

bugku.welcome to the bugkuctf

原创

林夕林@ 于 2019-09-25 19:05:19 发布 190 收藏

分类专栏: [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/qq_44598397/article/details/101380317

版权



[bugku](#) 专栏收录该内容

16 篇文章 0 订阅

订阅专栏

这里记录我花了一整天才理清题目意思的题目, 因为接触CTF时间还不长, 水平还停留在各个方面都稍有了解, 但是涉及的知识稍微深入一些就没有思路了, 这道题目涉及了php的代码审计, php的伪协议, php的序列化, php魔术方法。

因为我是小白一个, 所以这道题目稍微把一些细节的地方说多一点, 万一有些朋友看到其他writeup没看懂, 看到我说的一些地方有可能就懂了。。。。

```
1 you are not the number of bugku !
2
3 <!--
4 $user = $_GET["txt"];
5 $file = $_GET["file"];
6 $pass = $_GET["password"];
7
8 if(isset($user)&&(file_get_contents($user,'r')== "welcome to the bugkuctf")){
9     echo "hello admin!<br>";
10    include($file); //hint.php
11 }else{
12    echo "you are not admin ! ";
13 }
14 -->
```

<https://blog.csdn.net/yh1013024906>
https://blog.csdn.net/qq_44598397

首先打开题目查看源代码以后:

从上面的代码可以看出以下信息:

通过get方法传递三个值: txt,file,password

读取\$user文件的内容, 并且文件内容要与'welcome to the bugkuctf'相同

\$file经提示应该为hint.php

因为file_get_contents(

user,r)这个函数的意思是将 user这个文件的内容写到字符串里去, 就是说user文件里的内容会变成一

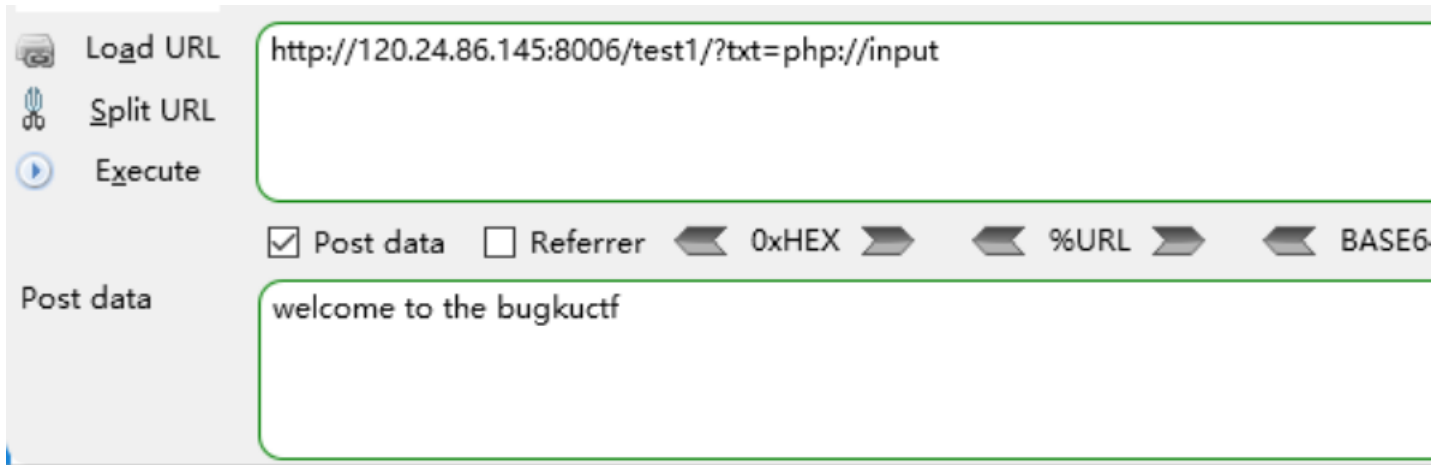
这里就涉及到一个php伪协议, 就是php://input, 它的大概意思就是可以读取我们post传递的数据。它的详解在这篇文章中有很好的解释 https://blog.csdn.net/qq_27682041/article/details/73326435。

所以看完了这篇文章的话, 你应该就知道file_get_contents()函数里面放的不止文件名哦, 还可以放php的伪协议, 如果把把这个

php://input作为文件名放进去的话，这个函数发现是一个伪协议，那作为一个“文件”，它里面肯定是没有内容的吧，那要怎么把它的内容变成一个字符串呢，它会读取我们post传递的数据作为它的“文件内容”，然后再变成一个字符串。

如说我们现在有这么一句 `file_get_contents("php://input")`，然后我们又post传递了一个数据的话，那这个数据就会被php://input读取到，然后`file_get_contents`又把它变成字符串。

