\_FILE\_ 文件路径和文件名

PHP\_VERSION php版本

**strcmp**

int strcmp ( string $str1 , string $str2 )1

参数 str1 第一个字符串。str2 第二个字符串。如果 str1 小于 str2 返回 <0； 如果 str1 大于 str2 返回> 0；如果两者相等，返回 0。区分大小写

绕过的漏洞就是数组或者一个 object 即可，就可以让比较直接变成0,就是相当于两个都相等，从而绕过验证。

**empty判断字符是否为空**

bool empty ( mixed $var )

**is\_numeric**

**PHP 提供了 is\_numeric 函数，用来变量判断是否为数字。但是函数的范围比较广泛，不仅仅是十进制的数字。**

bool is\_numeric ( mixed $var )

如果指定的变量是数字和数字字符串则返回 TRUE，否则返回 FALSE，注意浮点型返回 1，即 TRUE。

is\_numeric 函数对于空字符 %00，无论是 %00 放在前后都可以判断为非数值，而 %20 空格字符只能放在数值后。

**PHP preg\_match\_all() 函数**

1,方括号 [n] 在目标寻找字符n

2,连字符[A-Za-z] 匹配全部任意字符

2，点字符在正则表达式中是一个通配符，".er" 匹配所有三个字符中结尾是er的字符

’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’

认证email的正则表达

^[a-zA-Z0-9\_-]+@[a-zA-Z0-9\_-]+(\.[a-zA-Z0-9\_-]+)+$

[PHP 正则表达式(PCRE)](https://www.runoob.com/php/php-pcre.html)

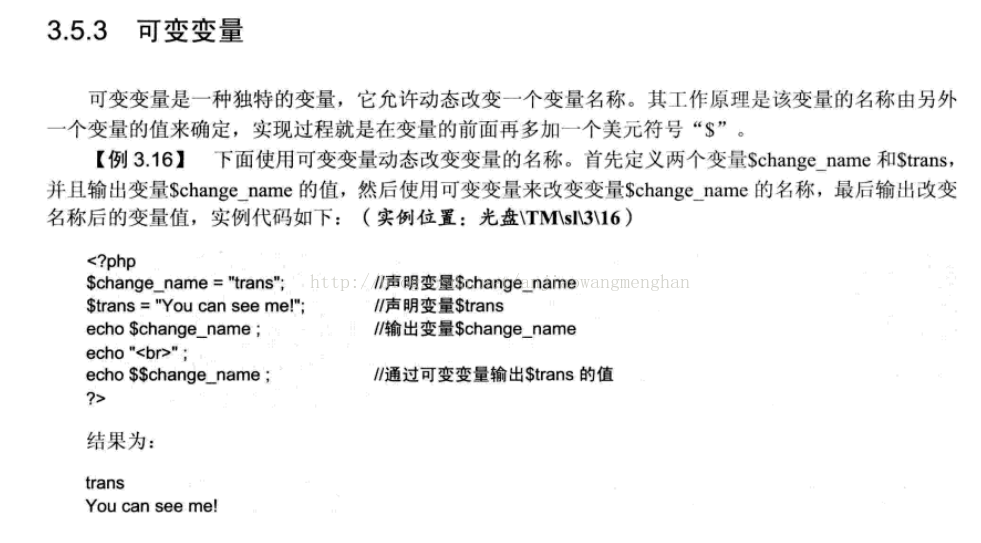
preg\_match\_all 函数用于执行一个全局正则表达式匹配

<?php

show\_source("test.php");

?>

原理: flag\_1234.php=\*\_1234.php (绕过)



正则分析 /key.\*key.{4,7}key:/./(.\*key)[a-z][[:punct:]]/

key就是普通的字符

.是除了换行的任意字符，假设就是a \*匹配0或多个正则表达式，假设有2个

key又是一个普通字符

.{4,7}就是匹配4-7个任意字符(包括4，不包括7)，假设有5个a

key普通字符

:又是普通字符

\反斜杠进行转义，此处取/的原意

.任意字符，假设取a

/同上，取/原意

()改变逻辑顺序，此处不影响，按.\*key顺序正则匹配

假设两个a和key

[a-z]取a-z中任意一个字符，假设取a

[:punct:]就是特殊字符的意思，[[:punct:]]就是在特殊字符里取一个，假设是;(当然也可以是@#等等)

