BugkuCTF web21_never give up writeup

原创

Mitch311 于 2021-01-14 21:11:15 发布 307 收藏 1
分类专栏： CTF 文章标签： unctf

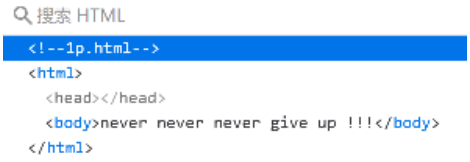
web21_never give up

原题链接

key:URL解码+php黑魔法

①打开环境后只有一串英文never give up

F12查看源码后得到提示□



②尝试访问1p.html，然而跳转到了bugku论坛

不妨查看一下源码

值得注意的是，因为有自动跳转，所以只能**view-source**来查看源码

在地址栏输入**view-source:http://123.206.87.240:8006/test/1p.html**

得到了源码□

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!--


var Words ="%3Cscript%3Ewindow.location.href%3D%27http%3A//www.bugku.com%27%3B%3C/script%3E%20%0A%3C%21--JT
function OutWord()
{
var NewWords;
NewWords = unescape(Words);
document.write(NewWords);
}
OutWord();
// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

③发现中间有URL编码，拿去URL解码□

```
<script>window.location.href='http://www.bugku.com';</script>
<!--JTIyJTNCaWYlMjglMjElMjRfR0VUJTVCJTI3aWQlMjclNUQlMjklMEElN0IlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGh
```

发现仍然存在base64加密，拿去base64解密□

```
%22%3Bif%28%21%24_GET%5B%27id%27%5D%29%0A%7B%0A%09header%28%27Location%3A%20hello.php%3Fid%3D1%27%29%3B%0A%
```

解密结果还是URL编码，再拿去URL解码，得到PHP代码□

```
if(!$_GET['id'])    //限制URL查询字符串中必须有非空非零变量id
    {
     header('Location: hello.php?id=1');
     exit();  //exit() 函数输出一条消息，并退出当前脚本
    }
 $id=$_GET['id'];
 $a=$_GET['a'];
 $b=$_GET['b'];
 if(stripos($a,'.'))   //变量a中不能出现.字符
    {
     echo 'no no no no no no no';
     return ;
    }
 $data = @file_get_contents($a,'r');
 if($data=="bugku is a nice plateform!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114")
//要满足以下5条表达式
//变量 $data 弱等于字符串 bugku is a nice plateform!
//变量 $id 弱等于整型数 0
//变量 $b 的长度大于 5
//字符串 1114 要与字符串 111 连接变量 $b 的第一个字符构成的正则表达式匹配
//变量 $b 的第一个字符弱不等于整型数 4
    {
     require("f4l2a3g.txt");
    }
 else
    {
     print "never never never give up !!!";
    }
```

④进行代码审计□

……thousands of years later……

源码中已暴露出flag文件，有可能是出题人的失误，也有可能是出题人故意用复杂的语句迷惑你，实际上可以直接绕过

因此，**可以直接访问链接http://123.206.87.240:8006/test/f4l2a3g.txt获得flag**

⑤不过也可以进行常规的绕过□

由下图可知，变量 $id 若想满足非空非零且弱等于整型数 0，则 $id 的值只能为非空非零字符串，这里假设 **$id = mitchell**

PHP 弱类型比较表

松散比较 ==

	true	false	1	0	-1	"1"	"0"	"-1"	null	array()	"php"	""
true	true	false	true	false	true	true	false	true	false	false	true	false
false	false	true	false	true	false	false	true	false	true	true	false	true
1	true	false	true	false	false	true	false	false	false	false	false	false
0	false	true	false	true	false	false	true	false	true	false	true	true
-1	true	false	false	false	true	false	false	true	false	false	false	false
"1"	true	false	true	false	false	true	false	false	false	false	false	false
"0"	false	true	false	true	false	false	true	false	false	false	false	false
"-1"	true	false	false	false	true	false	false	true	false	false	false	false
null	false	true	false	true	false	false	false	false	true	true	false	true
array()	false	true	false	false	false	false	false	false	true	true	false	false
"php"	true	false	false	true	false	false	false	false	false	false	true	false
""	false	true	false	true	false	false	false	false	true	false	false	true

有关 PHP 类型比较的详情可参考：PHP 类型比较表

源码中变量 $data 是由 file_get_contents() 函数读取变量 $a 的值而得，所以 **$a 的值必须为数据流**

在服务器中自定义一个内容为 bugku is a nice plateform! 文件，再把此文件路径赋值给 $a，显然不太现实

因此这里用伪协议 php:// 来访问输入输出的数据流，其中 **php://input 可以访问原始请求数据中的只读流**

这里令 **$a = php://input，并在请求主体中提交字符串 bugku is a nice plateform!**

> 有关 PHP 伪协议的详情可参考：支持的协议和封装协议

先来了解一下eregi()截断漏洞□

> **ereg()函数或eregi()函数存在空字符截断漏洞，即参数中的正则表达式或待匹配字符串遇到空字符则截断丢弃后面的数据**

源码中待匹配字符串（第二个参数）已确定"1114"，**正则表达式(第一个参数)由"111"连接b的第一个字符组成**，若令substr(b,0,1) = "\x00"，即满足 "1114" 与 "111"”匹配

> **\x00即0x00，是一个十六进制转义符，用来表示空字符**

因为$b是URL查询字符串中的变量，所以不应该在此放入空字符\x00，而应该为空字符的URL编码%00

注意，虽然**$b=%0012345**实际字符串长度为8字节，但在后台脚本读入数据时，会将URL 编码%00转换成1字节

> 有关**PHP**的各种黑魔法可参考：PHP函数黑魔法小总结

⑥用burpsuite抓一下包，放到repeater

用构造好的payload去go一下，得到flag□

值得注意的是，请求主体中**bugku is a nice plateform!**后面不要加回车或者空格

虽然看不出来毛病，然而a的值天差地别！！！