目录

- 一、题目
  - 1、源码
  - 2、知识点
  - 3、解读
  - 4、分析
  - 5、利用
- 二、CMS
  - 1、源码-CmsEasy 5.5
  - 2、知识点
  - 3、解读
  - 4、分析
  - 5、利用
  - 6、修复方案
  - 7、参考链接

一、题目

1、源码

```php
header("Content-Type: text/plain");

function complexStrtolower($regex, $value) {
    return preg_replace(
        '/(' . $regex . ')/ei',
        'strtolower("\\1")',
        $value
    );
}

foreach ($_GET as $regex => $value) {
    echo complexStrtolower($regex, $value) . "\n";
}
```

2、知识点

知识点	说明
header()	向客户端发送原始的HTTP报头
foreach($_GET as $regex => $value) {}	foreach循环遍历，将$_GET中接受的键和值分别传递给$regex和$value
preg_replace()	第一个参数为匹配项，第二个参数为替换值，第三个参数为对象
{}	用法一：处理处理{}中变量；用法二：字符串变量后面有{}，作为数组处理

知识点	说明
${phpinfo()}	phpinfo()会被当作变量先处理，执行成功后，变成${1}（phpinfo()执行成功后返回boolean值，也就是true，用1表示）
\1	获取缓冲区的第一个子匹配
preg_match()	用于执行一个正则表达式匹配

3、解读

1）第11行，函数foreach()遍历GET方法传入的数据，并将键和值依次传递给 $regex和$value，调用函数complexStrtolower()，传入这两个变量作为实参。

2）第3行，函数complexStrtolower()中，使用函数preg_replace()将$regex在$value中匹配，然后将匹配到的值替换成'strtolower'，并返回结果。

4、分析

1）preg_replace()中，/e模式表示字符串会作为代码执行，也就是说从$value中匹配到的$regex会被执行。

2）preg_replace()函数中的第二个参数为固定字符串 'strtolower("\\1")'。

3）\\1也就是\1，因为第一个\会被当作转义字符处理。\1是反向引用中的知识，表示如果正则表达式模式或部分模式两边添加圆括号，将导致相关匹配存储到一个临时缓冲区中，所捕获到的每个子匹配都按照空格为间隔，从左到右的顺序被存储，可按照1~99序号取出。

4）也就是说$regex中的一个值会被取出并执行。

5、利用

```
?\S*={${phpinfo()}}
```

二、CMS

1、源码-CmsEasy 5.5

```
/lib/tool/form.php
```

```php
function getform($name,$form,$field,$data) {
    if (get('table') &&isset(setting::$var[get('table')][$name]))
        $form[$name]=setting::$var[get('table')][$name];
    if (get('form') &&isset(setting::$var[get('form')][$name]))
        $form[$name]=setting::$var[get('form')][$name];
    if (isset($form[$name]['default']))
        $form[$name]['default']=preg_replace('/\{\?([^}]+)\}/e',"eval('return $1;')",$form[$name]['default']);
    if (!isset($data[$name]) &&isset($form[$name]['default']))
        $data[$name]=@$form[$name]['default'];
    if (preg_match('/templat/',$name) &&empty($data[$name]))
        $data[$name]=@$form[$name]['default'];
```

Cache/template/default/manage/guestadd.php

```
<div class="hid_box">
    <strong><?php echo lang(addcategory);?></strong>
    <div class="hbox" style="background:none;">
        <?php echo form::getform('catid',$form,$field,$data);?>
    </div>
</div>
```

lib/table/archive.php

```
function get_form() {
    return array(
        'catid'=>array(
            'selecttype'=>'select',
            'select'=>form::arraytoselect(category::option(0,'tolast')),
            'default'=>get('catid'),
            'regex'=>'/\d+/',
            'filter'=>'is_numeric',
            ),
        'typeid'=>array(
            'selecttype'=>'select',
            'select'=>form::arraytoselect(type::option(0,'tolast')),
            'default'=>get('typeid'),
            'regex'=>'/\d+/',
            'filter'=>'is_numeric',
            ),
            ......
        'tag_option'=>array(
            'selecttype'=>'select',
            'select'=>form::arraytoselect(tag::getTags()),
            ),
    );
}
```

lib/tool/front_class.php

```
function get($var) {
    if (front::get($var))
        return front::get($var);
    else if (front::post($var))
        return front::post($var);
    else if (config::get($var))
        return config::get($var);
    else if (session::get($var))
        return session::get($var);
}
```

```php
static function get($var) {
    if (isset(self::$get[$var]))
        return self::$get[$var];
    else
        return false;
}
static function post($var) {
    if (isset(self::$post[$var]))
        return self::$post[$var];
    else
        return false;
}
```

```php
final class front {
    ......
    static $get;
    static $post;
    ......
    function __construct() {
        ......
        self::$get=$_GET;
        self::$post=$_POST;
        ......
    }
}
```

lib/default/manage_act.php

```php
final class front {
    ......
    static $get;
    static $post;
    ......
    function __construct() {
        ......
        self::$get=$_GET;
        self::$post=$_POST;
        ......
    }
}
```