**json\_encode()函数**

**格式转换**

**echo 输出不了数组用 json\_encode()转换**

**如 echo json\_encode($data);**

**serialize序列化函数**

**对象转序列化**

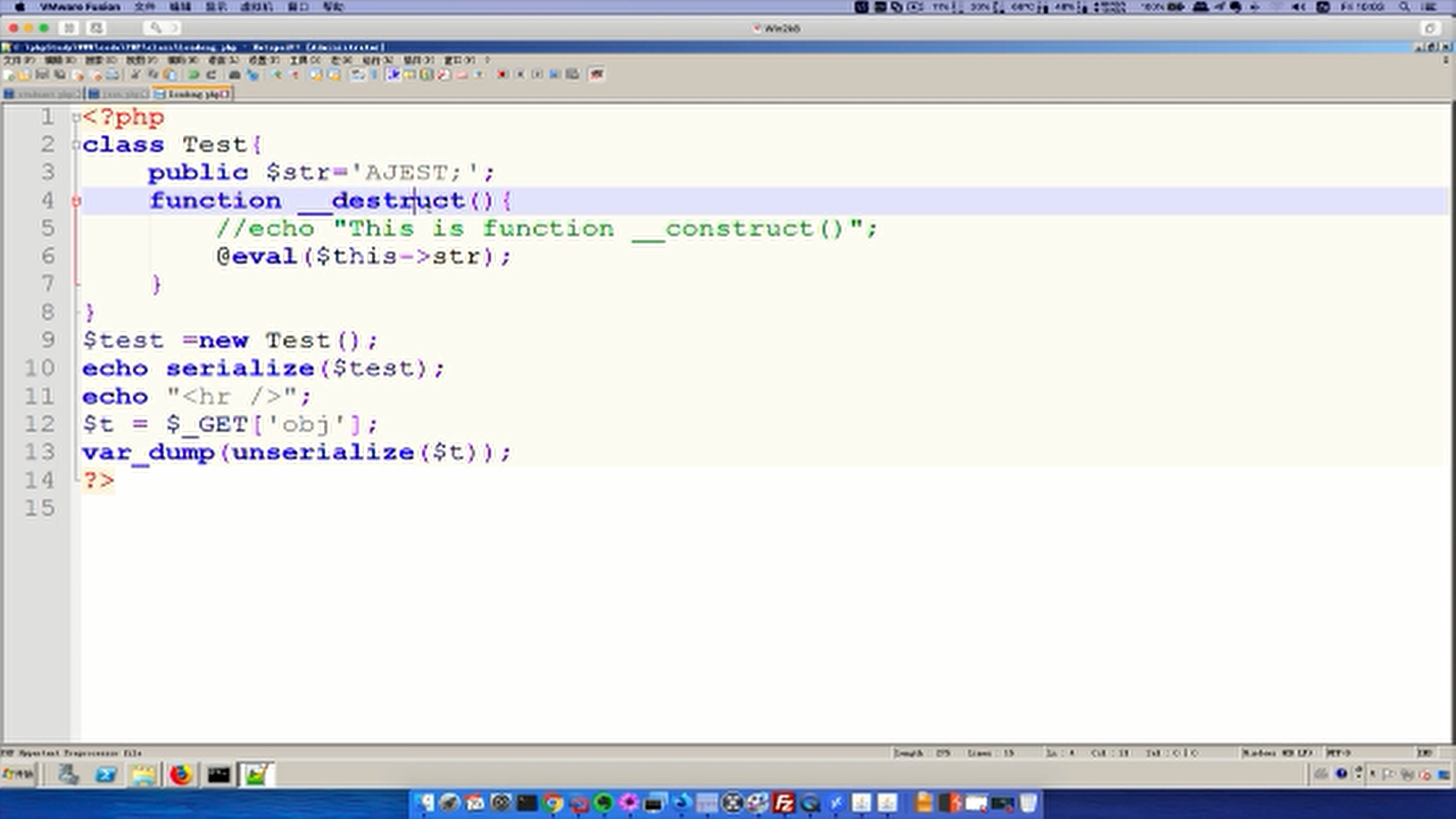
**如 echo serialize($data);**

**unserialize反序列化**

