Bugku — never give up （PHP 黑魔法函数绕过）— writeup

转载

xuchen16 于 2018-09-20 11:31:31 发布 3779 收藏 4
分类专栏： ctf 文章标签： Bugku never give up writeup Bugku — never give up wp

转载自：https://ciphersaw.me/2017/12/26/%E3%80%90Bugku%20CTF%E3%80%91%20Web%20%E2%80%94%E2%80%94%20never%20g

0x00 前言

此题为 Web 基础题，难度中低，需要的基础知识有：**HTML、PHP、HTTP 协议**。

首先考查**发现源码**的能力，其次重点考查 **PHP 黑魔法**的使用，相关链接如下：

- **题目链接：**http://123.206.31.85/challenges#never give up
- **解题链接：**http://120.24.86.145:8006/test/hello.php

0x01 拦截跳转

点开解题链接，除了一句 **never never never give up !!!** 之外空空如也，直接查看源码，发现一条注释中有线索：

根据提示打开链接：http://120.24.86.145:8006/test/1p.html，发现跳转回 Bugku 的主站，所以祭出 BurpSuite 进行抓包拦截。

请求数据包如下：

响应数据包如下：

0x02 三重解码

根据响应内容中变量 `Words` 的值，容易得出是一段 **URL 编码**后的数据：

```
%3Cscript%3Ewindow.location.href%3D%27http%3A//www.bugku.com%27%3B%3C/script%3E%20%0A%3C%21--
JTIyJTNCaWYlMjglMjElMjRfR0VUJTVCJTI3aWQlMjclNUQlMjklMEElN0IlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGhlbGxvLnBocCUzRmlkJTNEMSUyNyUyOSUzQiUwQSUwOWV4aXQlMjglMjklM0IlMEElN0QlMEElM
-%3E
```

对其进行解码，得到一条 Javascript 语句与一大段注释，易看出注释中的内容是一段 **Base64 编码**后的数据：

```
<script>window.location.href='http://www.bugku.com';</script>
<!--
JTIyJTNCaWYlMjglMjElMjRfR0VUJTVCJTI3aWQlMjclNUQlMjklMEElN0IlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGhlbGxvLnBocCUzRmlkJTNEMSUyNyUyOSUzQiUwQSUwOWV4aXQlMjglMjklM0IlMEElN0QlMEElM
-->
```

将注释中的内容进行解码，发现又是一大段 **URL 编码**后的数据：

```
%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%09exit%28%29%3B%0A%7D%0A%24id%3D%24_GET%5B%27id%2
```

进行 URL 解码后，终于得到一段不完整的 **PHP 核心源码**：

```
";if(!$_GET['id'])
{
header('Location: hello.php?id=1');
exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(stripos($a,'.'))
{
echo '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($b,0,1),"1114") and substr($b,0,1)!=4)
{
require("f4l2a3g.txt");
}
else
{
print "never never never give up !!!";
}
?>
```

其中各条核心语句的作用如下：

- 第 2行：限制 URL 查询字符串中必须有非空非零变量 `id`
- 第 10 行：限制变量 `$a` 中不能含有字符 `.`
- 第 15 行：要满足以下 5 条表达式才会爆 flag：
  - 变量 `$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

注意，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第 15 行复杂的语句迷惑你，实际上可以绕过。因此，直接访问链接 http://120.24.86.145:8006/test/f4l2a3g.txt 即可获得 flag。

不过，第 15 行的语句也是可解的（应该也是此题的本意），请继续往下看。

0x03 PHP 黑魔法

本节分别针对源码中 `$id`、`$a`、`$b` 三个变量需要满足的条件进行讲解。

PHP 弱类型比较

由上图可知，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 = "asd"`。

> 有关 PHP 类型比较的详情可参考：PHP 类型比较表

PHP 伪协议

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

eregi() 截断漏洞

CTF 题做多了就知道 `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）已确定为 `"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。因此，这里假设 `$b = "\x0012345"`，才能满足以上三个条件。

有关 PHP 的各种黑魔法可参考：

> PHP函数黑魔法小总结
> CTF之PHP黑魔法总结
> 那些年学过的PHP黑魔法

0x04 构造 payload 爆 flag

分析出以上三个变量应该等于什么值后，接下来构造出对应的 payload 自然就 get flag 了。之所以将构造 payload 单独拿出来讲，是想分享笔者在构造 payload 过程中踩过的坑。

在构造变量 `b` 中的空字符时，过早将空字符 `\x00` 放入，在提交请求时导致请求头截断，继而请求失败，得不到响应。

因为 `b` 是 URL 查询字符串中的变量，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。所以说，空字符应该在后台脚本的变量中出现，而不是在 URL 查询字符串变量中出现。

构造出正确的 payload 后，完成此题常规思路的做法：

stripos(字符串a，字符串b) 函数查找字符串b在字符串a中第一次出现的位置（不区分大小写）。

file_get_contents 将整个文件读入一个字符串

strlen() 函数返回字符串的长度

substr() 函数返回字符串的一部分。 **substr(string,start,length)** ，length参数可选。如 substr($b,0,1) 就是在参数b里面 ，从0开始返回1个长度的字符串

eregi("111".substr($b,0,1),"1114") 就是判断"1114"这个字符串里面是否有符合"111".substr($b,0,1)这个规则的

对b的要求为长度大于5且b[0]!=4，且’1114’中存在’111’+b[0]。Strlen检测长度时遇到%00不会被截断，而eregi会被截断，故构造b为字串，首位为%00,长度大于5，则检测时长度满足，eregi(‘111’,’1114’)也满足，b首位整形也不等于4也满足

!$_GET[‘id’]并且id==0：令id=%00或者令id=.字符串都可以绕过

php认定0e开头的 == 0， php://input 绕过file_get_contents， 星号或问号或.点号绕过正则表达式的匹配

payload： http://120.24.86.145:8006/test/hello.php?id=0e123&a=php://input&b=*23456