CTF学习笔记——Easy Calc

原创

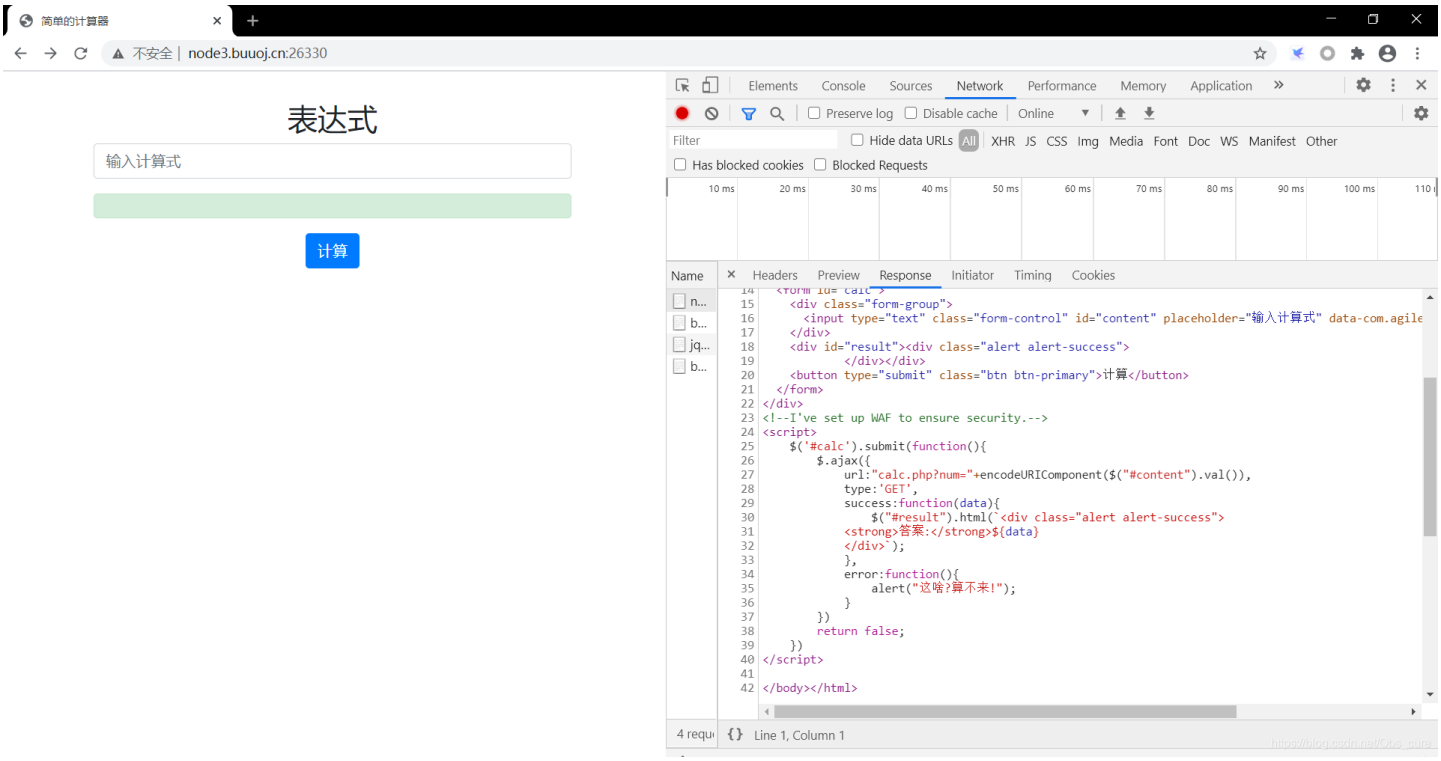
Obs_cure 于 2020-09-06 16:57:38 发布 394 收藏 1