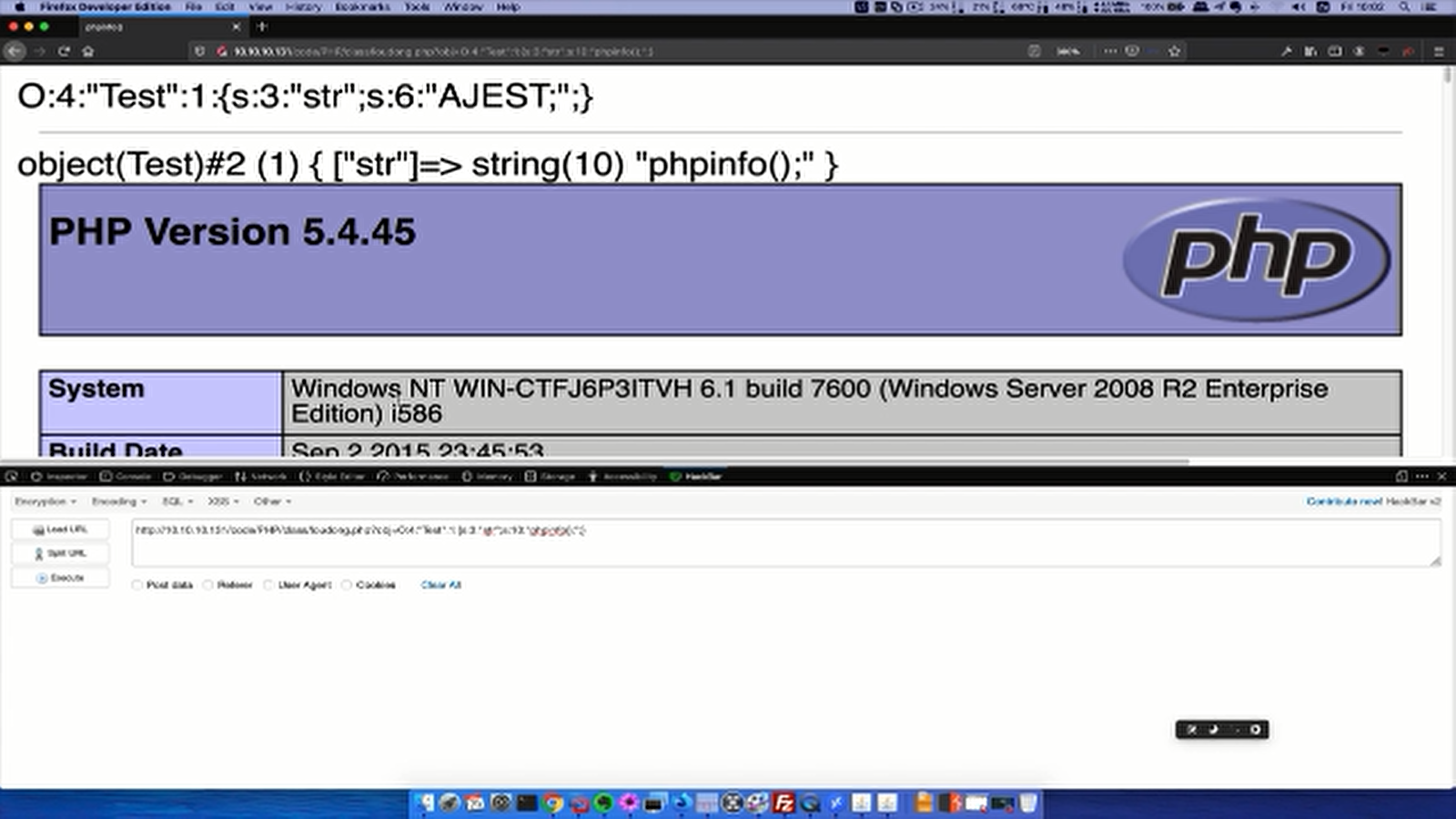
**序列化转对象**

**在序列化转对象时候如果没过滤并且有执行函数就可以传php代码**

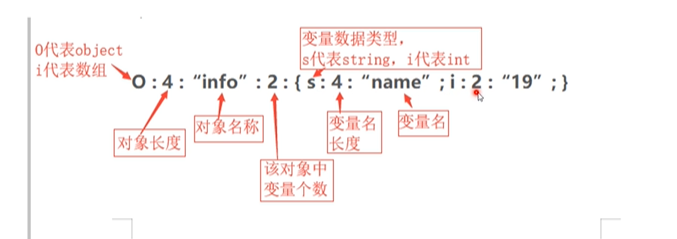
**如**



**有@eval()函数和$\_GET变量利用**



**注意长度也要符合**



2 要绕过正则表达式

(preg\_match(’/[oc]:\d+:/i’, $var))

