnopqrstuvwxyz0123456789+/.
.0000000000000041 C MhHhYw1le1cd191XzRyM183aDNfRzBkXzBmX3VweCZ1NHmzNjRfRDnzMwdufQ==.
.0000000000000011 C Input your flag:.
.0000000000000008 C Length Error!.
.0000000000000006 C pause.
.0000000000000007 C Great!.
.0000000000000008 C NoNoNo!.
.000000000000001F C Argument domain error (DOMAIN).
.000000000000001C C Argument singularity (SIGN).
.0000000000000020 C Overflow range error (OVERFLOW).
.0000000000000025 C Partial loss of significance (LOSS).
.0000000000000023 C Total loss of significance (TLOSS).
.0000000000000036 C The result is too small to be represented (UNDERFLOW).
.000000000000000E C Unknown error.
.000000000000002B C _matherr(): %s in %s(%g, %g) (retval=%g)\n.
.000000000000001C C Mingw-w64 runtime failure:\n.
.0000000000000020 C Address %p has no image-section.
.0000000000000031 C VirtualQuery failed for %d bytes at address %p.
.0000000000000027 C VirtualProtect failed with code 0x%.
.0000000000000032 C Unknown pseudo relocation protocol version %d.\n.
.000000000000002A C Unknown pseudo relocation bit size %d.\n.
.0000000000000007 C .pdata.
.000000000000003F C GCC: (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

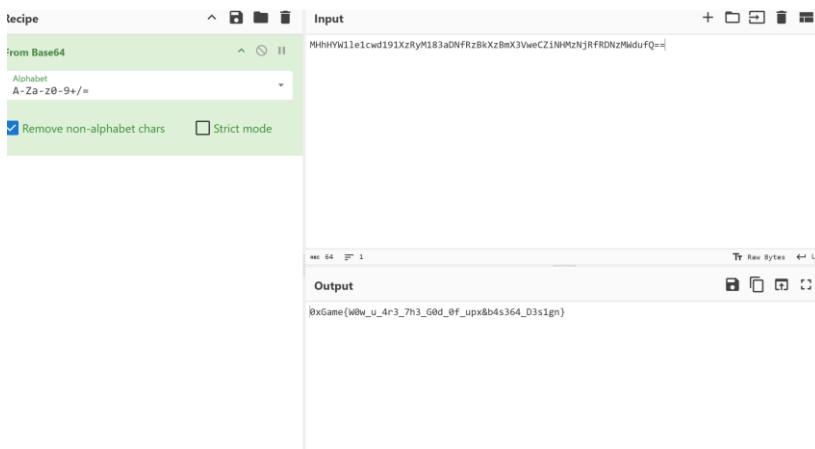
```

```

.rdata:0000000000405000 4B 4C 4D 4E 4F 50 51 52 53 54+ ; DATA XREF: .data:base64_chars$0
.rdata:0000000000405041 00 00 00 00 00 00 00 align 8
.rdata:0000000000405048 4D 48 68 48 59 57 31 6C 65 31+aMhhhyw1le1cd191XzRyM183aDNfRzBkXzBmX3VweCZ1NHmzNjRfRDnzMwdufQ==,0 ; DATA XREF: .data:str$0
.rdata:0000000000405048 63 77 64 31 39 31 58 7A 52 52 79+ ; const char Buffer[]
.rdata:0000000000405089

```

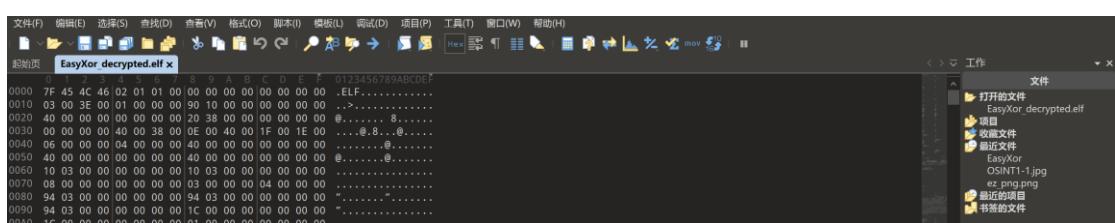
丢进去解码一下



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 .....  
3 .....raputa0x  
0 Game2025.....  
5 Do you know abou  
4 t bitwise operat  
E 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 ....;.....piiÿ  
F x...Điiÿ ...àiiÿ  
0 H...Éöÿÿ,.....  
1 .....zR...x..  
0 .....  
0 .....
```

