[红日安全]代码审计Day1 - in_array函数缺陷

红日安全 / 2018-07-17 22:24:23 / 浏览数 10948 技术文章 技术文章 顶(5) 踩(0)

本文由红日安全成员： 七月火 编写，如有不当，还望斧正。

前言

大家好，我们是红日安全-代码审计小组。最近我们小组正在做一个PHP代码审计的项目，供大家学习交流，我们给这个项目起了一个名字叫 PHP-Audit-Labs 。现在大家所看到的系列文章，属于项目 第一阶段 的内容，本阶段的内容题目均来自 PHP SECURITY CALENDAR 2017 。对于每一道题目，我们均给出对应的分析，并结合实际CMS进行解说。在文章的最后，我们还会留一道CTF题目，供大家练习，希望大家喜欢。下面是 第1篇 代码审计文章：

Day 1 - Wish List

题目叫做愿望清单，代码如下：

```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']);
```


漏洞解析 ：

这一关卡考察的是一个任意文件上传漏洞，而导致这一漏洞的发生则是不安全的使用 in_array() 函数来检测上传的文件名，即上图中的第12行部分。由于该函数并未将第三个参数设置为 true ，这导致攻击者可以通过构造的文件名来绕过服务端的检测，例如文件名为 7shell.php 。因为PHP在使用 in_array() 函数判断时，会将 7shell.php 强制转换成数字7，而数字7在 range(1,24) 数组中，最终绕过 in_array() 函数判断，导致任意文件上传漏洞。（这里之所以会发生强制类型转换，是因为目标数组中的元素为数字类型）我们来看看PHP手册对 in_array() 函数的定义。

> in_array ：(PHP 4, PHP 5, PHP 7)
>
> 功能 ：检查数组中是否存在某个值
>
> 定义 ： `bool in_array ( mixed $needle , array $haystack [, bool $strict = FALSE ] )`

在 $haystack 中搜索 $needle ，如果第三个参数 $strict 的值为 TRUE ，则 in_array() 函数会进行强检查，检查 $needle 的类型是否和 $haystack 中的相同。如果找到 $haystack ，则返回 TRUE，否则返回 FALSE。

实例分析

本次实例分析，我们选取的是 piwigo2.7.1 版本。该版本由于SQL语句直接拼接 $rate 变量，而 $rate 变量也仅是用 in_array() 函数简单处理，并未使用第三个参数进行严格匹配，最终导致sql注入漏洞发生。下面我们来看看具体的漏洞位置。漏洞的入口文件在 include\functions_rate.inc.php 中，具体代码如下：

```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);
            }
......
```


当 $_GET['action'] 为 rate 的时候，就会调用文件 include/functions_rate.inc.php 中的 rate_picture 方法，而漏洞便存在这个方法中。我们可以看到下图第23行处直接拼接 $rate 变量，而在第2行使用 in_array() 函数对 $rate 变量进行检测，判断 $rate 是否在 $conf['rate_items'] 中， $conf['rate_items'] 的内容可以在 include\config_default.inc.php 中找到，为 `$conf['rate_items'] = array(0,1,2,3,4,5);`

```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);
}
```


由于这里（上图第6行）并没有将 in_array() 函数的第三个参数设置为 true ，所以会进行弱比较，可以绕过。比如我们将 $rate 的值设置成 1,1 and if(ascii(substr((select database()),1,1))=112,1,sleep(3)));# 那么SQL语句就变成：

```
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 da
```