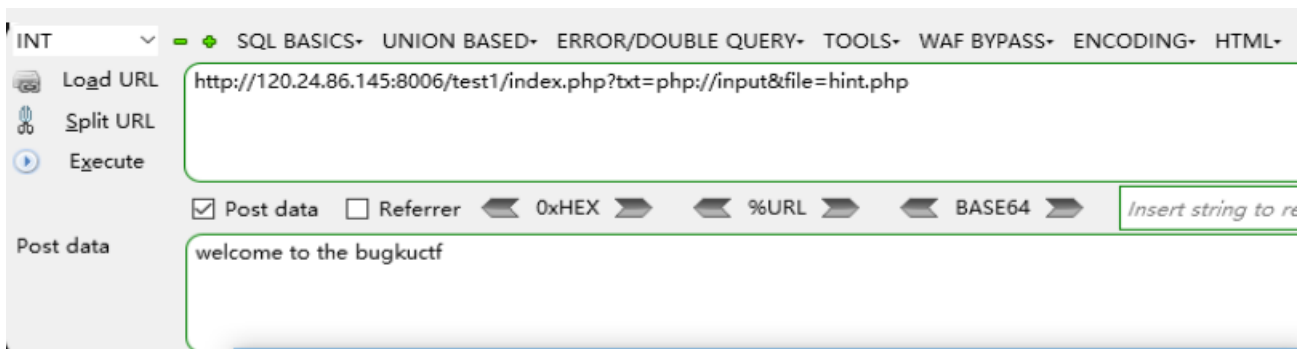
所以我们构造`txt=php://input`，并且post一个"welcome to the bugkuctf"试试看



hello friend!

<https://blog.csdn.net/vh1013024906>
<https://blog.csdn.net/vh1013024906>

网页变了，但是为什么是一个hello friend!呢。。。我也不知道，不过我们代码中的第一层已经解开了，再往下看吧。有一个 `include(file);//hint.php`，首先我们知道`include(file)`是动态读取文件名，然后又提示我们`hint.php`，那岂不是说我们可以直接让`file=hint.php`就可以得到下一步信息了！



hello friend!



<https://blog.csdn.net/vh1013024906>

果然还是没有那么简单。。。不然怎么引出我们第二个php伪协议呢？php://filter 这个协议现在我只知道可以

以用来读取网页base64编码后的源代码。用这句 `file=php://filter/read=convert.base64-encode/resource=hint.php`

就可以得到hint.php这个网页的base64编码后的源代码了。

```
1 <?php
2
3 class Flag{//flag.php
4     public $file;
5     public function __toString(){
6         if(isset($this->file)){
7             echo file_get_contents($this->file);
8             echo "<br>";
9             return ("good");
10        }
11    }
12 }
13 ?>
```

<https://blog.csdn.net/yh1013004806/article/details/105939357>

光看这个代码是不是感觉信息不是很多呢，要不我们再看看index.php这个文件的源代码。

```
1 <?php
2 $txt = $_GET["txt"];
3 $file = $_GET["file"];
4 $password = $_GET["password"];
5
6 if(isset($txt)&&(file_get_contents($txt,'r')==="welcome to the bugkuctf")){
7     echo "hello friend!<br>";
8     if(preg_match("/flag/", $file)){
9         echo "不能现在就给你flag哦";
10        exit();
11    }else{
12        include($file);
13        $password = unserialize($password);
14        echo $password;
15    }
16 }else{
17     echo "you are not the number of bugku ! ";
18 }
19
```