上一个脚本找到 elf 文件

```
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])

    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}")

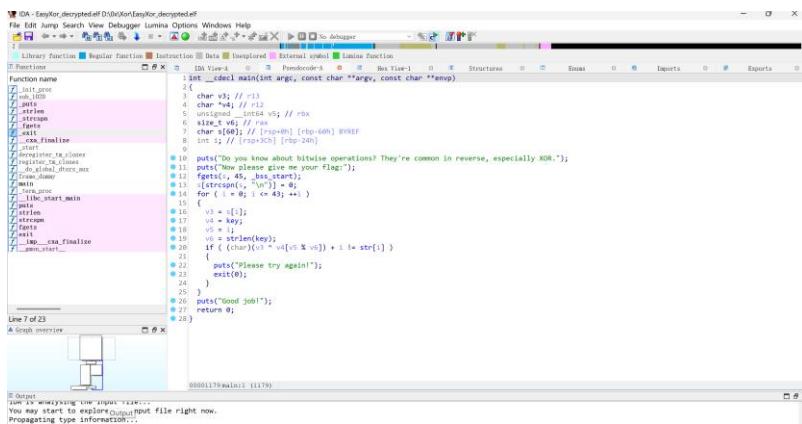
        # 检查文件中是否包含flag提示
        if b'flag' in decrypted_bytes.lower():
            print("文件中包含flag相关字符串")

    return True
```

```
    return False

# 使用你的文件路径
file_path = "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 00 00 00] key dq offset aRaputa0xgame20 ; DATA XREF: main+757r
.data:0000000000004090          ; main+827r
.data:0000000000004090          _data ends ; "raputa0xGame20$"
.data:0000000000004090
.data:0000000000004090
.data:0000000000004090
```

Str 数组

```
000000004060          public str
000000004060          ; unsigned __int8 str[48]
000000004060 42 1A 39 17 1D 09 51 55 2C 5F+str db 42h, 1Ah, 39h, 17h, 10h, 9, 51h, 55h, 2Ch, 5Fh, 63h, 0Ch, 0Dh, 16h, 62h, 27h, 55h, 64h, 55h, 26h
000000004060 63 0C 0D 16 62 27 55 64 55 26+                                         ; DATA XREF: main+8Cto
000000004060 6D 6A 18 34 88 65 6E 1C 21 6E+6Dh, 6Ah, 18h, 34h, 88h, 65h, 6Eh, 1Ch, 21h, 6Eh, 3Dh, 23h, 6Ah, 25h, 6Bh, 63h, 68h, 7Eh, 77h, 75h
000000004060 3D 23 6A 25 6B 63 68 7E 77 75+db 9Ah, 7Dh, 39h, 43h, 4 dup(0)
000000004090          public key
000000004090          ; char *key
```

脚本再+1

```
1 def decrypt_flag():
2     key = b"raputa@xGame2025"
3     key_len = len(key)
4
5     # str数组的44个字节（从IDA中提取）
6     str_data = [
7         0x42, 0x1A, 0x39, 0x17, 0x1D, 0x09, 0x51, 0x55, 0x2C, 0x5F,
8         0x63, 0x0C, 0x0D, 0x16, 0x62, 0x27, 0x55, 0x64, 0x55, 0x26,
9         0x6D, 0x6A, 0x18, 0x34, 0x88, 0x65, 0x6E, 0x1C, 0x21, 0x6E,
10        0x3D, 0x23, 0x6A, 0x25, 0x6B, 0x63, 0x68, 0x7E, 0x77, 0x75,
11        0x9A, 0x7D, 0x39, 0x43
12    ]
13
14    flag = ""
15    for i in range(44):
16        # 反向计算: flag[i] = (str[i] - i) XOR key[i % key_len]
17        flag_char = (str_data[i] - i) ^ key[i % key_len]
18        flag += chr(flag_char & 0xFF) # 确保是有效字节
19
20    return flag
21
22 # 计算并打印flag
23 flag = decrypt_flag()
24 print(f"Flag: {flag}")
```

输出

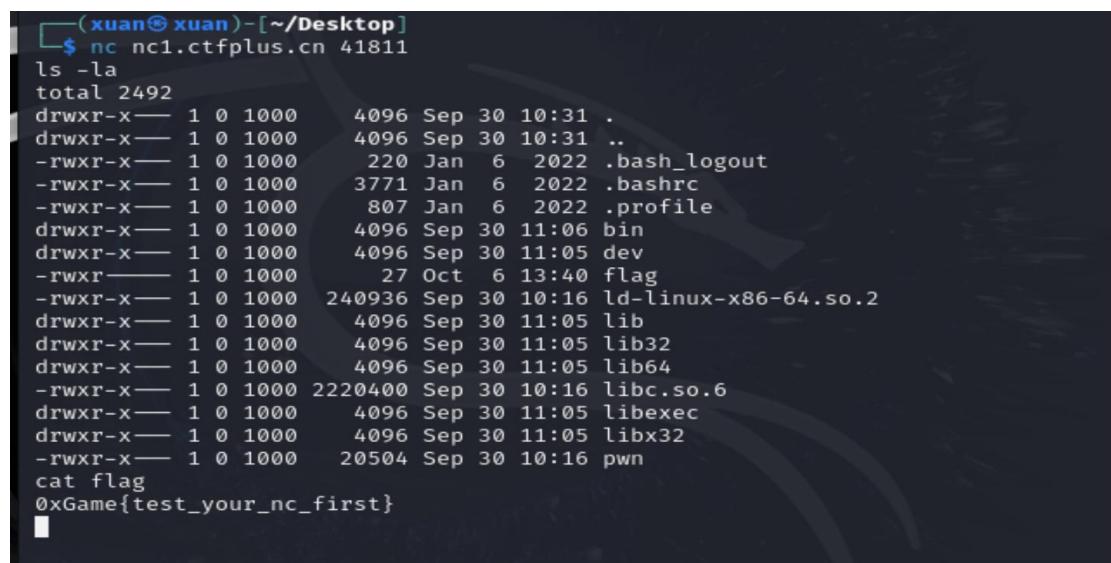
Flag: 0xGame{6c74d39f-723f-42e7-9d7a-18e9508a655b}