这样就可以进行盲注了，如果上面的代码你看的比较乱的话，可以看下面简化后的代码：

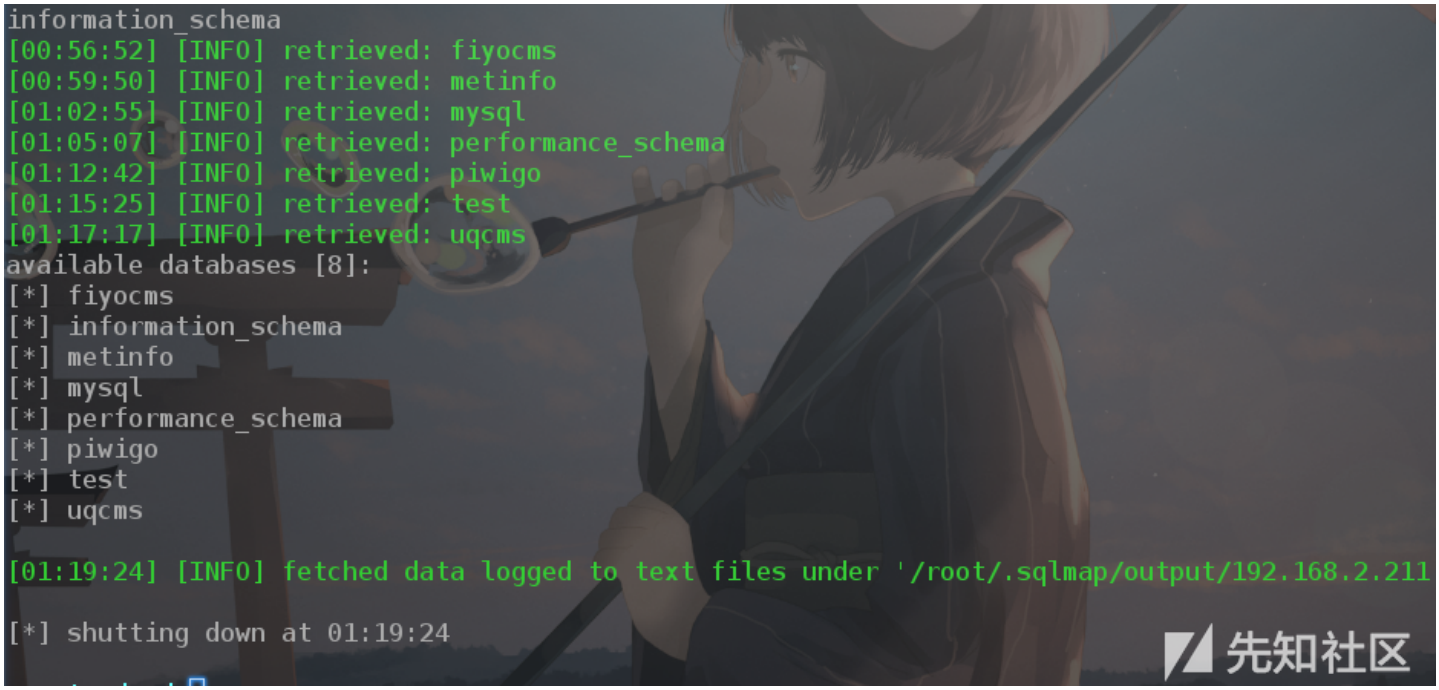
```php
<?php
$rate = $_POST['rate'];
$conf['rate_items'] = array(0,1,2,3,4,5);
if (!isset($rate) or !$conf['rate'] or !in_array($rate, $conf['rate_items']))
{
    return false;
}
$query = 'INSERT INTO '.RATE_TABLE.'(user_id,anonymous_id,element_id,rate,date) VALUES('
         .$user['id'].','.'\''.$anonymous_id.'\','.$image_id.','.$rate.',NOW());';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

?>
```


漏洞利用

接下来我们直接用sqlmap进行验证， payload 如下：

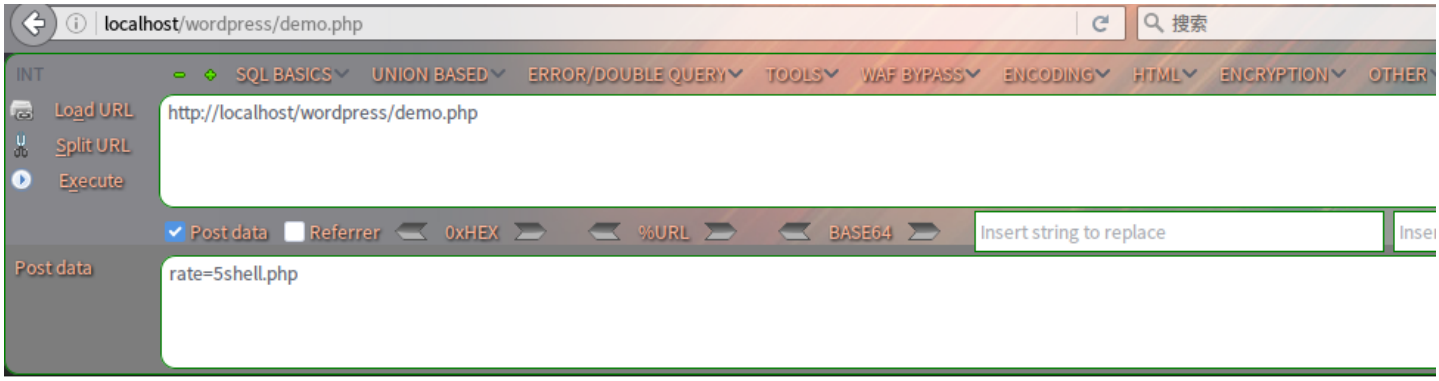
```
sqlmap -u "http://192.168.2.211/piwigo/picture.php?/1/category/1&action=rate" --data "rate=1" --dbs --batch
```



