BUUCTF-WarmUp

原创

闭关不更新 于 2021-11-20 21:38:59 发布 3467 收藏 2
文章标签： 网络 安全 php

本文链接：https://blog.csdn.net/weixin_45441315/article/details/121445319

BUUCTF-WarmUp(代码审计)

1、打开靶场后，查看源码即可看到

```
<!--source.php-->
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <!--source.php-->

    <br><img src="https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg" /></body>
</html>
```


2、进入sourcep.php
代码如下

```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\" />";
    }
?>
```

3.审计php代码

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

发现另一个地址，打开以后提示我们flag在ffffllllaaaagggg里面

flag not here, and flag in ffffllllaaaagggg

4、继续审计php代码

PHP is_string() 函数

```
if (! empty($_REQUEST['file'])  //$_REQUEST['file']值非空
        && is_string($_REQUEST['file'])  //$_REQUEST['file']值为字符串
        && emmm::checkFile($_REQUEST['file'])  //能够通过checkFile函数校验
    ) {
        include $_REQUEST['file'];  //包含$_REQUEST['file']文件
        exit;
    } else {
        echo "<br><img src=\"https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg\" />";
        //打印图片
```

这段代码告诉们需要满足三个条件

1、file参数值非空

2、file参数为字符串

3、能通过函数chekFile()检验

否则打印图片

5、查看checkFile()函数

PHP in_array() 函数

PHP mb_substr() 函数

```
highlight_file(__FILE__); //打印代码
class emmm  //定义emmm类
{
    public static function checkFile(&$page)//将传入的参数赋给$page
    {
        $whitelist = ["source"=>"source.php","hint"=>"hint.php"];//声明$whitelist（白名单）键值对数组，
        if (! isset($page) || !is_string($page)) {//若$page变量不存在或非字符串
            echo "you can't see it";//打印"you can't see it"
            return false;//返回false
        }

        if (in_array($page, $whitelist)) {//若$page变量存在于$whitelist数组中
            return true;//返回true
        }

        $_page = mb_substr(//该代码表示截取$page中'?'前部分，若无则截取整个$page
            $page,
            0,
            mb_strpos($page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {//若$_page变量存在于$whitelist数组中
            return true;
        }

        $_page = urldecode($page);//url解码$page
        $_page = mb_substr(//该代码表示截取$_page中'?'前部分，若无则截取整个$page
            $_page,
            0,
            mb_strpos($_page . '?', '?')
        );
        if (in_array($_page, $whitelist)) {//若$_page变量存在于$whitelist数组中
            return true;
        }
        echo "you can't see it";
        return false;
    }
}
```

6、可以看出函数里面含有4个

第一个if语句对变量进行检验，要求$page为字符串，否则返回false
第二个if语句判断$page是否存在于$whitelist数组中，存在则返回true
第三个if语句判断$page中?之前的字符串是否存在于$whitelist数组中，存在则返回true
第四个if语句判断$page参数url解密后的$_page中?之前的字符串是否存在于$whitelist数组中，存在则返回true
都不满足，最后返回false
从中可以看出
有三个if语句可以返回true，第二个语句直接判断$page，不可用
第一个if语句只要满足了外部函数的if语句这个就不会成立，所以不用管
第四个if语句事多余的，因为只要第三个if语句成立，就会返回true
那么我们应该如何构造payload呢
由in_array($_page, $whitelist)我们可以看出payload里面一定要包含source.php或者hint.php
这个时候如果我们在file内容中加入?，就能实现截断的效果，这个时候page将会验证?前方的str是否包含在array中，因此就能实现checkFile函数的绕过。
这个时候一定能过checkFile函数。
但是我们的include文件读取函数却是直接将file进行了读取，这个时候file的内容是无效的，因此什么东西也读不出来。
我们通过hint.php可以知道，flag隐藏在ffffllllaaaagggg中。
所以现在我们的目标是通过拼接的方式构造出读取flag的绕过姿势。
url里面我们比较常见的是#绕过，即#后面的内容不会被读取执行。
但是我们究竟该怎么样让前面的hint.php%253f不被放到include中影响我们flag的正常读取呢？
我们知道如果直接放入hint.php?ffffllllaaaaggg会通过checkFile的验证返回true，但是毫无疑问这种错误的文件名是无法被include正确读取的。
这个时候我们就要用到一个关键符号：/
例如：include 'hint.php/flag.txt'
hint.php无法被正确读取，这个时候/后面的flag.txt就会被include函数读取并解析
因此我们可以构建payload：
?file=source.php?/ffffllllaaaagggg

我们需要将目录回退四次。
因此构建新的payload：
?file=source.php?../../../../../ffffllllaaaagggg

?> flag{2ba8cc33-fc78-49a4-b864-5bfe85cd5e2d}