目录

**[MD5绕过 3](#_Toc25859)**

[一、题目描述 3](#_Toc616)

[二、解题工具 3](#_Toc27887)

[三、解题过程 3](#_Toc27047)

[步骤1 3](#_Toc29000)

[步骤2 4](#_Toc2297)

[步骤3 5](#_Toc32345)

[方法一：利用==比较漏洞 5](#_Toc20559)

[方法二：传入数组 6](#_Toc19930)

[2.MD5绕过+SHA1 7](#_Toc10993)

[一、题目描述 7](#_Toc18103)

[二、解题过程 7](#_Toc15186)

[步骤1 7](#_Toc5040)

[步骤2 8](#_Toc32311)

[拓展——sha1碰撞 8](#_Toc24880)

[四． 知识点总结 11](#_Toc6219)

[MD5 绕过的一些思路 11](#_Toc15322)

**[MD5 爆破 15](#_Toc31677)**

[一、题目描述 15](#_Toc27855)

[二、解题工具 15](#_Toc13845)

[三、解题过程 15](#_Toc14527)

[步骤1 16](#_Toc14720)

[步骤2 16](#_Toc28785)

[步骤3 17](#_Toc26030)

四.知识点总结...................................................................................................................................................17

[①已知MD5值与前面部分内容爆破值 18](#_Toc9915)

[②与时间戳相互结合的考法 20](#_Toc24723)

[③MD5加密快速构造提交包的一种做法–>直接构造后面的时间进行循环遍历 20](#_Toc32489)

[五．例题汇总： 21](#_Toc29697)

[例1.——NJUPT2019南邮校赛——easyphp 21](#_Toc25360)

[例2——双MD5碰撞绕过 22](#_Toc19226)

[例3——php中md5($str,true)注入 23](#_Toc5101)

[例4——强网杯2018——Web 签到 24](#_Toc24125)

[例5——强网杯2020——Funhash 25](#_Toc20578)

# MD5绕过

# 一、题目描述

题目链接：https://ctf.bugku.com/challenges/detail/id/83.html

题目来源平台：bugku

题目名称：备份是个好习惯

题目描述：MD5绕过 [http://114.67.246.176:17298](http://114.67.246.176:17298" \t "https://ctf.bugku.com/challenges/detail/id/_blank)

题目类型：WEB

题目附件：

FLAG：flag{38038ab293afa153472d4d4d90402f90}

# 二、解题工具

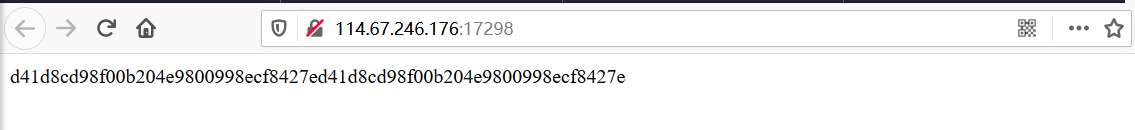
|  |  |  |  |
| --- | --- | --- | --- |
| 工具类型 | 工具名称 | 版本 | 备注 |
| 基础环境 | Python | 3.8 |  |
| Windows system | 10.0.18363 |  |
| 工具 | 御剑后台扫描工具 |  |  |
| Sublime Text 3 |  |  |
| 脚本 |  | \ |  |
|  |  |  |
|  |  |  |

# 三、解题过程

（详细说明CTF赛题的解题过程，分步骤说明）

### 步骤1

根据：



发现是两段重复的字符串

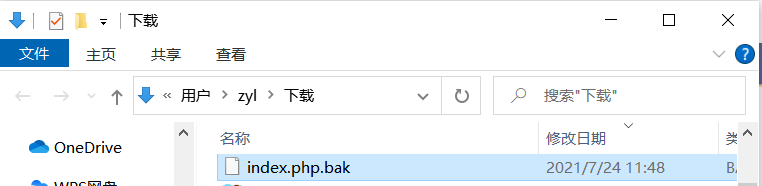
d41d8cd98f00b204e9800998ecf8427e d41d8cd98f00b204e9800998ecf8427e

