BUUCTF Easy Calc

原创

H9_dawn 于 2020-04-12 08:15:17 发布 554 收藏 2

分类专栏： CTF 文章标签： web 安全

本文链接：https://blog.csdn.net/qq_43622442/article/details/105462439

CTF 专栏收录该内容

20 篇文章 2 订阅

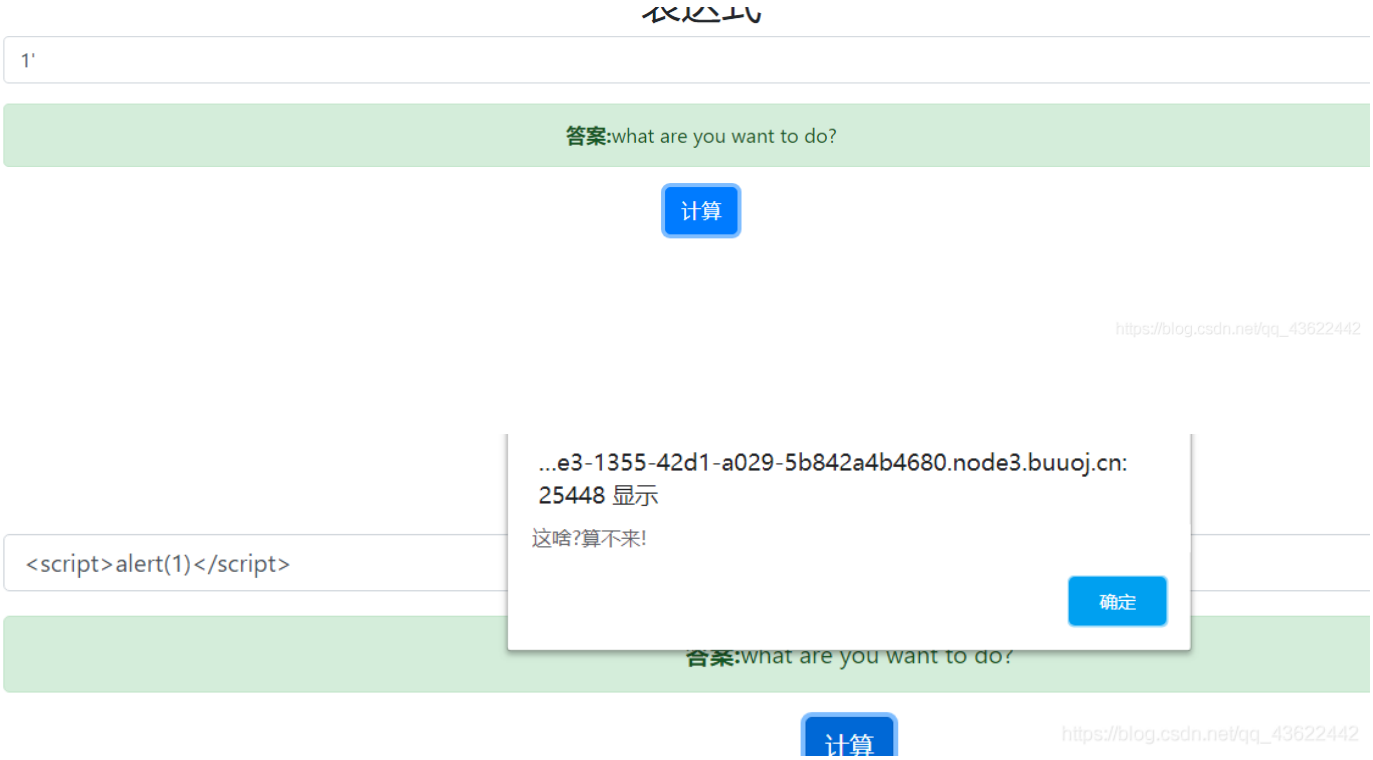
订阅专栏

BUUCTF Easy Calc

= =

1.打开网站，让我们算术：

2.看到框框肯定要插一下：（这里两种不同的提示方式，就应该想到waf另有其人的，害）



3.拦截了，猜测后端应该是直接对语句执行了输出的，看看源码：

```
<div class="container text-center" style="margin-top:30px;">…</div>
<!--I've set up WAF to ensure security.-->
<script>
        $('#calc').submit(function(){
            $.ajax({
                url:"calc.php?num="+encodeURIComponent($("#content").val()),
                type:'GET',
                success:function(data){
                    $("#result").html(`<div class="alert alert-success">
                <strong>答案:</strong>${data}
                </div>`);
                },
                error:function(){