<http://blog.csdn.net/qiyij448823886>

不要问我为什么知道要看index.php里的，因为这个一般是主页文件，有很重要的信息嘛，嘿嘿嘿嘿嘿嘿。

可以看到这么多代码，虽然很烦，但是先从简单的地方来理解。

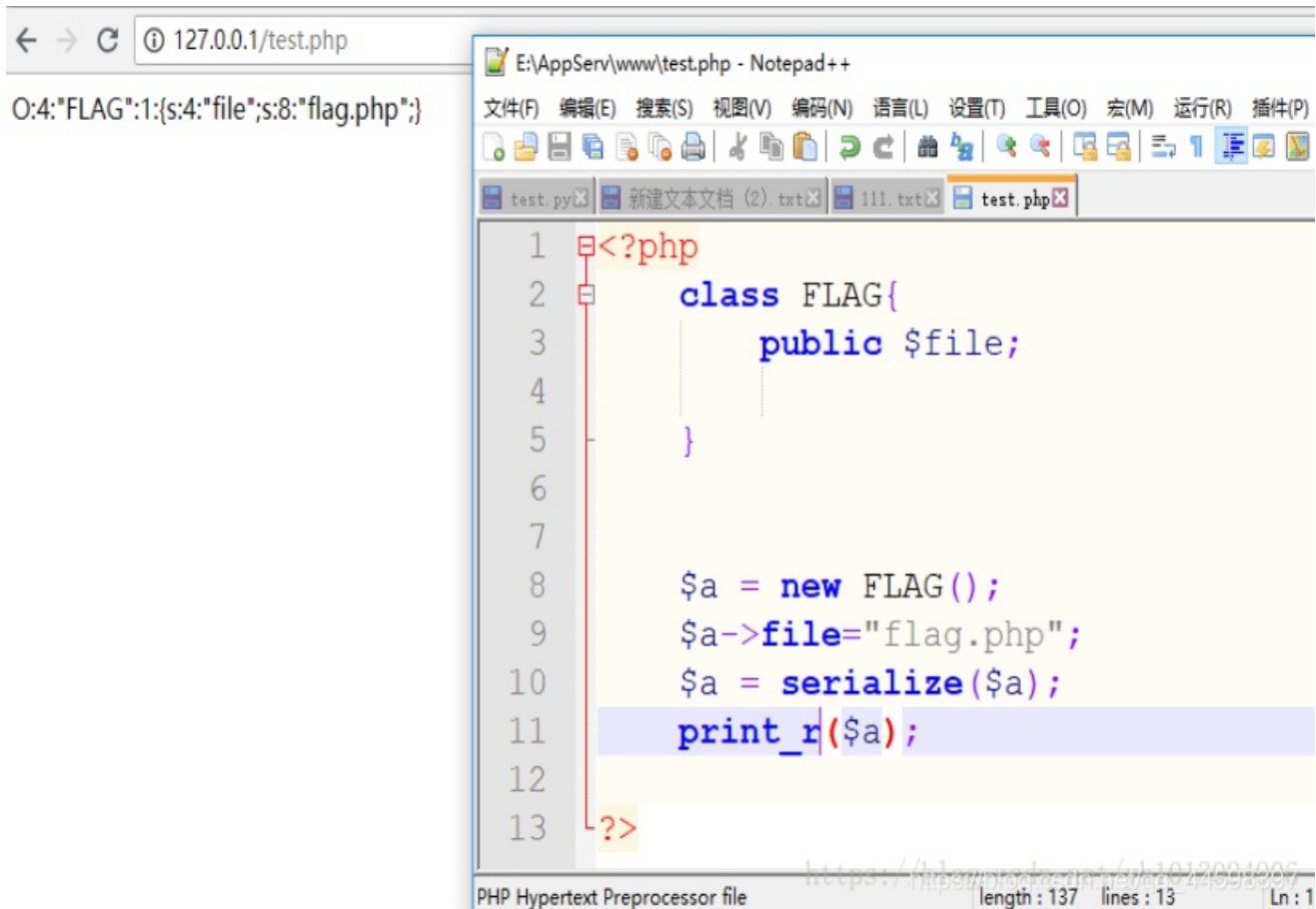
第一张图提示了我们flag.php，但是我们可以看第二张图，如果我们设定的文件名中包含'flag'，那么就会跳出“不能现在就给你flag哦”然后exit（）；

继续看第二张图，第一个if告诉我们，我们之前的大前提并没有改变，但是在这个前提下还附加了一些条件，如果文件名没有"flag"了，就会把这个文件包含进来，然后password进行反序列化，再输出password的值。什么是反序列化呢，那就要先知道序列化啦，在PHP中,序列化用于存储或传递 PHP 的值的 过程中,同时不丢失其类型和结构（这是百度的）。我们可以把它想成一种编码嘛。

回到第一张图，定义了一个类 **FLAG**，类里面有一个 **\$file** 属性，并且有一个魔术方法 **_tostring()**，这个方法的作用就是当调用实例化对象时就会自动执行 **_tostring()** 这个方法。简单来说创建一个这个类的对象就会调用这个方法。魔术方法呢，就是一个很神奇的方法（一本正经的胡说八道），大概就是一种类里面默认的方法，你可以对它进行改造，类似于构造方法。反正大家都要了解的，去百度看看用法和详解吧。

_tostring() 方法里面又定义了如果 **\$file** 这个属性有赋值的话，那么就会输出这个文件的内容（输出成一个字符串）。

所以根据上面的这些条件，我们可以让 **password** 为 **FLAG** 类型，并且让 **FLAG** 中的 **file** 就等于 **flag.php**，这样我们就可以得到 **flag.php** 的内容了，不过要记得，前面 **\$password** 进行了反序列化的操作，所以我们要先把它序列化。写一个 **php** 脚本吧！