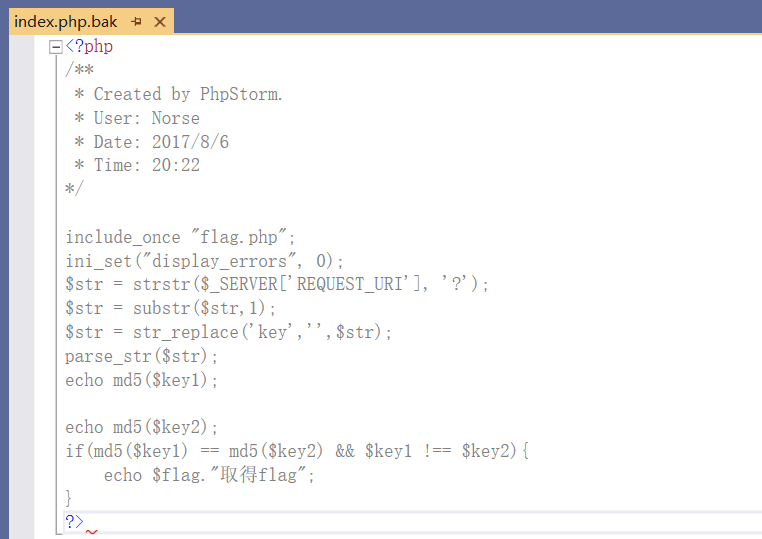
MD5解码得：[md5在线解密破解,md5解密加密 (cmd5.com)](https://www.cmd5.com/)



### 步骤2

御剑扫描后发现bak的备份文件打开得源码





  strstr(string,search,before\_search)：找到search字符串在string中第一次出现的位置，并把后面的字符串输出；before\_search如果为true，则返回位置之前的字符串；默认为false；

  substr(string,start,length)：看参数很容易理解，返回字符串的一部分；

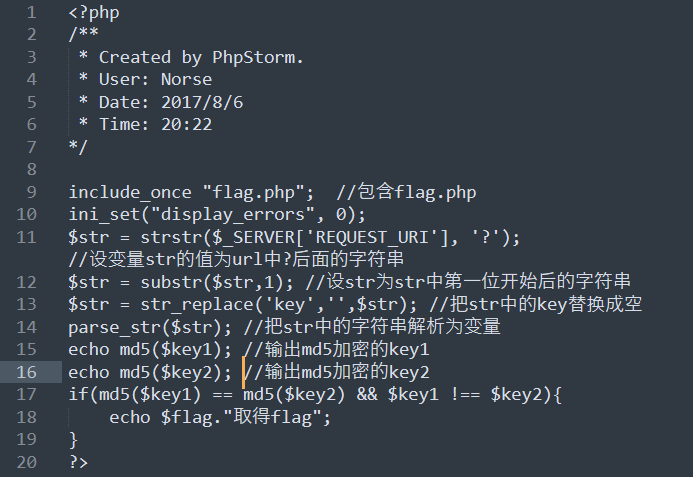
  str\_replace(find,replace,string,count)：在字符串sting中找到字符串find，并把它替换成replace；

  parse\_str(string,array)：把查询字符串解析到变量中；

则代码从?后开始截取到str中，也就是变量后开始截取字符串；然后从位置1开始截到字符串结束；之后把字符串中的kay字段全用‘ ’替代，这个我们可以通过构造"kekeyy"来绕过；最后把key1和key2解析成变量

### 步骤3

用sublim打开



要key1和key2加密后的md5值相等，但是key1和key2不相等，**md5绕过**

方法一：利用==比较漏洞

如果两个字符经MD5加密后的值为 0exxxxx形式，就会被认为是科学计数法，且表示的是0\*10的xxxx次方，还是零，都是相等的。 PHP在处理哈希字符串时，会利用”!=”或”==”来对哈希值进行比较，它把每一个以”0E”开头的哈希值都解释为0，所以如果两个不同的密码经过哈希以后，其哈希值都是以”0E”开头的，那么PHP将会认为他们相同，都是0。

下列的字符串的MD5值都是0e开头的：

QNKCDZO

240610708

s878926199a

s155964671a

s214587387a

s214587387a

将get的两个参数中的key替换为空，（用kekeyy绕过），然后对key1,key2的值进行md5加密，并进行比较，

如果md5加密的值一样而未加密的值不同，就输出flag.

$str = str\_replace('key','',$str);

即传入的参数带有key就会被替换为空，双写kkeyey来绕过，这样k~~key~~ ey即使key替换成空了，一头一尾拼起来还是key

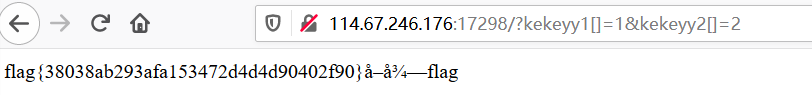
正确的payload:?kekeyy1=QNKCDZO&kekeyy2=240610708