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

Pwn

1.test_your_nc

如图



```
(xuan@xuan)@[~/Desktop]
$ nc nc1.ctfplus.cn 41811
ls -la
total 2492
drwxr-x-- 1 0 1000 4096 Sep 30 10:31 .
drwxr-x-- 1 0 1000 4096 Sep 30 10:31 ..
-rwxr-x-- 1 0 1000 220 Jan 6 2022 .bash_logout
-rwxr-x-- 1 0 1000 3771 Jan 6 2022 .bashrc
-rwxr-x-- 1 0 1000 807 Jan 6 2022 .profile
drwxr-x-- 1 0 1000 4096 Sep 30 11:06 bin
drwxr-x-- 1 0 1000 4096 Sep 30 11:05 dev
-rwxr-- 1 0 1000 27 Oct 6 13:40 flag
-rwxr-x-- 1 0 1000 240936 Sep 30 10:16 ld-linux-x86-64.so.2
drwxr-x-- 1 0 1000 4096 Sep 30 11:05 lib
drwxr-x-- 1 0 1000 4096 Sep 30 11:05 lib32
drwxr-x-- 1 0 1000 4096 Sep 30 11:05 lib64
-rwxr-x-- 1 0 1000 2220400 Sep 30 10:16 libc.so.6
drwxr-x-- 1 0 1000 4096 Sep 30 11:05 libexec
drwxr-x-- 1 0 1000 4096 Sep 30 11:05 libx32
-rwxr-x-- 1 0 1000 20504 Sep 30 10:16 pwn
cat flag
0xGame{test_your_nc_first}
```

Flag: 0xGame{test_your_nc_first}

2.命令执行 😱

Cat 用不了，管我 ca' t' 什么事 OvO

```
(xuan㉿xuan) [~/Desktop]
$ nc nc1.ctfplus.cn 11646
Please input your command,no cat no sh!
ca't' flag
0xGame{y0u_c4n_4ls0_3x3cu73_c0mm4nd_w17h0u7_5h_4nd_c47}
sh: 2: : not found
```

Flag: 0xGame{y0u_c4n_4ls0_3x3cu73_c0mm4nd_w17h0u7_5h_4nd_c47}

Crypto

1.Vigenere

脚本一位~

```
1  from string import digits, ascii_letters, punctuation
2
3  key = "Welcome-2025-0xGame"
4  alphabet = digits + ascii_letters + punctuation
5
6  ciphertext = 'WL"mKAaequ{q_aY$oz8`wBqLAF_{cku|eYAczt!pmoqAh+'
7
8  def vigenere_decrypt(ciphertext, key):
9      plaintext = ""
10     key_index = 0
11     for char in ciphertext:
12         bias = alphabet.index(key[key_index])
13         char_index = alphabet.index(char)
14         new_index = (char_index - bias) % len(alphabet)
15         plaintext += alphabet[new_index]
16         key_index = (key_index + 1) % len(key)
17
18     return plaintext
19
20 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！）

百度AI+·识图

追加图片相关问题

伊豆大室山^Q 相似度75% 听



现在, 请随我一起望向那座抹茶色的圆锥! 海拔580米的大室山, 像一颗被绿袖包裹的火山宝石, 山顶30米深的火山口仿佛大地睁开的眼睛

1 2 。

你知道吗

- 她的“年龄”秘密：约4000年前火山喷发的碎屑堆成这座“石头蛋糕”，比金字塔还年长1000岁呢 [1](#) [2](#) !
- 烧山祭的智慧：每年2月第二个星期日“点火迎春”，用火焰给山林“理发”，顺便给春耕除虫，这可是延续千年的生态魔法 [1](#) 。

解锁隐藏玩法

火山口旁藏着弓箭靶场！花500日元租套装备，试试在火山之巅当一回“伊豆射手”，箭矢划过的可是4000年的地质史诗 [3](#) 。

查看一下第二张图属性

纬度	32; 7; 8.9799999999594678
经度	118; 55; 35.6900000000021578
高度	15.7
文件	

换算一下可得经纬度

Flag: 0xGame{大室山_32.1191_118.9265}