                    alert("这啥?算不来!");
                }
            })
```


4.说是有**waf**，而且之前插入抓包的时候也发现了**/calc.php?num=**

Raw Params Headers Hex

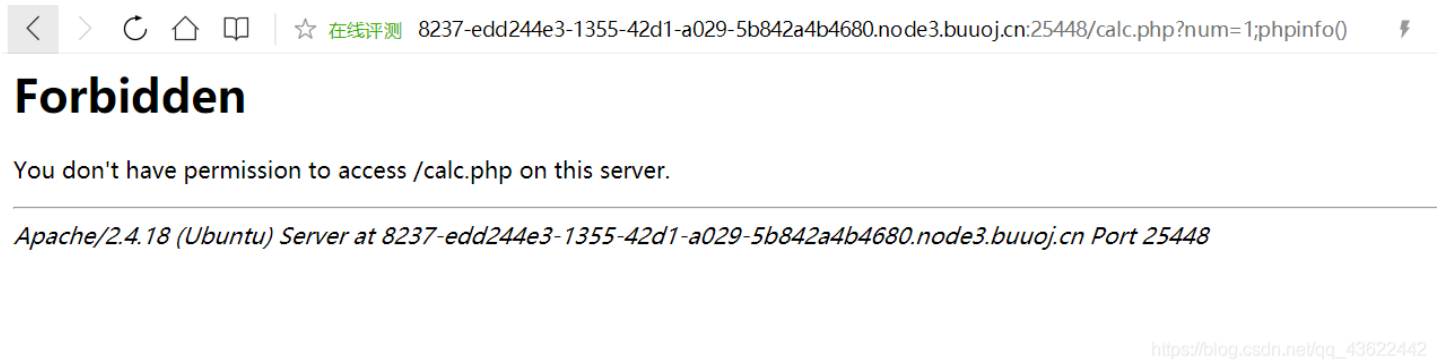
```
GET /calc.php?num=1 HTTP/1.1
Host: 8237-edd244e3-1355-42d1-a029-5b842a4b4680.node3.buuoj.cn:25448
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://8237-edd244e3-1355-42d1-a029-5b842a4b4680.node3.buuoj.cn:25448/
```


5.==可我真没想到，不加参数的话会直接显示源码的，做题和实战有偏差**==**

```
        show_source(__FILE__);
}else{
                $str = $_GET['num'];
                $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
                foreach ($blacklist as $blackitem) {
                        if (preg_match('/' . $blackitem . '/m', $str)) {
                                die("what are you want to do?");
                        }
                }
                eval('echo '.$str.';');
}
?>
```

