polarD&N靶场web部分

一、简单rce

登陆页面，

```
PolarD&N CTF

*/
highlight_file(__FILE__);
function no($txt){
  if(!preg_match("/cat|more|less|head|tac|tail|nl|od|vim|uniq|system|proc_open|shell_exec|popen| /i", $txt)){
  return $txt;}
  else{
die("what's up");}}
$yyds=($_POST['yyds']);
if(isset($_GET['sys'])&&$yyds=='666'){
 eval(no($_GET['sys']));
 }
 else
  {echo "nonono";
}
?> nonono
```

简单分析，过滤了以下命令

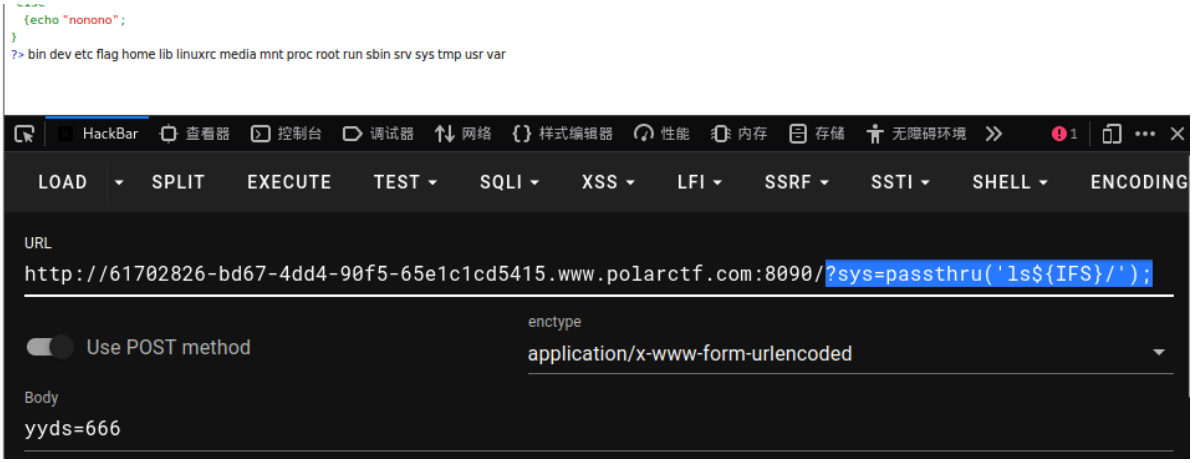
```
cat|more|less|head|tac|tail|nl|od|vim|uniq|system|proc_open|shell_exec|popen|
```

需要get传入sys参数，post传入“yyds=666”

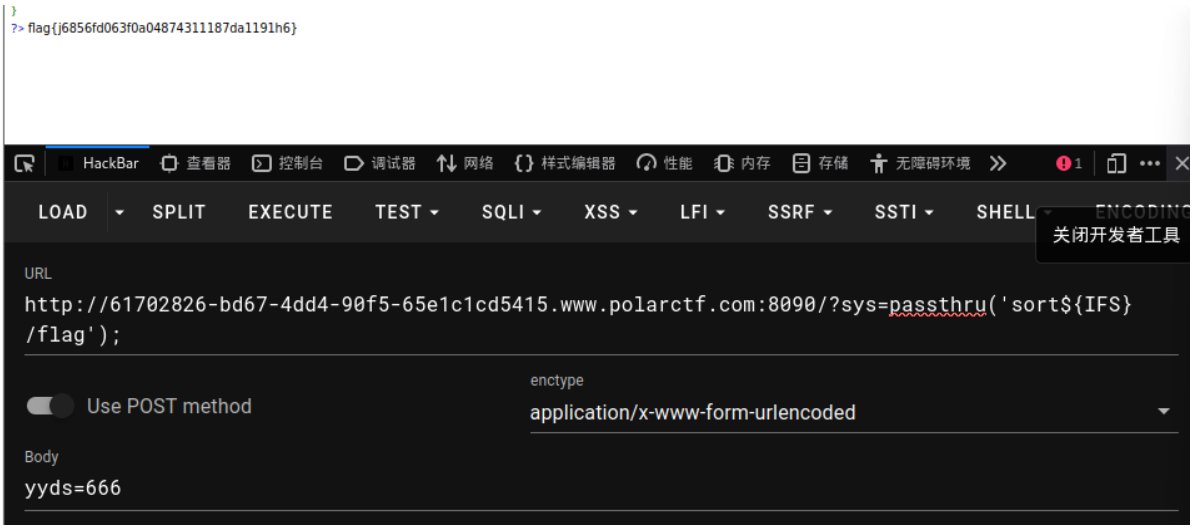
进行传参后，进行目录查询

发现此目录下仅有一个php文件，查询一下根目录，因为空格过滤，所以需要空格绕过，get传入下面代码

```
?sys=passthru('ls${IFS}/');
```



发现flag，因为过滤，所以需要利用sort进行输出flag



得到flag

总结一下

此题考了命令过滤，以及命令执行函数

命令执行函数
system() 输出并返回最后一行shell结果。（被过滤！）
exec() 不输出结果，返回最后一行shell结果，所有结果可以保存到一个返回的数组里面。
shell_exec() 将字符串作为OS命令执行，需要输出执行结果，且输出全部的内容。（被过滤！）
passthru() 只调用命令，把命令的运行结果原样地直接输出到标准输出设备上。（替换system）
popen()/proc_open() 该函数也可以将字符串当作OS命令来执行，但是该函数返回的是文件指针而非命令执行结果。该函数有两个参数。（被过滤！）

输出函数
cat函数 由第一行开始显示内容，并将所有内容输出（被过滤！）
tac函数 从最后一行倒序显示内容，并将所有内容输出（被过滤！）
nl 类似于cat -n，显示时输出行号（被过滤！）
more 根据窗口大小，一页一页的现实文件内容（被过滤！）
less 和more类似，但其优点可以往前翻页，而且进行可以搜索字符（被过滤！）
head 只显示头几行（被过滤！）
tail 只显示最后几行（被过滤！）
sort 文本内容排列
uniq 可检查文本文件中重复出现的行列。
od od （Octal Dump）命令用于将指定文件内容以八进制、十进制、十六进制、浮点格式或ASCII编码字符方式显示，通常用于显示或查看文件中不能直接显示在终端的字符。od命令系统默认的显示方式是八进制。

空格绕过：
${IFS}
{IFS}$9
IFS9
重定向符：<>（但是不支持后面跟通配符）
%09 水平制表符
%0a 回车
%0d 换行