方法二：传入数组

md5()函数无法处理数组，如果传入的为数组，会返回NULL，所以两个数组经过加密后得到的都是NULL,也就是相等的。

正确的payload:?kkeyey1[]=1&kkeyey2[]=2

### 步骤4



# 2.MD5绕过+SHA1

# 一、题目描述

题目链接：https://ctf.bugku.com/challenges/detail/id/101.html

题目来源平台：bugku

题目名称：各种绕过

题目描述：MD5绕过，SHA1 [http://114.67.246.176:15156](http://114.67.246.176:15156" \t "https://ctf.bugku.com/challenges/detail/id/_blank)

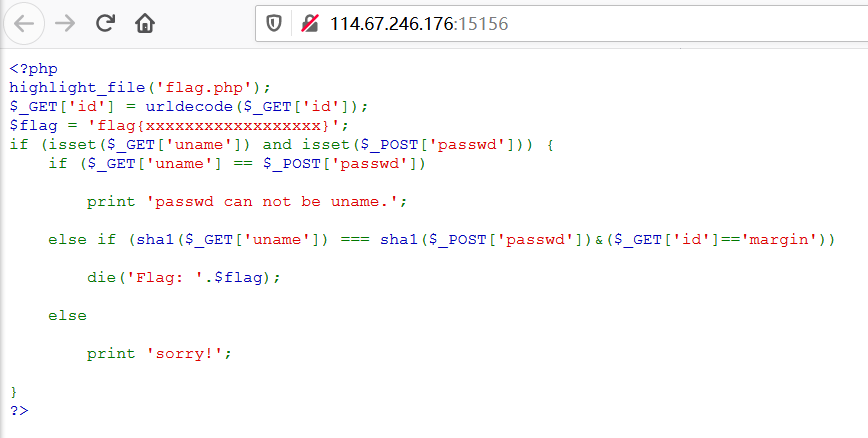
题目类型：WEB

题目附件：（若有请附上附件，若无可忽略）

FLAG：flag{c7a5eb801fe64a5bc4d26d1a98e94934}

# 二、解题过程

### 步骤1



代码中要求uname != passwd且uname的哈希值（sha1）要等于passwd的哈希值时，才会执行die('Flag: '.$flag);