而正则匹配的规则是: 在不区分大小写的情况下 ， 若字符串出现 “o:数字” 或者 "c:数字’ 这样的格式 ， 那么就被过滤 .很明显 ， 因为 serialize() 的参数为 object ，因此参数类型肯定为对象 " O " ， 又因为序列化字符串的格式为 参数格式:参数名长度 ， 因此 " O:4 " 这样的字符串肯定无法通过正则匹配

绕过

而O:+4没被过滤说明绕过了过滤而且最后的值不变。

//wakeup绕过 在原来的1个元素加一个数

<?php

class Demo {

private $file = 'index.php';

public function \_\_construct($file) {

$this->file = $file;

}

function \_\_destruct() {

echo @highlight\_file($this->file, true);

}

function \_\_wakeup() {

if ($this->file != 'index.php') {

//the secret is in the fl4g.php

$this->file = 'index.php';

}

}

}

$A = new Demo ('fl4g.php');//创建对象

$C = serialize($A); //对对象A进行序列化

$C = str\_replace('O:4','O:+4',$C); //绕过正则表达式过滤 #将O:4 替换O:+4

$C = str\_replace(':1:',':2:',$C); //wakeup绕过

var\_dump($C);

var\_dump(base64\_encode($C)); //base64加密

?>

**getimagesize检查文件大小，一般用于文件上传**

**preg\_replace()函数**

替换函数

例如：

<?php

$str=preg\_replace('/a/,'b',"aaaaaaaaa'')

echo $str

?>

会将aaaaaaaaa里面包含a的全部替换成b

一般用于过滤非法字符

如：

<?php

if(isset($\_GET['code'])){

$code=$\_GET['code'];

preg\_replace("\/[(.\*)\}/e",'\\1',$code) #正则匹配将中括号的值转换成1

}

?>

trim函数：去除首尾空格

file\_get\_contents函数：读取文件内容



**PHP内置过滤函数**

php.ini 魔术引号：

magic\_quotes\_gpc = On

magic\_quotes\_runtime =Off

本特性已自 PHP 5.3.0 起废弃并将自 PHP 5.4.0 起移除。

当打开时，所有的 '（单引号），"（双引号），\（反斜线）和 NULL 字符都会被自动加上一个反斜线进行转义。这和 addslashes() 作用完全相同

**addslashes()**

将单引号，双引号，反斜杠，NULL进行加反斜杠转义

用法：addslashes($x) #就可以将$x之前不管是使用什么传入方式发现可以转义就在前面添加反斜杠转义

<?php

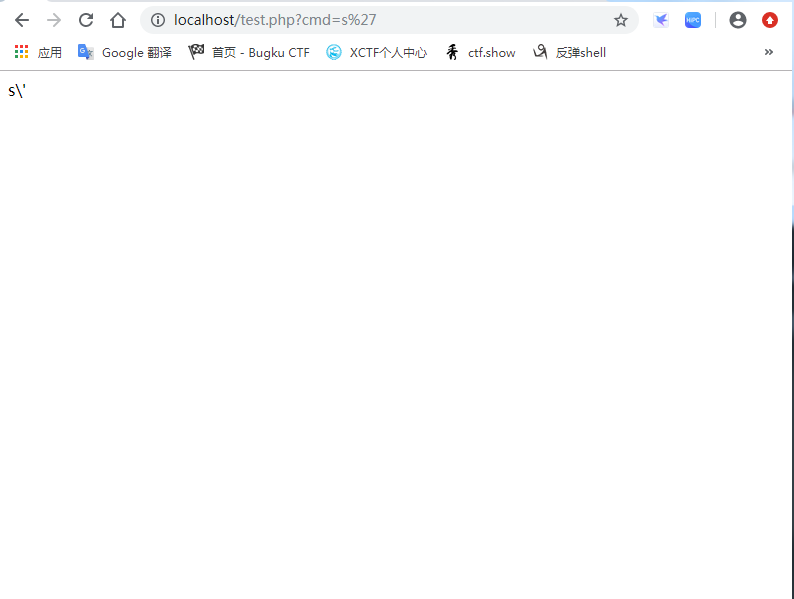
    header("content-type:text/html;charset=utf-8");

    $x = $\_GET['cmd'];

    $y=addslashes($x);

    print\_r($y);