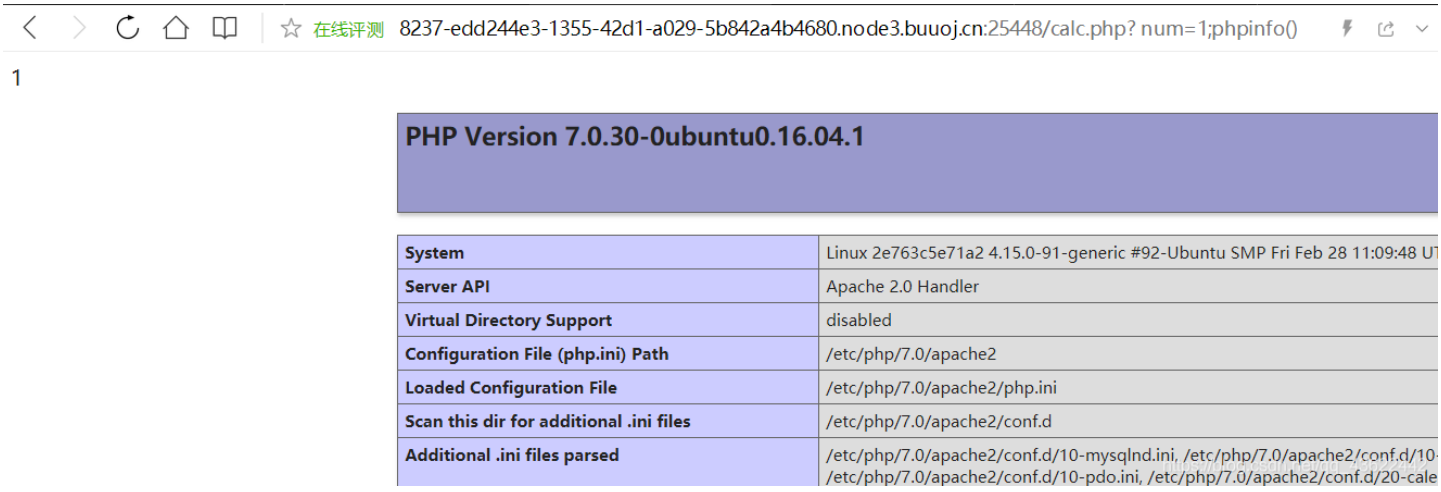

这里就是设置了黑名单，然后代码执行。看到这儿就想到了一句话，但是“[”这些被拦截了，我又想到了我写过的异或免杀马儿，但是这里有些字符组不出来。

6.好吧，看到这个拼接语句，我们肯定是可以拼接自己的语句的。但是想想有什么能在eval运行的…只知道一句话和phpinfo = =我tcl

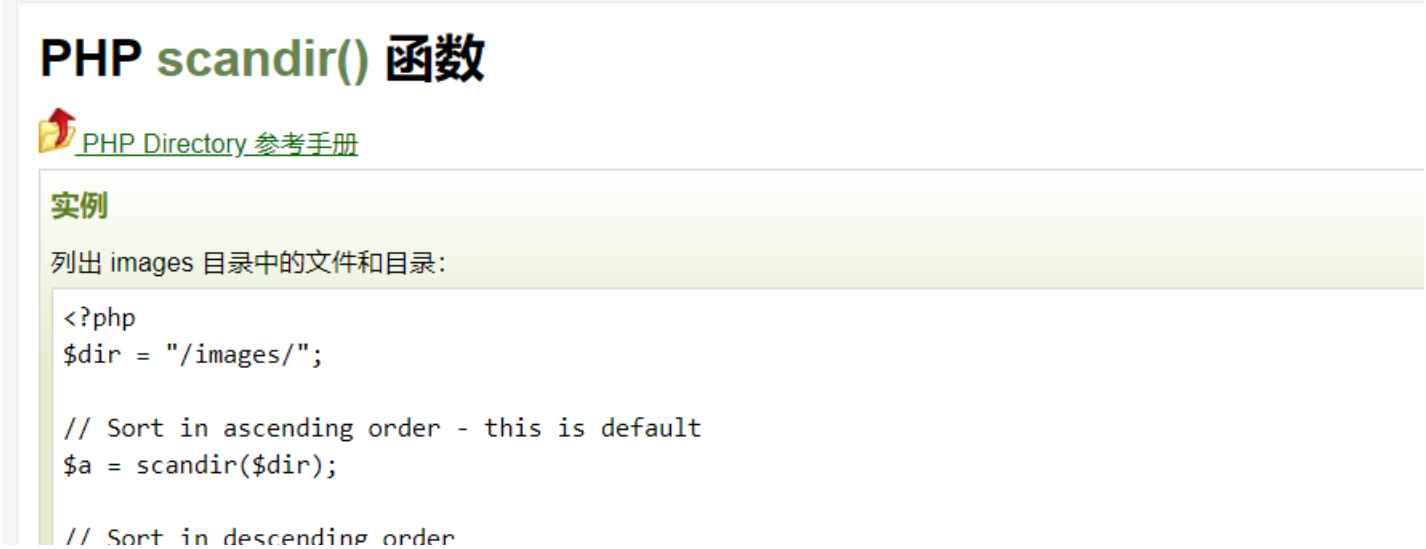


wc，403，为什么。突然发现waf另有其人，我一直以为就是之前源代码里面的拦截…怎么绕过呢？要不参数污染、要不参数值绕过…恕我知识有限、没有成功。

7.瞄了一眼wp，长知识了。php会把传过来的参数进行处理后再存储到GET或者POST这种数组中，包括：
1）删除初始空格
2）把某些字符转换为下划线
但是在waf层面没有进行这个操作，那waf可能不是php写的吧…
在num参数前面加空格绕过：（有时间试试绕waf）



8.拿到phpinfo页面也就看一下allow_url…，看一下绝对路径，看一下open_basedir和disable_functions现在也利用不上。去查一些函数，结合disable_functions发现，scandir()函数可以用：



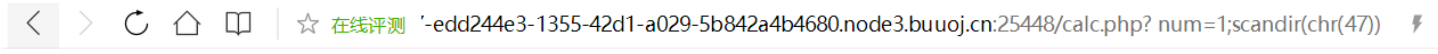
```
$b = scandir($dir,1);

print_r($a);
print_r($b);
?>
```

结果：

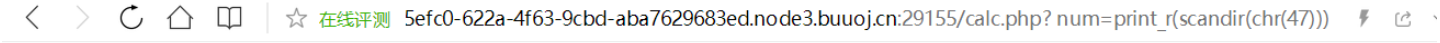
```
Array
(
[0] => .
[1] => ..
[2] => cat.gif
[3] => dog.gif