这个和“md5加密的值一样而未加密的值不同”一样可以绕过：

1、md5()函数无法处理数组，如果传入的为数组，会返回NULL，所以两个数组经过加密后得到的都是NULL,也就是相等的。

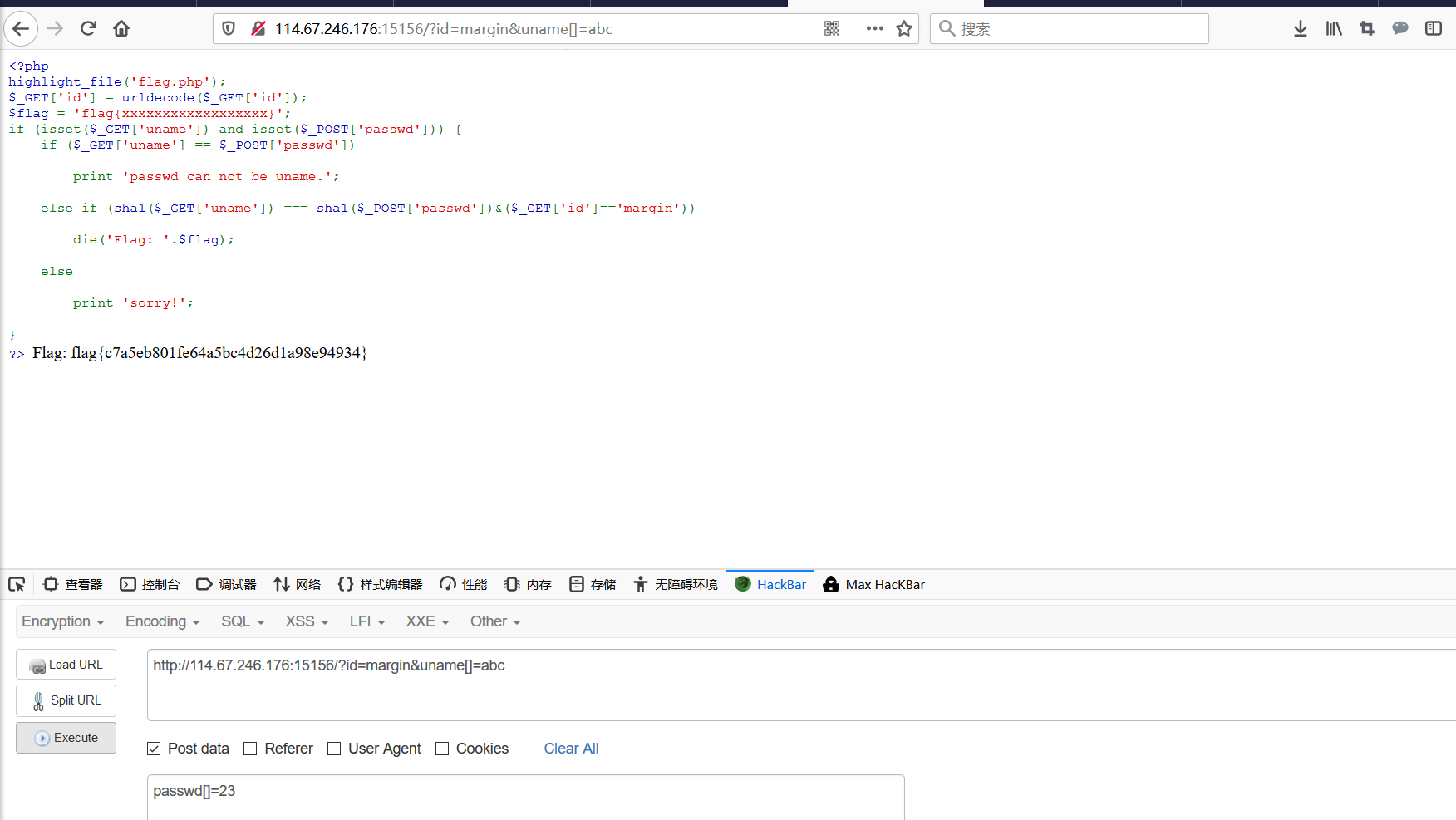
所以，同理sha1()也是可以这样做，

在url构造：id=margin&uname[]=1

### 步骤2

[http://114.67.246.176:15156](http://114.67.246.176:15156" \t "https://ctf.bugku.com/challenges/detail/id/_blank)/?id=margin&uname[]=1