?>

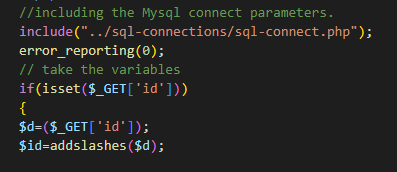


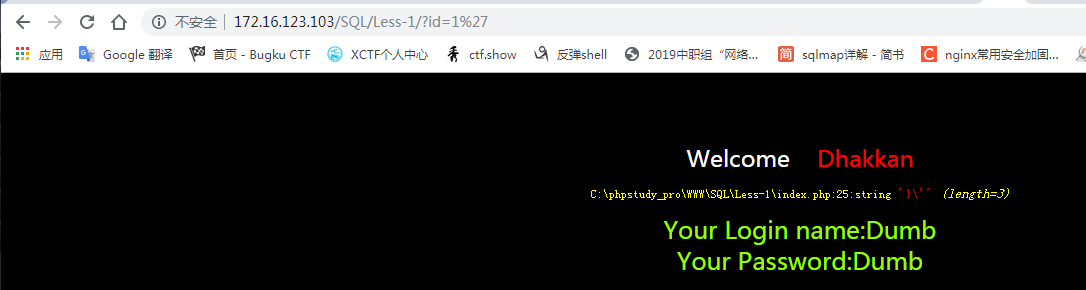
**stripslashes（）**

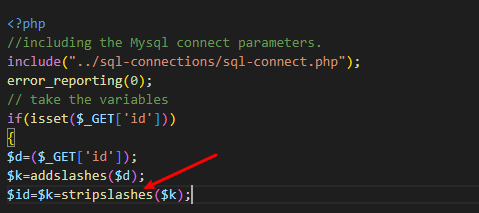
该方法会删除所有的反斜杠

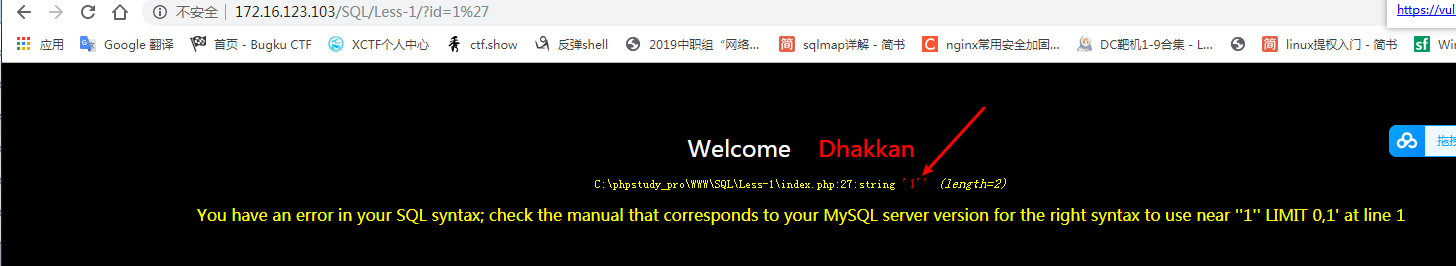
防止文件包含的绝技

stripslashes使用方法：









**htmlspecialchars()函数**

**防止xss注入的绝招**



将危险字符进行html实体化使用方法,无效编码就返回空：

htmlspecialchars($x);

<?php

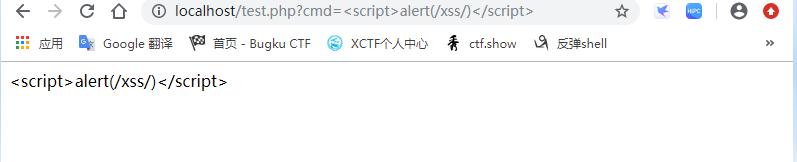
    header("content-type:text/html;charset=utf-8");

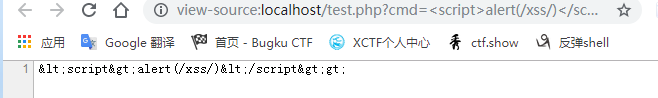
    $x = $\_GET['cmd'];

    $y=htmlspecialchars($x);

    print\_r($y);

?>





**strip\_tags()函数**

**防止xss好办法**

**去除空字符串，html和php标记**

**用法也是：strip\_tags($x);**

<?php

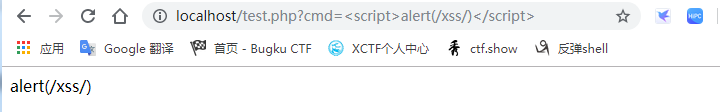
    header("content-type:text/html;charset=utf-8");

    $x = $\_GET['cmd'];

    $y=strip\_tags($x);

    print\_r($y);