修复建议

可以看到这个漏洞的原因是弱类型比较问题，那么我们就可以使用强匹配进行修复。例如将 in_array() 函数的第三个参数设置为 true ，或者使用 intval() 函数将变量强转成数字，又或者使用正则匹配来处理变量。这里我将 in_array() 函数的第三个参数设置为 true ，代码及防护效果如下：

```php
<?php
$rate = $_POST['rate'];
$conf['rate_items'] = array(0,1,2,3,4,5);
if (in_array($rate, $conf['rate_items']))
{
    echo "\$rate is in \$conf['rate_items']"."<br>";
}
else{
    echo "\$rate is not in \$conf['rate_items']"."<br>";
}
//修复代码如下
if (in_array($rate, $conf['rate_items'], true))
{
    echo "\$rate is in \$conf['rate_items']"."<br>";
}
else{
    echo "\$rate is not in \$conf['rate_items']"."<br>";
}
?>
```



$rate is in $conf['rate_items'] 修复前
$rate is not in $conf['rate_items'] 修复后

结语

看完了上述分析，不知道大家是否对 in_array() 函数有了更加深入的理解，文中用到的CMS可以从 这里
下载，当然文中若有不当之处，还望各位斧正。如果你对我们的项目感兴趣，欢迎发送邮件到 hongrisec@gmail.com 联系我们。Day1
的分析文章就到这里，我们最后留了一道CTF题目给大家练手，题目如下：

```
//index.php
<?php
include 'config.php';
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
   die("■■■■: ");
}

$sql = "SELECT COUNT(*) FROM users";
$whitelist = array();
$result = $conn->query($sql);
if($result->num_rows > 0){
   $row = $result->fetch_assoc();
   $whitelist = range(1, $row['COUNT(*)']);
}

$id = stop_hack($_GET['id']);
$sql = "SELECT * FROM users WHERE id=$id";

if (!in_array($id, $whitelist)) {
   die("id $id is not in whitelist.");
}

$result = $conn->query($sql);
if($result->num_rows > 0){
   $row = $result->fetch_assoc();
   echo "<center><table border='1'>";
   foreach ($row as $key => $value) {
       echo "<tr><td><center>$key</center></td><br>";
       echo "<td><center>$value</center></td></tr><br>";
   }
   echo "</table></center>";
}
else{
   die($conn->error);
}

?>

//config.php
<?php
$servername = "localhost";
$username = "fire";
$password = "fire";
$dbname = "day1";

function stop_hack($value){
```

```
    $pattern = "insert|delete|or|concat|concat_ws|group_concat|join|floor|\/\*|\*|\.\.\/|\.\/|union|into|load_file|outfile|dump
    $back_list = explode("|",$pattern);
    foreach($back_list as $hack){
        if(preg_match("/$hack/i", $value))
            die("$hack detected!");
    }
    return $value;
}
?>

# ■■CTF■■■■■sql■■
create database day1;
use day1;
create table users (
id int(6) unsigned auto_increment primary key,
name varchar(20) not null,
email varchar(30) not null,
salary int(8) unsigned not null );

INSERT INTO users VALUES(1,'Lucia','Lucia@hongri.com',3000);
INSERT INTO users VALUES(2,'Danny','Danny@hongri.com',4500);
INSERT INTO users VALUES(3,'Alina','Alina@hongri.com',2700);
INSERT INTO users VALUES(4,'Jameson','Jameson@hongri.com',10000);
INSERT INTO users VALUES(5,'Allie','Allie@hongri.com',6000);

create table flag(flag varchar(30) not null);
INSERT INTO flag VALUES('HRCTF{1n0rrY_i3_Vu1n3rab13}');
```

题解我们会阶段性放出，如果大家有什么好的解法，可以在文章底下留言，祝大家玩的愉快！

上一篇：ChakraCore-JSRT 下一篇：Blackgear复出，使用社交媒...