然后用hackbar把passwd的值post进去：passwd[]=2：

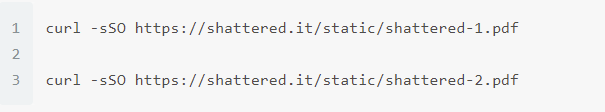


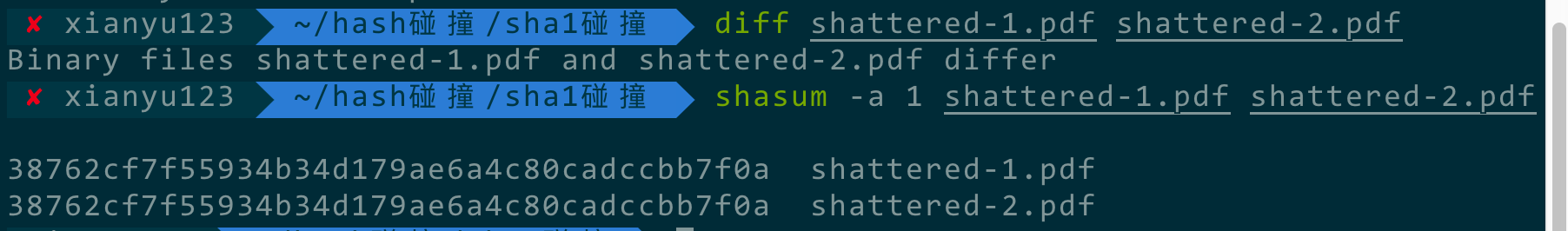
得到FLAG

Flag: flag{c7a5eb801fe64a5bc4d26d1a98e94934}

# 拓展——sha1碰撞

下载文件





### **任意多个文件，文件内容不同，md5值相同**

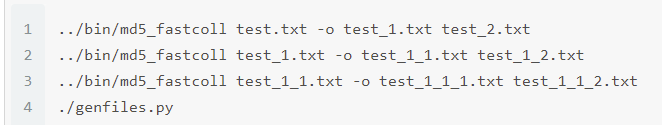
**运用的原理如下：**



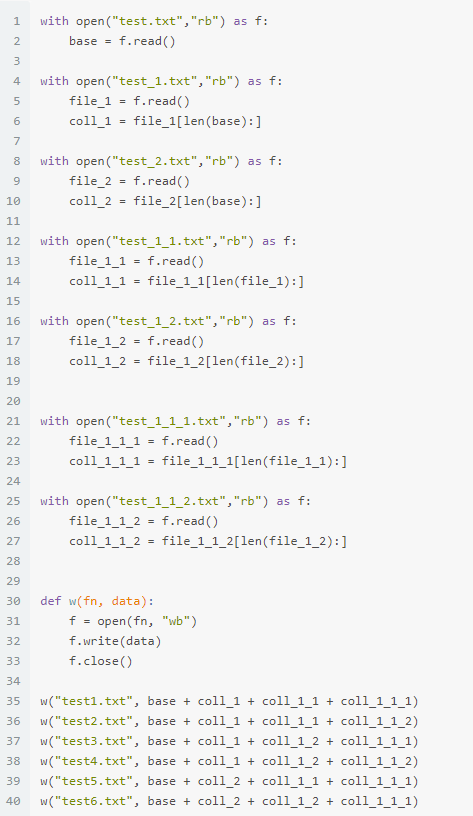
**这里使用的工具还是hashclash**

**我们在在hashclash文件夹下创建工作目录，然后执行下面的shell脚本**

**gen.sh**



**genfiles.py**



**这样就可以生成6个文件内容不同，但是它们的md5都相同，事实上，你可以生成任意多个**

# 知识点总结

## MD5 绕过的一些思路

* 常规的0e绕过
* 数组绕过
* 强类型绕过（碰撞）
* [\$a==md5($a)](https://blog.csdn.net/CSDNiamcoming/article/details/108837347?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_title~default-1.readhide&spm=1001.2101.3001.4242" \l "4_amd5a_62" \t "https://blog.csdn.net/CSDNiamcoming/article/details/_self)
* Md5与SQL注入

### **1. 常规的0e绕过**

* QNKCDZO
* 240610708
* s878926199a
* s155964671a
* s214587387a
* s214587387a

这些字符串的 md5 值都是 0e 开头，在 php 弱类型比较中判断为相等

可以用kekeyy绕过，

$str = str\_replace('key','',$str);

即传入的参数带有key就会被替换为空，双写kkeyey来绕过，这样kkeyey即使key替换成空了，一头一尾拼起来还是key

1. **数组绕过**

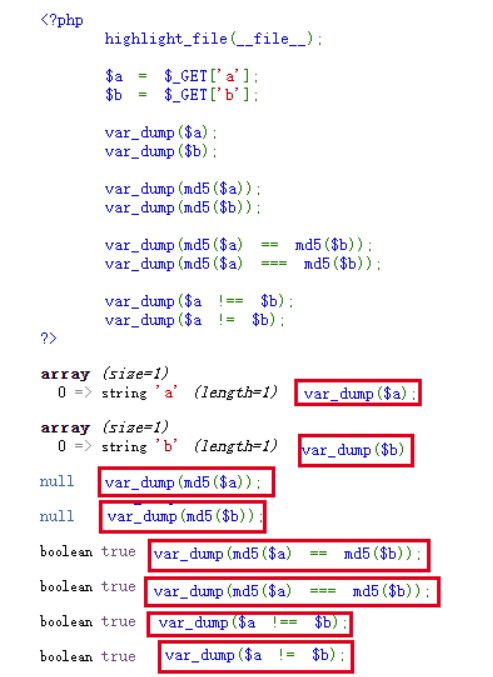
如下代码

var\_dump(md5($\_GET['a']) == md5($\_GET['b']))

传入

a[]=a&b[]=b

虽然会报错，但是判断为真



可以看到，MD5一个数组返回了null，null==null，成功绕过

**3. 强类型绕过**

if((string)$\_POST['a'] !== (string)$\_POST['b'] && md5($\_POST['a']) === md5($\_POST['b']))

[安洵杯 2019] easy\_web（buu有环境）的源码

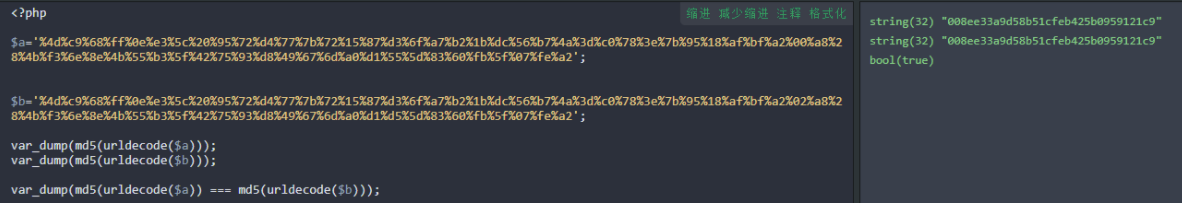
这应该是md5**碰撞**了

payload如下：

a=%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

&b=%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

进行url解码后的MD5值相等



收录一些MD5值相等的字符串



**4. $a==md5($a)**

0e215962017 的 MD5 值也是由 0e 开头，在 PHP 弱类型比较中相等

****exp.py****



**5. md5 与SQL注入**

buu easy\_md5

当 md5 函数的第二个参数为 true 时，返回为 字符串

一个特殊的 md5 值：ffifdyop



# MD5 爆破

# 一、题目描述

题目链接：https://ctf.bugku.com/challenges/detail/id/94.html?page=3

题目来源平台：bugku

题目名称：MD5

题目描述：md5 collision

题目类型：WEB [http://114.67.246.176:19576](http://114.67.246.176:19576" \t "https://ctf.bugku.com/challenges/detail/id/_blank)