?>



**escapeshellcmd()函数**

**对字符串中可能会欺骗shell命令执行任意的字符进行转义，保证用户输入数据是安全的**

**用法：escapeshellcmd($x);**

<?php

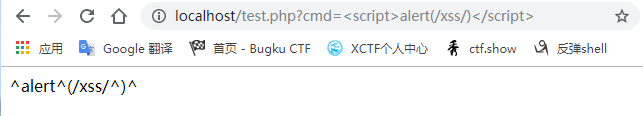
    header("content-type:text/html;charset=utf-8");

    $x = $\_GET['cmd'];

    $y=escapeshellcmd($x);

    print\_r($y);

?>



**intval函数**

函数用于获取变量的整数值。

**使用方法：**

**$id=intval($\_GET['id']);**

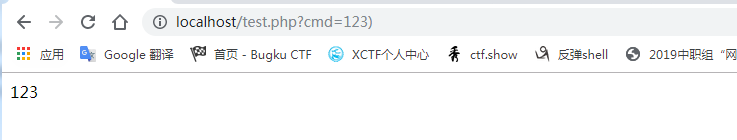
<?php

    header("content-type:text/html;charset=utf-8");

    $x = intval($\_GET['cmd']);

    print\_r($x);

?>



**http-only防御那么js脚本将无法读取cookie信息。**



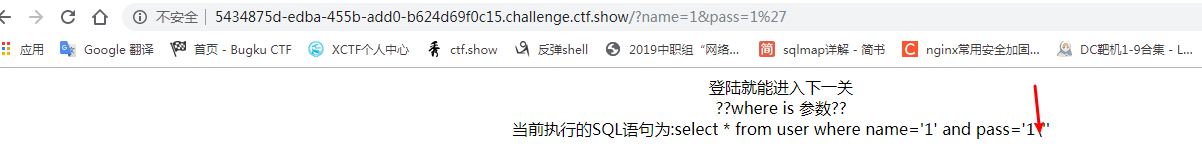
**sprintf函数漏洞**

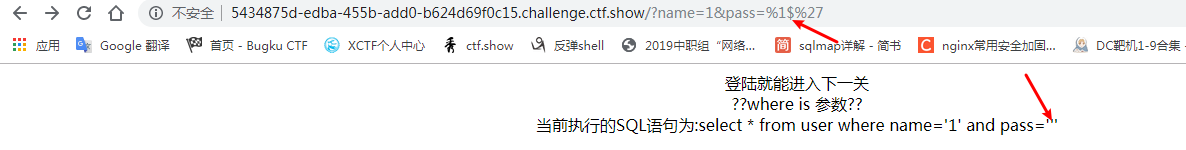
<https://www.ctf.show/challenges#%E7%BB%99%E5%A5%B9-119>**靶场**

sprintf函数使用switch case对15种类型做了匹配，包括%s、%d、%u…但如果在15种类型之外就会直接break。

当我们输入%\或%1$\时，sprintf会把反斜杠当做格式化字符串的类型，但他们并不在15种类型之中，就会未经任何处理而被替换为空

具体漏洞可见 php sprintf 漏洞





**-----------------------------------------------------------------------------------------------------------------------------------------------**

**文件上传笔记**

**上传页面设置.上传方式是POST还是GET，还有enctype类型，和上传到文件的页面,加入上传的文件类型，和名字定义，**然后加入<buttor>提交</buttor>按钮

    <form action="update.php" method="post" enctype="multipart/form-data">

        <input type="file" name="up">

        <button>提交</button>

</form>

**并且设置好php.ini的** upload\_tmp\_dir = "D:/phpstudy\_pro/WWW/update" （windows格式，默认在前面有；要去掉）

**然后创建update.php文件**

**用print\_r(**$\_FILES**)#**通过$\_FILES超级全局变量进行读取

然后文件就回上传到临时文件夹中不过一瞬间就 回消失可以设置sleep(10)这样就可以停留10秒，然后在保存到永久的文件夹中

将临时文件转移到永久目录里要写的函数：

function uploader() #定义函数

{

    if(is\_uploaded\_file($\_FILES['up']['tmp\_name'])) #判断是否合法，如果没成功就false

    {

        $to ='upload/' . $\_FILES['up']['name']; #将保存文件地址为源文件名赋值到$to,可以使用time时间戳添加以防有相同

        if(move\_uploaded\_file($\_FILES['up']['tmp\_name'],$to))#将文件转移到$to的地方去

        {

            return $to; #成功打印地址

        }

    }

    return false; #错误打印false

}

var\_dump(uploader());

