目录

一、题目

1、源码

```php
class Challenge {
    const UPLOAD_DIRECTORY = './solutions/';
    private $file;
    private $whitelist;

    public function __construct($file) {
        $this->file = $file;
        $this->whitelist = range(1, 24);
    }

    public function __destruct() {
        if (in_array($this->file['name'], $this->whitelist)) {
            move_uploaded_file(
                $this->file['tmp_name'],
                self::UPLOAD_DIRECTORY . $this->file['name']
            );
        }
    }
}

$challenge = new Challenge($_FILES['solution']);
```

2、知识点

知识点	说明
const	定义常量
__construct()	构造函数/方法
__destruct()	析构函数/方法
in_array()	搜索数组中是否存在指定的值
move_uploaded_file()	把上传的文件移动到新位置
$_FILES	获取通过POST方法上传文件的相关信息
range()	创建一个包含指定范围元素的数组

3、解读

1）定义常量`UPLOAD_DIRECTORY`、私有属性`file`、私有属性`whitelist`。
2）创建构造函数，形参为`$file`。当类被实例化成对象时，将传入的实参赋值给`$file`，并定义`$whitelist`取值为`1-24`的数组。
3）创建析构函数，当对象被销毁时，如果`$file`的值在`1-24`之间存在（不考虑类型是否匹配），就将文件进行上传。
4）实例化对象，并接收`POST`中传入的文件信息。

4、分析

`in_array()`函数用来检测上传的文件名，由于该函数并未将第三个参数设置为`true`（强检查，对类型也进行匹配），导致攻击者可以通过构造的文件名来绕过匹配机制，从而达到任意文件上传目的。

5、利用

直接上传木马，木马名字中需包含`1-24`之间任意的一个数字，即可被`in_array()`函数进行接收。

二、CMS

1、源码-piwigo2.7.1

```
include\functions_rate.inc.php
```

```php
<?php
define('PHPWG_ROOT_PATH', './');
include_once (PHPWG_ROOT_PATH . 'include/common.inc.php');
include (PHPWG_ROOT_PATH . 'include/section_init.inc.php');
include_once (PHPWG_ROOT_PATH . 'include/functions_picture.inc.php');
// Check Access and exit when user status is not ok
check_status(ACCESS_GUEST);
// access authorization check
if (isset($page['category'])) {
    check_restrictions($page['category']['id']);
}
......
// +-----------------------------------------------------------------------+
// |                                actions                                |
// +-----------------------------------------------------------------------+

/**
 * Actions are favorite adding, user comment deletion, setting the picture
 * as representative of the current category...
 *
 * Actions finish by a redirection
 */
if (isset($_GET['action'])) {
    switch ($_GET['action']) {
        ......
        case 'rate': {
                include_once (PHPWG_ROOT_PATH . 'include/functions_rate.inc.php');
                rate_picture($page['image_id'], $_POST['rate']);
                redirect($url_self);
            }
......
```

include\config_default.inc.php

```php
<?php
function rate_picture($image_id, $rate)
{
    global $conf, $user;

    if (!isset($rate) or !$conf['rate'] or !in_array($rate, $conf['rate_items']))
    {
        return false;
    }
    $user_anonymous = is_autorize_status(ACCESS_CLASSIC) ? false : true;

    if ($user_anonymous and !$conf['rate_anonymous'])
    {
        return false;
    }
    ......
    if ($user_anonymous)
    {
        $query.= ' AND anonymous_id = \''.$anonymous_id.'\'';
    }
    pwg_query($query);
    $query = 'INSERT INTO '.RATE_TABLE.'(user_id,anonymous_id,element_id,rate,date) VALUES('
         .$user['id'].',\''.$anonymous_id.'\','.$image_id.','.$rate.',NOW());';
    pwg_query($query);//进行SQL查询

    return update_rating_score($image_id);
}
```

简化后的代码

```php
<?php
$rate = $_POST['rate'];
$conf['rate_items'] = array(0,1,2,3,4,5);
if (!isset($rate) or !$conf['rate'] or !in_array($rate, $conf['rate_items']))
{
    return false;
}
$query = 'INSERT INTO '.RATE_TABLE.'(user_id,anonymous_id,element_id,rate,date) VALUES('
         .$user['id'].','.'\''.$anonymous_id.'\','.$image_id.','.$rate.',NOW());';
pwg_query($query);//进行SQL查询

?>
```

2、知识点

知识点	说明
include_once()	包含并运行指定文件，如已经包含，则忽略本次包含
global	全局变量，类似于Java的public

3、解读

1）通过POST方法接收用户传入的 rate参数值，赋值给变量$rate
2）定义二级数组 $conf['rate_items'] 的值为 array(0,1,2,3,4,5)
3）如果用户没有通过POST传入 rate参数值，或者数组$conf中的rate为空，或者$conf['rate_items']中不包含用户传入的rate参数值，那么返回false
4）使用$rate等变量进行拼接数据库查询语句，赋值给$query
5）通过pwg_query()函数对$query执行数据库操作

4、分析

1）上面if判断中，使用了in_array函数进行判断，但是没有加第三个参数true作为强类型判断
2）也就是说用户只需要传入 带有 $conf['rate_items'] 数组中任意一个值，就可以绕过服务端的检测
3）用户传入的值会被带入到数据库操作语句中，并且其中的变量可控，那么此时就会造成SQL注入漏洞

5、利用

1）POST中传入 rate 的值：
1,1 and if(ascii(substr((select database()),1,1))=112,1,sleep(3)));#
2）拼接后的SQL语句：
INSERT INTO piwigo_rate (user_id,anonymous_id,element_id,rate,date) VALUES (2,'192.168.2',1,1,1 and if(ascii(substr((select database()),1,1))=112,1,sleep(3)));#,NOW()) ;

6、修复方案

漏洞原因是弱类型比较问题，可以使用强匹配进行修复。如在 in_array()中加入第三个参数 true，或者使用正则匹配处理变量。

7、参考链接

```
    https://github.com/hongriSec/PHP-Audit-
Labs/blob/master/Part1/Day1/files/README.md
```