题目附件：（若有请附上附件，若无可忽略）

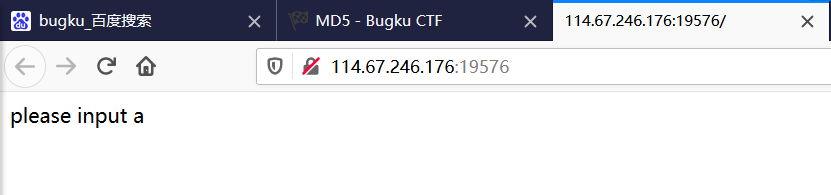
FLAG：

# 二、解题工具

|  |  |  |  |
| --- | --- | --- | --- |
| 工具类型 | 工具名称 | 版本 | 备注 |
| 基础环境 | Python | 3.8 |  |
| Windows system | 10.0.18363 |  |
| 工具 | CTFcrackTools | 2.2beta |  |
| hashclash |  |  |
| 脚本 | CRYPTO.py | \ | Python3.8环境下运行，用于爆破解密 |
|  |  |  |
|  |  |  |

# 三、解题过程

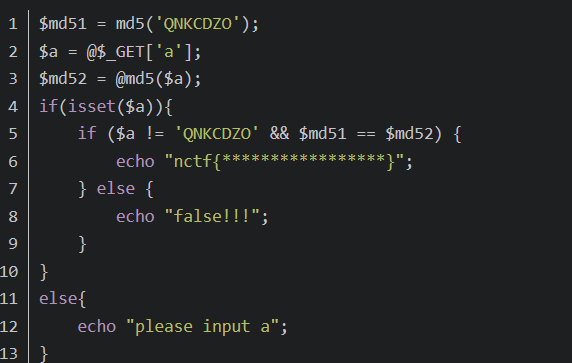
### 步骤1



post一个a=1进去结果提示false，上网寻找源代码

### 步骤2

在网上查找到源代码



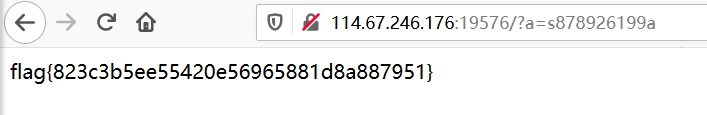
PHP在处理哈希字符串时，会利用”!=”或”==”来对哈希值进行比较，它把每一个以”0E”开头的哈希值都解释为0，所以如果两个不同的密码经过哈希以后，其哈希值都是以”0E”开头的，那么PHP将会认为他们相同，都是0。

攻击者可以利用这一漏洞，通过输入一个经过哈希后以”0E”开头的字符串，即会被PHP解释为0，如果数据库中存在这种哈希值以”0E”开头的密码的话，他就可以以这个用户的身份登录进去，尽管并没有真正的密码。

### 步骤3

### 构造并上传payload：

### [http://114.67.246.176:19576/?a=s878926199a](http://114.67.246.176:14391/?a=s878926199a)



**四．知识点总结**

**1.md5类常用脚本总结**

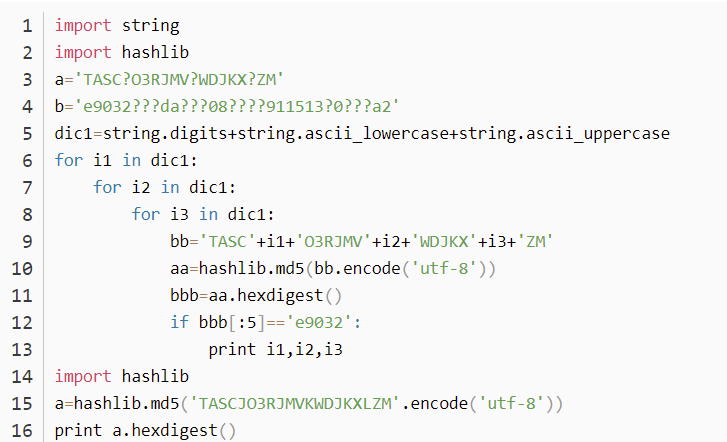
# **①已知MD5值与前面部分内容爆破值**

## **确定长度类**

脚本类型1

# 

类型2



截断验证码:



## **不确定长度类**





# **②与时间戳相互结合的考法**

# 

# **③MD5加密快速构造提交包的一种做法–>直接构造后面的时间进行循环遍历**



# 五．例题汇总：

## **例1.——NJUPT2019南邮校赛——easyphp**

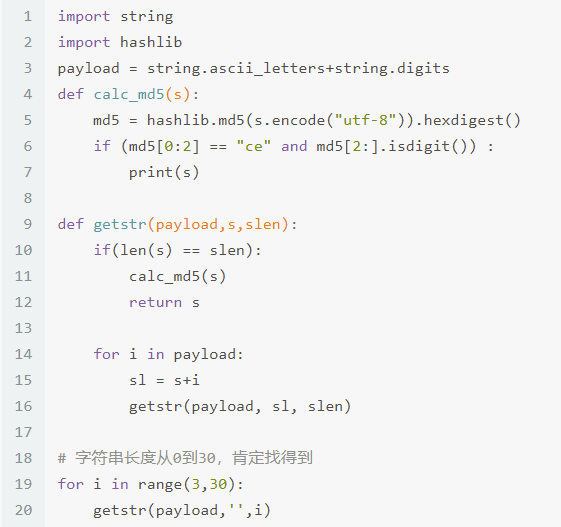
# 

# 这里的关键就是，需要我们找两个个是ce开头的，然后ce后面是纯数字的md5值

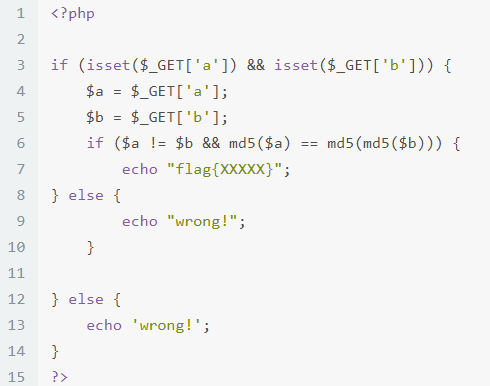
爆破好的例子：



爆破的脚本如下exp.py



## **例2——双MD5碰撞绕过**



这里其实我们只要找出md5(md5($b))是0e开头的且0e后面是纯数字的字符串即可

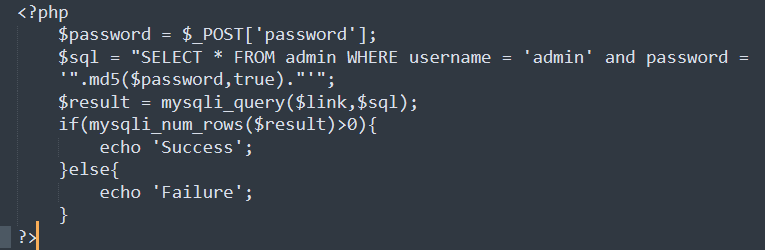
****爆破好的例子：****



****exp.py****



## **例3——php中md5($str,true)注入**



这里md5输出的是16 字符二进制格式，并不是我们平时看到的32字符十六进制数，所以我们只需要找md5加密后字符串中是否存在'or'字符串

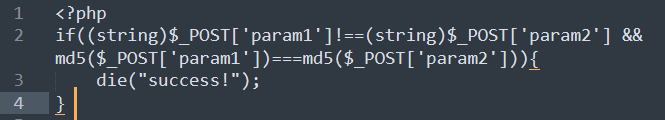
爆破后：



exp.php



## **例4——强网杯2018——Web 签到**

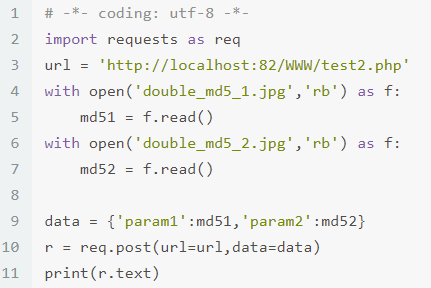


这里因为对两个参数都进行了强制类型转换，所以一般的方法（用数组报错绕过肯定是行不通的了），所以我们必须找到两个文件，他们的内容不一样，但是md5值相等

网上找到两张图片，内容不一样，但是md5值一样

[](http://images.xianyu123.club/20200826171352.jpg)[](http://images.xianyu123.club/20200826171357.jpg)

