

# CTF-web 第一部分 MD5

原创

[iamsongyu](#) 于 2018-10-08 15:31:49 发布 7392 收藏 12

分类专栏: [理论知识](#) [编程实践](#) 文章标签: [CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/iamsongyu/article/details/82968660>

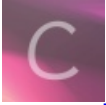
版权



[理论知识](#) 同时被 2 个专栏收录

109 篇文章 7 订阅

订阅专栏



[编程实践](#)

19 篇文章 0 订阅

订阅专栏

## 一. 哈希解密与攻击

哈希就是把任意长度的输入 (又叫做预映射 pre-image) 通过散列算法变换成固定长度的输出, 通常用来进行文件摘要或者信息加密。虽说很难具有相同哈希的文件, 但是某些特意构造的

信息还是会满足相同的哈希, 而且有些时候可以使用字典的方式进行解密和使用哈希攻击。

### 0x00 SHA

```
$_GET['name'] == $_GET['password']  
sha1($_GET['name']) === sha1($_GET['password'])
```

其实最简单的是报错, false, 至于为什么, 其实仔细研究SHA1加密你就发现, 其要求参数不能为数组, 那我将传入的参数改成数组, 两边return的结果不就都为false, 从而, 满足不等与相等了么。

### 0x01 MD5加密

salt型的md5加密可以考虑哈希长度拓展攻击, 详情见我的博客哈希拓展攻击详解

第二种情况, md5加密后的比较, 其实也是利用了"=="弱类型的判断 (详情见我的博客php弱类型总结)

将加密后开头为0e的可以直接判等, 原理这里不做解释了

特殊子串举例如下:

240610708、QNKCDZO、aabg7XsS、aabC9RqS

0x02 urldecode注意点

```
其实也没啥好说的, 就是注意一点$_GET['name'], $_POST['name']均相当于进行了urldecode, 如果存在  
$a = $_GET['name'];  
$b = urldecode($a);
```

\$b相当于被解码了两次, 注意区分

#### (1) MD5碰撞



```
import hashlib
from multiprocessing.dummy import Pool as ThreadPool

# MD5截断数值已知 求原始数据
# 例子 substr(md5(captcha), 0, 6)=60b7ef

def md5(s): # 计算MD5字符串
    return hashlib.md5(str(s).encode('utf-8')).hexdigest()

keymd5 = '8e6d35' # 已知的md5截断值
md5start = 0 # 设置题目已知的截断位置
md5length = 6

def findmd5(sss): # 输入范围 里面会进行md5测试
    key = sss.split(':')
    start = int(key[0]) # 开始位置
    end = int(key[1]) # 结束位置
    result = 0
    for i in range(start, end):
        # print(md5(i)[md5start:md5length])
        if md5(i)[0:6] == keymd5: # 拿到加密字符串
            result = i
            print(result) # 打印
            break

list=[] # 参数列表
for i in range(10): # 多线程的数字列表 开始与结尾
    list.append(str(10000000*i) + ':' + str(10000000*(i+1)))
pool = ThreadPool() # 多线程任务
pool.map(findmd5, list) # 函数 与参数列表
pool.close()
pool.join()
```