**通过前台表单的形式过滤字节大小：**

**字节大小过滤要放在<form></form>里面的最前面**

<input type="hidden" name="MAX\_FILE\_SIZE" value="2000">#字节为单位

    <form action="update.php" method="post" enctype="multipart/form-data">

        <input type="hidden" name="MAX\_FILE\_SIZE" value="2000">

        <input type="file" name="up">

        <button>提交</button>

    </form>

**加固设置文件上传目录无执行权限（简单粗暴）**

**文件上传设置白名单jpg/png/gif**

<?php

function uploader(){

    if(($\_FILES['up']['type'] == 'image/jpeg') || ($\_FILES['up']['type'] == 'image/png') || ($\_FILES['up']['type'] == 'image/gif'))

    {

        $to ='upload/' . $\_FILES['up']['name'];

        if(move\_uploaded\_file($\_FILES['up']['tmp\_name'],$to))

        {

            return $to;

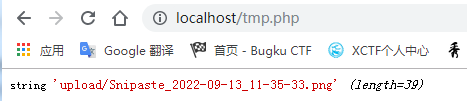
        }

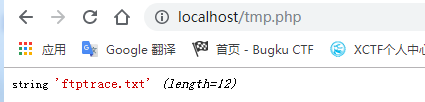
    }

    return $\_FILES['up']['name'];#失败返回上传名字

}

var\_dump(uploader());





文件上传加固

截图.png

设置目录内禁止解析php后缀文件

**文件上传利用的常见函数**

**deldot函数（）**

**去除字符串尾部的点**

**使用方法：echo deldot("hell.world...);**

**输出为:hell.world**

**strrchr**

strrchr(string $haystack, mixed $needle): string

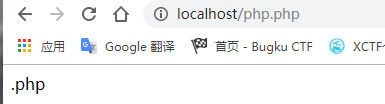
strrchr函数在字符串$haystack中查找$needle，并将最后一次查找到的$needle及其后面的字符串返回。如果没有在该字符串中查找到$needle，则返回false。

注：

如果第二个参数不是不是单个字符，则只使用该字符串的第一个字符进行查找匹配。

如果第二个参数是一个数值，则将该数值转换为对应的ASCII码进行匹配。

<?php$s="1.phP";$k=strtolower($s);echo $e=strrchr($k,".");?>



**5. strtolower**

strtolower(string $string): string

将字符串$string中的**各个英文字符转换为小写并返回**。

当然也可以是数组集体转换

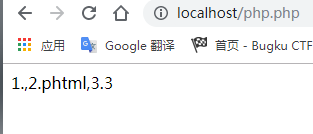
<?php

$s="1.phP,2.pHTml,3.PHp3";

$k=strtolower($s);

echo str\_ireplace("php","",$k)."\n";

?>



**str\_ireplace函数**

**用于对数组中的元素或字符串中的子串进行替换。（不区分大小写，只会过滤一次绕过双写绕过就没办法了）**

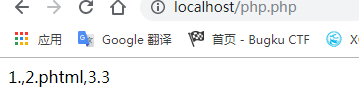
**用法：**echo str\_ireplace("php","","hello.php")."\n";

<?php

$s="1.php,2.phtml,3.php3";

echo str\_ireplace("php","",$s)."\n";

?>



**strstr**

**strchr :也是返回第一次出现的地方和strstr用法一样**

strstr(string $haystack, mixed $needle, bool $before\_needle = false): string

查找字符串$needle在$haystack中首次出现的位置，并将$needle及其之后的字符串返回。

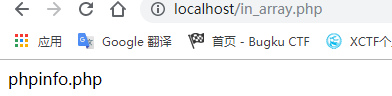
PHP5起新增第三个参数$before\_needle，如果$before\_needle取值为true，则返回$needle前面的部分。

<?php

$s="hahahphpinfo.php";

echo strstr($s,"php");

?>



**substr函数**

**语法**

substr(**string,start,length**)

<?php

echo substr("Hello world",0,10)."<br>";

echo substr("Hello world",1,8)."<br>";

echo substr("Hello world",0,5)."<br>";

echo substr("Hello world",6,6)."<br>";