exp.py

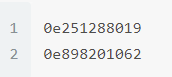


## **例5——强网杯2020——Funhash**



重点关注level 1，脚本：

已经爆破好的例子：

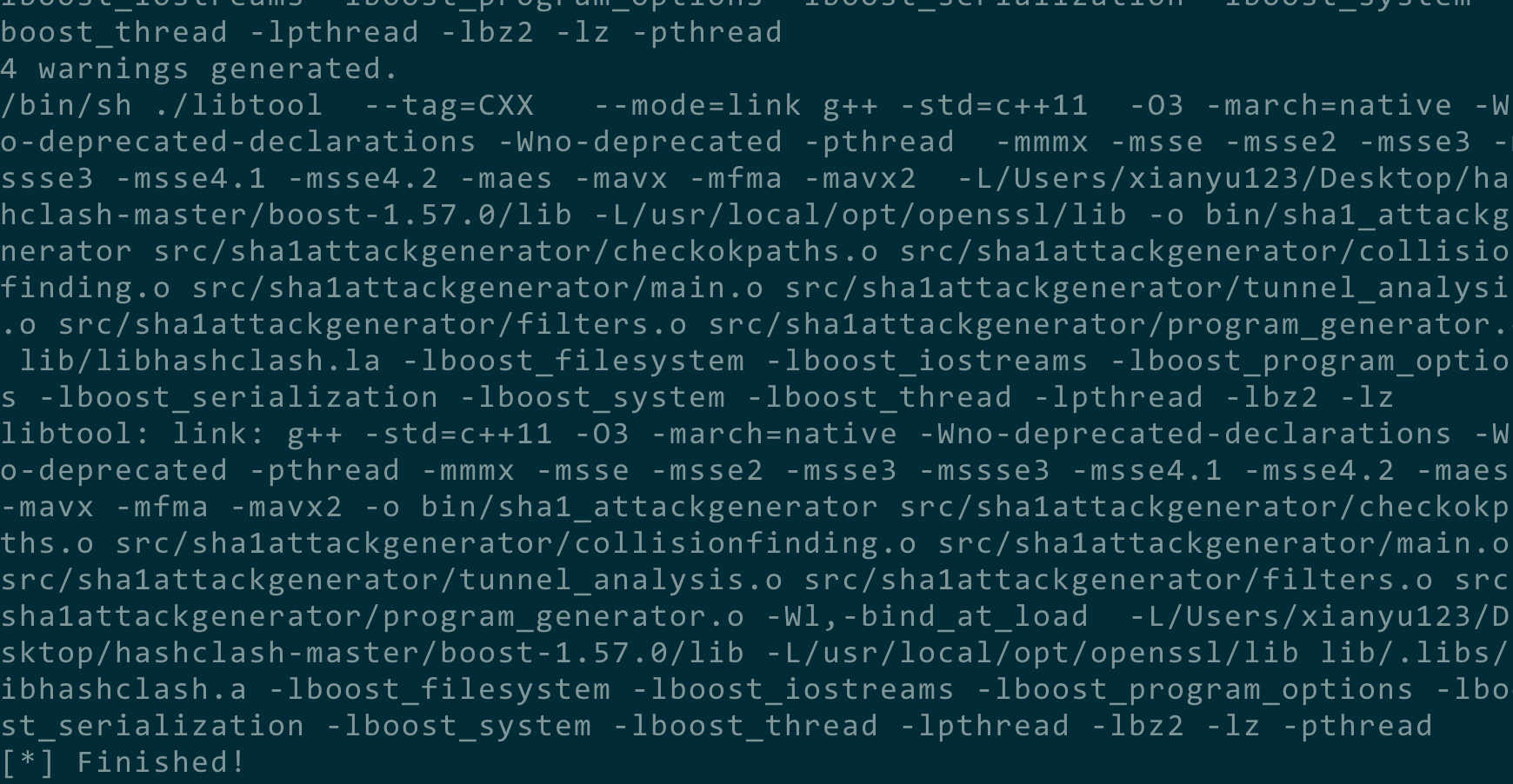


exp.php

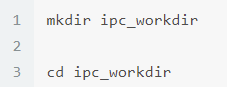


### **两个其他类型的文件md5碰撞**

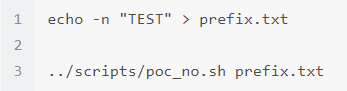
首先下载[hashclash](https://github.com/cr-marcstevens/hashclash" \t "https://0clickjacking0.github.io/2020/08/24/CTF%E4%B8%AD%E5%85%B3%E4%BA%8Emd5%E7%9A%84%E4%B8%80%E4%BA%9B%E6%80%BB%E7%BB%93/_blank)，如果是linux环境的话可以下载编译好的二进制文件，如果是macos系统的话，就需要下载源码，然后自行编译，编译成功如下：



创建目录:



运行脚本：



碰撞：