2、知识点

Null

3、解读

1）图1，函数getform()中，第6行，如果设置了$form[$name]['default']的值，就会调用函数preg_replace()，并且使用/e模式，也就是字符串可被执行。

2）图2，跟进函数getform()被调用的位置到guestadd.php，第4行传入的 'catid' 也就是对应了函数getform()中的$name参数，也就是这个'catid'存在命令执行功能。

3）图3，跟进catid到archive.php，文件中的函数get_form()对其进行了定义，并return了一个二维数组，其中包含 'catid'等数组，在catid数组中也包含了default字段，也就对应着图1中的$form[$name]['default']，值为get('catid')。

4）图4，跟进get()方法到front_class.php，其中调用了front类的静态函数get和post进行判断，并返回对应的值。

5）图5，跟进front类的静态函数get()和post()，可以看到值对应的是self类下的静态变量$get和$post。

6）图6，跟进$get和$post，可以看到其是front类中定义的静态属性，并对应的是$_GET和$_POST。也就是说图1函数getform()中$form[$name]['default']的值是用户可控的。

7）图7，跟进get_form()调用的位置到manage_act.php中，这里直接调用了函数get_form()，并通过View模板直接传到前台页面中。

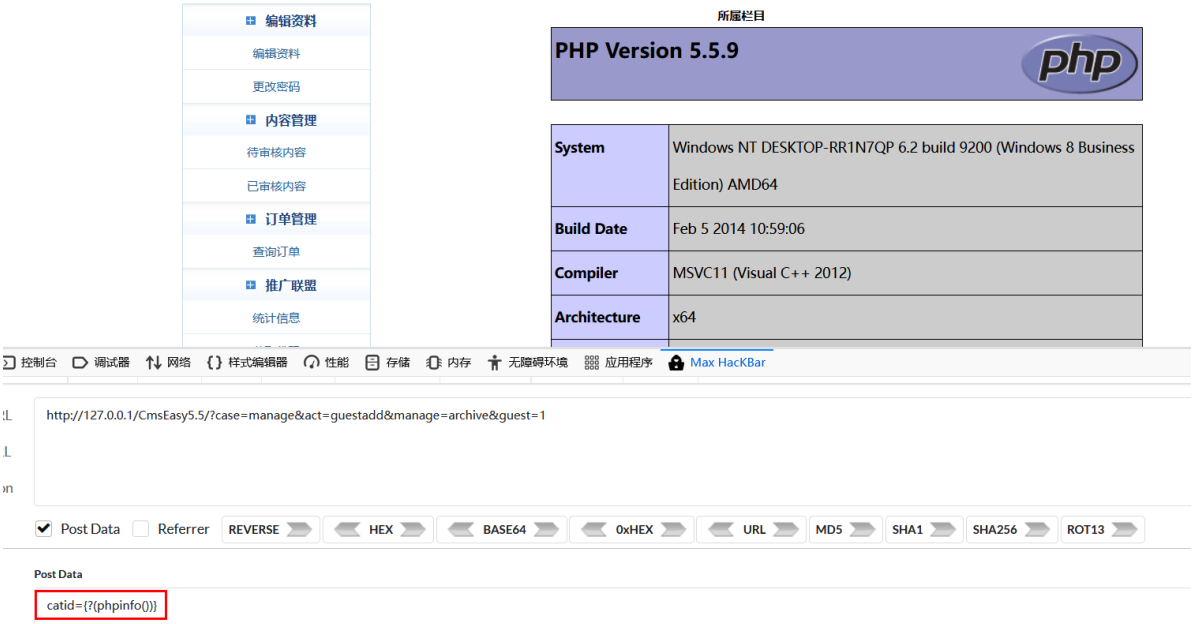
4、分析

传入catid的值为要执行的代码，即可成功造成命令执行攻击。

5、利用

1）打开网站用户登录界面，点击游客投稿

2）POST中传入catid参数，由于函数preg_replace()是需要匹配到 /\{\?([^}]+)\}/e，也就是匹配{?(任意内容)}作为命令执行，构造Payload：catid={?(phpinfo())}，成功执行。



6、修复方案

该漏洞命令执行的根本就是开启了函数preg_replace()的/e模式，如果匹配成功就会造成代码执行。所以避免使用/e模式即可。

7、参考链接

https://github.com/hongriSec/PHP-Audit-Labs/blob/master/Part1/Day8/files/README.md