The screenshot shows a Notepad++ window titled "E:\AppServ\www\test.php - Notepad++". The address bar shows "127.0.0.1/test.php". The code in the editor is as follows:

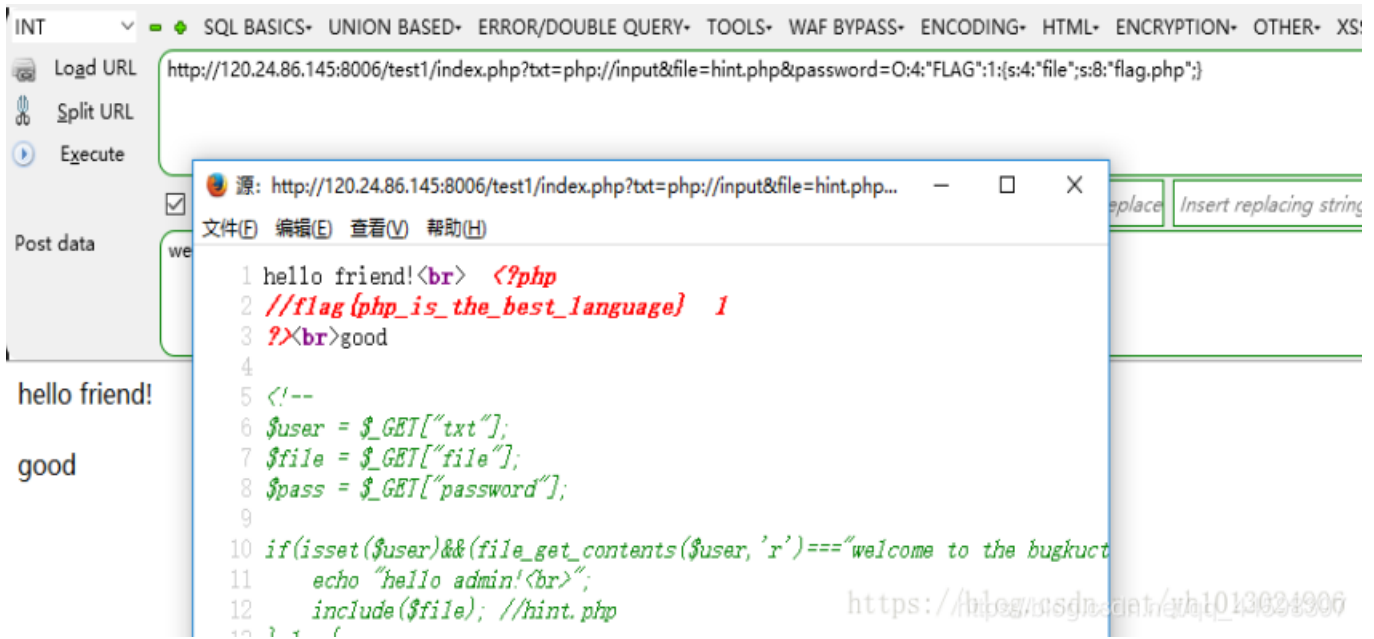
```
0:4:"FLAG":1:{s:4:"file";s:8:"flag.php"};

1  <?php
2      class FLAG{
3          public $file;
4      }
5
6
7
8      $a = new FLAG();
9      $a->file="flag.php";
10     $a = serialize($a);
11     print_r($a);
12
13  ?>
```

The status bar at the bottom indicates "PHP Hypertext Preprocessor file", "length: 137", "lines: 13", and "Ln: 1".

在本地的服务器跑一下，我们就可以得到序列化的一个 **FLAG** 类咯。大家如果还没有搭本地服务器的话，推荐先自己学习一下怎么搭建环境，在本地可以自己试验各种 **php** 代码，可以自己了解数据库一些操作等等。下一个 **APPSEV** 就可以啦，这是一个合集，也是有中文的，网上都有教程的，很简单。

那最后我们就给\$password赋值吧！



得到了flag啦。

感谢您的观看，这是我的第一篇文章，没有什么写作的经验，而且自己现在的技术没有多高，就是一个刚入门的小白，因为经常在writeup上看不懂，所以我会用自己理解的方式做下笔记，方便自己以后查看的时候能够看懂，在一些名词的解释方面可能不太正确，推荐大家去看看大佬们的解释，我只是用自己的方式记忆。。。如果文章中有哪里出错，希望您能指出让我改进，如果有和我一样刚入门的小白想知道bugku上其他一些题目的思路也可以告诉我，如果我会的话，会尽力写一篇writeup讲解的，再次感谢！

作者：你好我叫易烺千玺

来源：CSDN

原文：<https://blog.csdn.net/yh1013024906/article/details/81087939>

版权声明：本文为博主原创文章，转载请附上博文链接！

以下自己的过程：