文章标签： 网络安全

本文链接：https://blog.csdn.net/Obs_cure/article/details/108424183

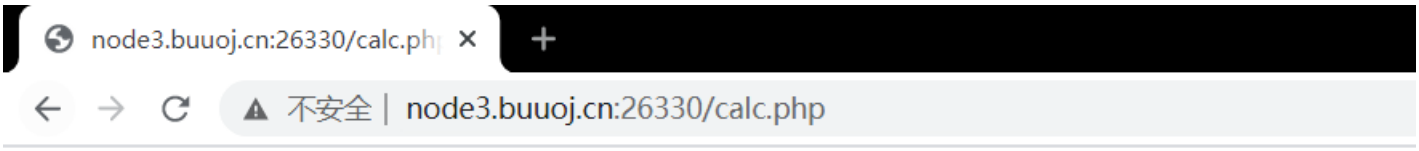
一、[RoarCTF 2019]Easy Calc

1.题目



2.解题步骤

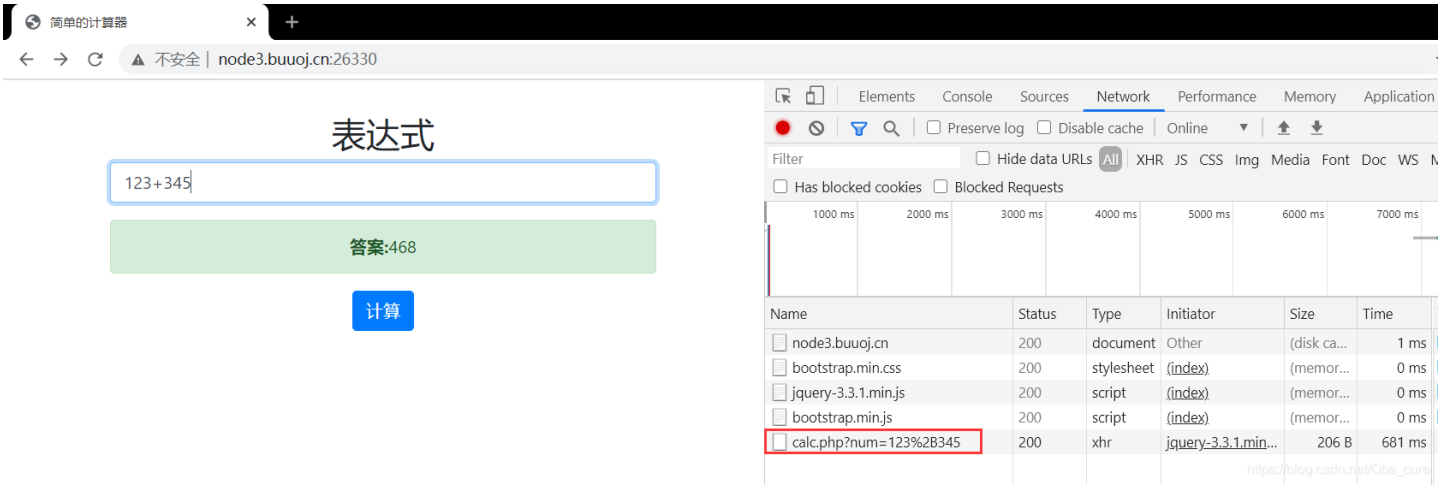
发现框框...应该是注入题...源码中提示有个waf，继续看源码，发现有个calc.php，进去看看




```php
<?php
error_reporting(0);
if(!isset($_GET['num'])){
        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.';');
}
?>
```