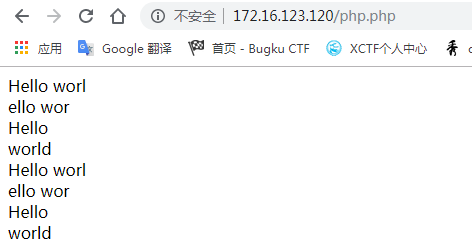
echo substr("Hello world",0,-1)."<br>";

echo substr("Hello world",-10,-2)."<br>";

echo substr("Hello world",0,-6)."<br>";

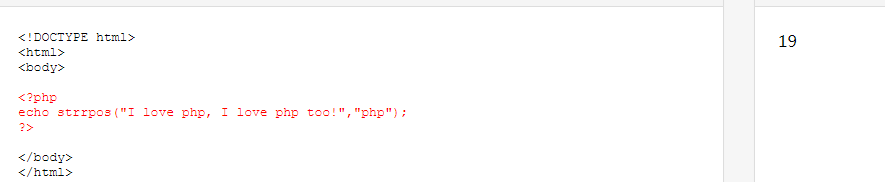
echo substr("Hello world",-2-3)."<br>";

?>



**strrpos()**

函数查找字符串在另一字符串中最后一次出现的位置（区分大小写）



将两个组合在一起就是strrchr函数一样的作用（文件上传第11题就是用这个方法代替想借此误导我们分析）

<?php

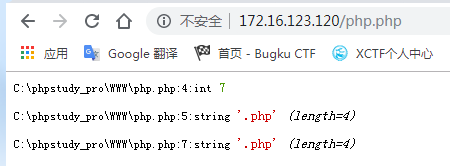
$s="phpinfo.php";

var\_dump(strrpos($s,"."));

var\_dump(substr($s,strrpos($s,".")));

var\_dump(strrchr($s,"."));

?>



in\_array函数

**in\_array() 函数搜索数组中是否存在指定的值。**

in\_array(mixed $needle, array $haystack, bool $strict = false): bool

其中第一个参数$needle为待搜索的值，$haystack为被搜索的数组，第三个参数决定是否进行类型比较。

第三个类型默认为false，即不考虑类型是否相同。

<?php

$sites = array("Google", "Runoob", "Taobao", "Facebook");#创建$sites数值

if (in\_array("Runoob", $sites))#判断$sites数值中是否有Runoob

{

    echo "找到匹配项！";

}

else

{

    echo "没有找到匹配项！";

}

?>

利用这些写的加固代码

//2022.9.14

<?php

function uploader(){

    $suffis = array('.jpg','.gif','.png'); //定义白名单后缀的数组

    $file\_name = trim($\_FILES['up']['name']); //文件名字去空

    $file\_suffis = strrchr($file\_name,'.'); //查找点出现最后出现的位置.也就是后缀名赋值

    $file\_suffis = strtolower($file\_suffis);//防止大写绕过，后缀名全部转为小写

    $file\_suffis = str\_ireplace('::$DATA','',$file\_suffis); //将::$DATA转换为空，防止::$DATA,因为这个还是替换是不管大小写所以小写也可以使用

    $file\_suffis = trim($file\_suffis);//再去空

    if(in\_array($file\_suffis,$suffis)) //判断白名单

    {

        $to ='upload/' .rand(1000,9999).$file\_suffis; //自定义文件名字,及时你成功绕过，但是文件名后缀是我的白名单赋值的名字是时间戳。

        if(move\_uploaded\_file($\_FILES['up']['tmp\_name'],$to))

        {

            return $to;

        }

    }

    return $\_FILES['up']['name'];

}

var\_dump(uploader());

其实7,8,9行都没必要因为只要后缀是.jpg,.png,.gif，就保存，并且保存时会重新添加这三个白名单的后缀。

**最简最强白名单**

<?php

function uploader(){

    $suffis = array('.jpg','.gif','.png');

    $file\_name = trim($\_FILES['up']['name']);

    $file\_suffis = strrchr($file\_name,'.');

    var\_dump($file\_suffis);

    if(in\_array($file\_suffis,$suffis))

    {

        $to ='upload/'.time().rand(100,999).$file\_suffis;

        if(move\_uploaded\_file($\_FILES['up']['tmp\_name'],$to))

        {

            return $to;

        }

    }

    return $\_FILES['up']['name'];

}

var\_dump(uploader());

**2.trim函数仅仅是去掉空格，不能去掉空白字符。因此本关可以通过在sh.php文件名末尾加十六进制81~99（比如下图中加的是0x88）来绕过后缀黑名单限制 对这个码也是无解**

**<?php include('文件名');?>包含查看文件内容**

**弱类型绕过**