1. 13 条回复

红日安全 2018-07-17 22:30:05

[红日安全]代码审计Day1 - in_array函数缺陷

[红日安全]代码审计Day2 - filter_var函数缺陷

[红日安全]代码审计Day3 - 实例化任意对象漏洞

[红日安全]代码审计Day4 - strpos函数缺陷

[红日安全]代码审计Day5 - escapeshellarg与escapeshellcmd函数

[红日安全]代码审计Day6 - preg_replace之正则表达式绕过

[红日安全]代码审计Day7 - parse_str函数缺陷

[红日安全]代码审计Day8 - preg_replace函数之命令执行

[红日安全]代码审计Day9 - str_replace函数缺陷

这是我们已经完成的部分，欢迎感兴趣的朋友加入我们，一起完善这个项目。

1 回复Ta

chybeta 2018-07-17 23:19:05

感谢分享。这里再附上关于 in_array 的一段代码，可以明确看到非严格模式与严格模式下的区别：

```
<?php

$array = array(
   'egg' => true,
   'cheese' => false,
   'hair' => 765,
   'goblins' => null,
   'ogres' => 'no ogres allowed in this array'
);

// Loose checking -- return values are in comments

// First three make sense, last four do not

var_dump(in_array(null, $array)); // true
var_dump(in_array(false, $array)); // true
var_dump(in_array(765, $array)); // true
var_dump(in_array(763, $array)); // true
var_dump(in_array('egg', $array)); // true
var_dump(in_array('hhh', $array)); // true
var_dump(in_array(array(), $array)); // true

// Strict checking
var_dump(in_array(null, $array, true)); // true
var_dump(in_array(false, $array, true)); // true
var_dump(in_array(765, $array, true)); // true
```

```
var_dump(in_array(763, $array, true)); // false
var_dump(in_array('egg', $array, true)); // false
var_dump(in_array('hhh', $array, true)); // false
var_dump(in_array(array(), $array, true)); // false

?>
```

1 回复Ta

灰度 2018-07-18 14:15:58

这是一个好项目，感谢分享

0 回复Ta

烧包包儿 2018-07-19 09:21:36

你好，下载地址失效了

能不能发布一个新的下载地址？

0 回复Ta

mochazz 2018-07-19 11:35:27

@159****0259 文中的piwigo2.7.1可以下载，需要挂梯子才能下载

0 回复Ta


xuanhu****@qq.co 2018-08-24 09:42:24

能在订阅号转载 这个系列吗

0 回复Ta


红日安全 2018-08-24 21:01:01

@xuanhu****@qq.co 可以啊，开头注明转载自先知社区即可。

0 回复Ta


xuanhu****@qq.co 2018-08-25 10:03:06

@红日安全 好的，多谢

0 回复Ta

遗忘城 2018-09-07 14:03:51

你好，下载地址失效了

能不能发布一个新的下载地址？

0 回复Ta

红日安全 2018-11-02 14:17:18

@遗忘城
下载链接只是被墙了，是可以下载的。或者你可以直接从我们的项目上下载：https://github.com/hongriSec/PHP-Audit-Labs/blob/master/Part1/Day1/files/piwig

0 回复Ta

白猫 2018-12-03 15:00:39

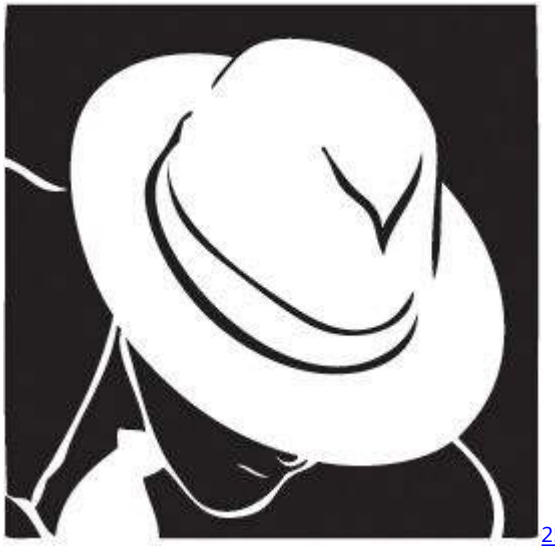
非常棒的项目,感谢分享

0 回复Ta


LJH 2019-04-05 16:13:14

棒极了,谢谢分享

0 回复Ta


2524****@qq.com 2019-11-04 16:04:26

兄弟，看到你发到文章，挺感兴趣，可有兴趣录一些视频，有兴趣发邮件到test@vvsec.cn，发邮箱留下联系方式就行。

0 回复Ta

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