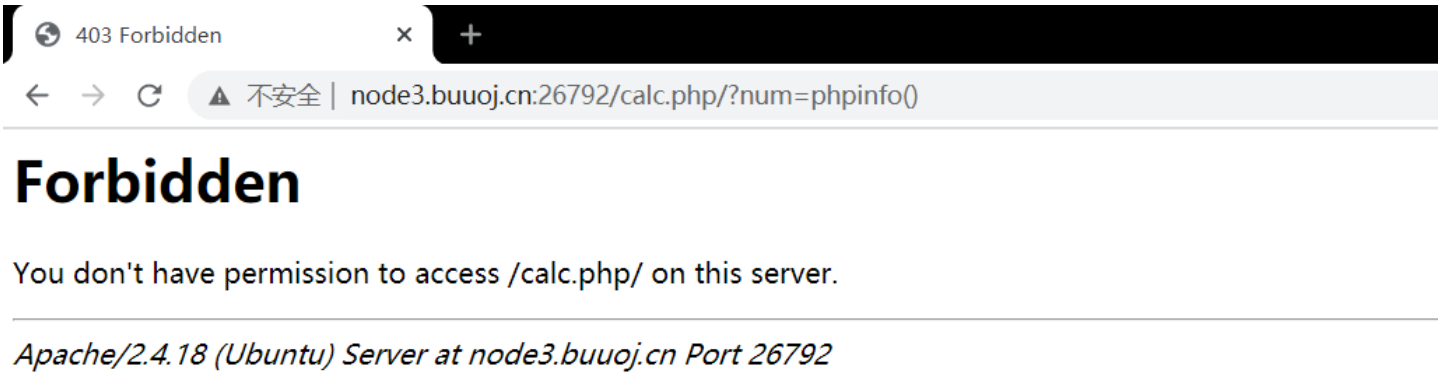
初步理解应该是用num传参，然后返回计算结果，这个php文件用于过滤非法字符。

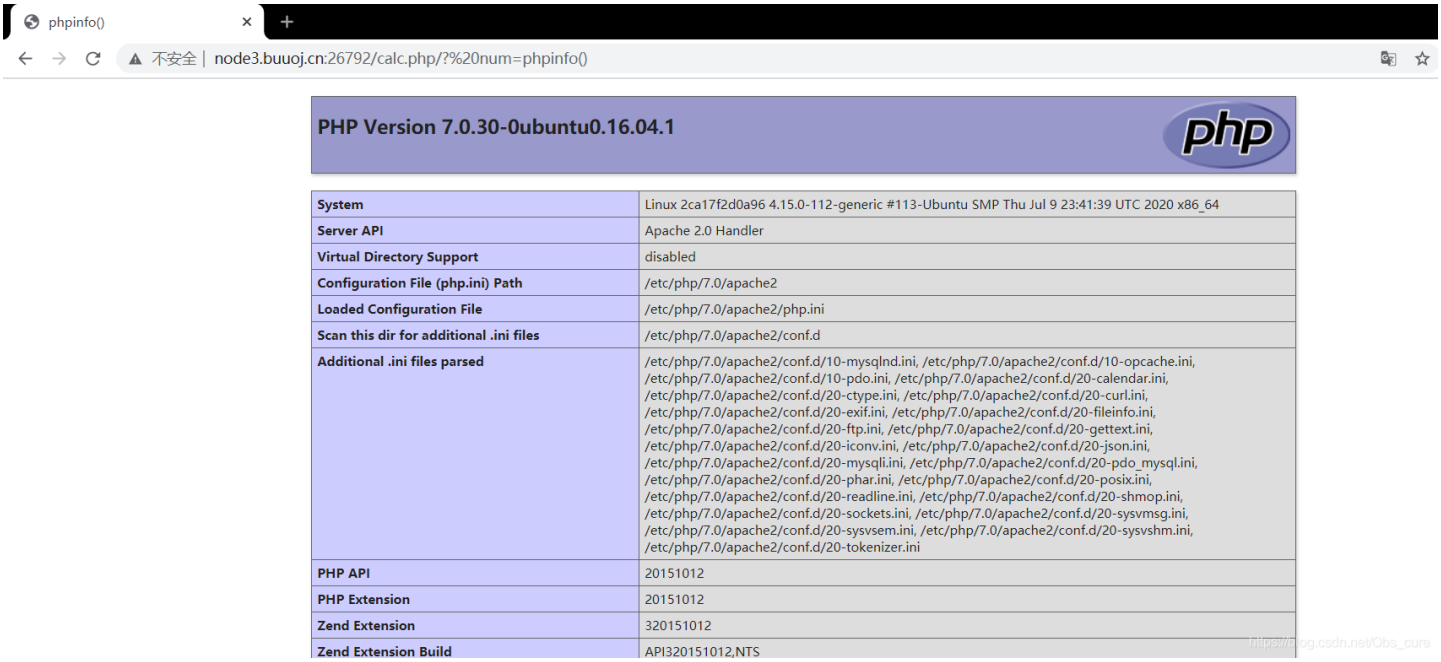
被拦下来了，大概是利用num构造一个payload，不能含有字母和非法字符，只能有数字和符号



这里已经不会了，看writeup吧…

先贴效果





可以发现，num和 num仅仅差了一个空格，但是 num就绕过了waf，执行了代码。这是一个利用php字符串解析的漏洞。这个漏洞的产生主要是有两个原因。

1. 在calc.php中，可以发现是对num变量进行的一个过滤，并没有对 num变量过滤，因此可以对 num进行传参
2. 由于php的字符串解析特性，在解析php字符串时，会自动将空格过滤掉，因此传递的参数依旧是num。