[4] => horse.gif
[5] => myimages
)
Array
```


9.这个函数是指定获取某个目录下的目录和文件，但是引号和**$**都被过滤了，既不能直接输入目录也不能构造参数然后传入。然后很容易想到了**chr**函数。



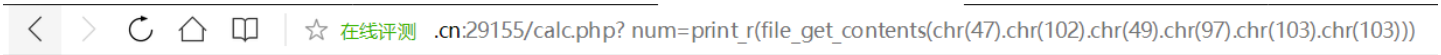
1

10.这是获取根目录下的文件和目录，但是为什么没有回显呢？因为它返回的是数组形式，在**php**里我们不能直接输出，要用**var_dump**或者**print_r**。这两个函数都可以把**flag**打印出来，**print_r**和**var_dump**都能输出数组和对象，但**print_r**对布尔型的输出不太明显；**var_dump**输出比较详细，一般调试时用得多。



Array ([0] => . [1] => .. [2] => .dockerenv [3] => bin [4] => boot [5] => dev [6] => etc [7] => f1agg [8] => home [9] => lib [10] => lib6
=> root [16] => run [17] => sbin [18] => srv [19] => start.sh [20] => sys [21] => tmp [22] => usr [23] => var) 1

11.我看到了**f1agg**，那没跑了。获取文件内容肯定是要用**file_get_contents**了。不过这里也要用**ascii**代替。



flag{bd06a68e-5de1-4c5b-8e65-4571c8cb6b51} 1

~~昨晚上撑不住睡了，hhh，今儿早上起来补完。~~