在calc.php文件的源码中，可以看到最后一句会执行num的语句的内容。因此我们只需要向 num这个变量传递代码就能执行了。

接下来要认识几个函数：

1.`void var_dump ( mixed $expression [, mixed $... ] )` 用于输出变量的相关信息。（和echo差不多）

参数名	描述
$expression	你要输出的变量

2.`scandir(directory,sorting_order,context)` 返回指定目录中的文件和目录的数组。

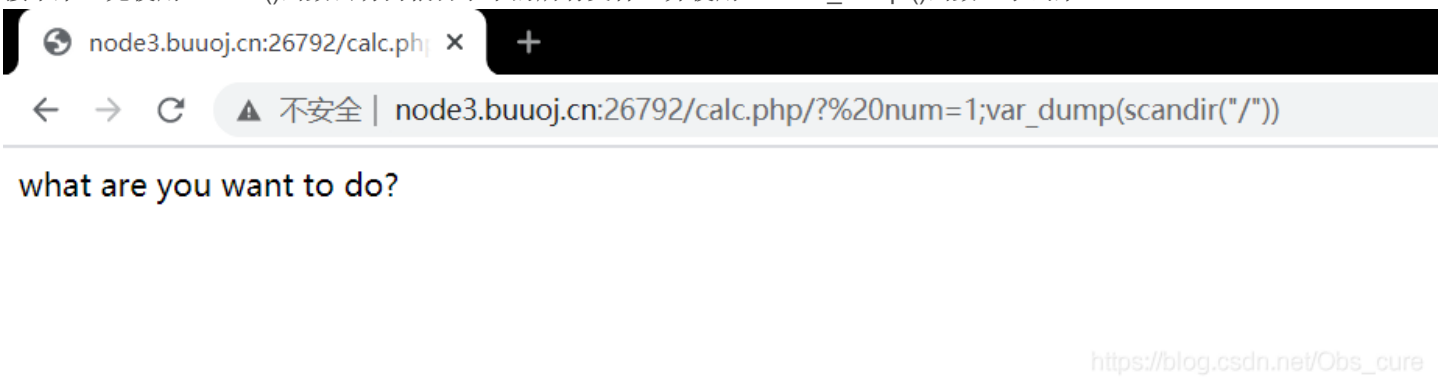
参数名	描述
directory	必需。规定要扫描的目录
sorting_order	可选。规定排列顺序。默认是 0，表示按字母升序排列。如果设置为 SCANDIR_SORT_DESCENDING 或者 1，则表示按字母降序排列。如果设置为 SCANDIR_SORT_NONE，则返回未排列的结果
context	可选。规定目录句柄的环境。context 是可修改目录流的行为的一套选项

3.`file_get_contents(path,include_path,context,start,max_length)` 把整个文件读入一个字符串中。

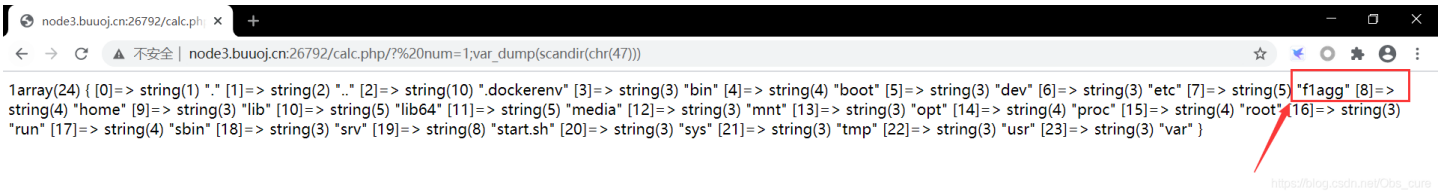
参数名	描述
path	必需。规定要读取的文件。

参数名	描述
include_path	可选。如果您还想在 include_path（在 php.ini 中）中搜索文件的话，请设置该参数为 '1'。
context	可选。规定文件句柄的环境。context 是一套可以修改流的行为的选项。若使用 NULL，则忽略。
start	可选。规定在文件中开始读取的位置。该参数是 PHP 5.1 中新增的。
max_length	可选。规定读取的字节数。该参数是 PHP 5.1 中新增的。

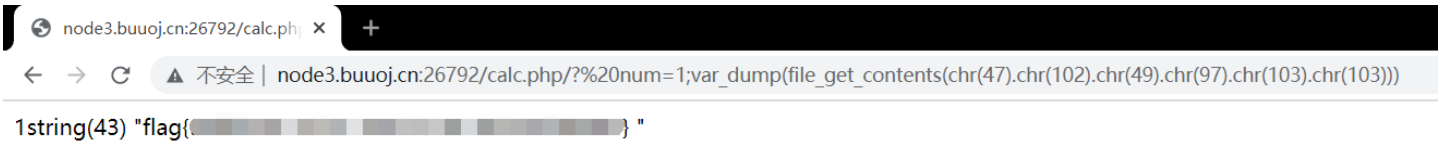
接下来，先使用scandir()函数去访问根目录下的所有文件，并使用void var_dump ()函数显示出来：



很遗憾被过滤了，使用ascii码绕过



查看根目录下文件，发现有f1agg，使用file_get_contents()访问。这次不浪费时间，同样使用ascii码绕过



得到flag。(文件名是f1agg，没看出来又检查了半天…)

3.总结

1. 第一次见这样的漏洞，利用php的字符串解析漏洞，能直接绕waf，还是很危险的
2. 熟悉了一些必要的函数，学会了php的ascii绕过（上次的还是sql的ascii绕过，用法有区别），涨姿势了
3. 做题要认真一些…f1agg我看成flagg，又查了好半天

4.参考资料

- [RoarCTF 2019]Easy Calc(http走私 && 利用PHP的字符串解析特性Bypass)_a3320315的博客-CSDN博客
- [RoarCTF 2019]Easy Calc_沐目的博客-CSDN博客
- Web-[RoarCTF 2019]Easy Calc - 高诺琪 - 博客园