MD5和密码保护

*13331231*

*孙圣*

最开始对MD5的了解来源于在网站上下载软件时MD5的信息。当时仅仅知道MD5可以用作校验文件是否被他人篡改。但这几年间，下载时MD5的信息大多被SHA-1所取代，我也就没有过多的关注MD5了。但是上了几节Web安全的课程后又了解到MD5其实在密码保护这一方面有很大的用处。下面我会对MD5的优势和劣势进行分析，这是造成它在前几年间盛行和这几年间逐渐走向下坡路的主要原因。

MD5的前身是MD4，它提高了算法的安全性：增强了运算的复杂度和不可逆性[1]。它具有以下几个优点：

第一，MD5有压缩性，无论多长的字符串，得到的MD5的值都是等长的，即32位16进制，128位2进制数；

第二，计算耗时较少。由于MD5没有涉及特别复杂的算法，因此计算出某个数据的MD5的值很快，在普通的民用机上也基本上能够在毫秒级得出结果；

第三，抗修改性强。如果对原始数据进行修改，即使只修改了1个字符，MD5的值会出现巨大的不同，因此很难通过MD5的值来倒推出原始的数据；

第四，强抗碰撞性，已知原数据和其MD5值，想找到一个具有相同MD5值的数据（即伪造数据）是非常困难的。这两点都大大的保证了MD5算法的安全性，不容易被轻易攻破。[2]

之前一直以为，我们在网路上注册的用户信息（用户名和密码等）是以明码的方式来存储的。其实，这样是很不安全的。首先，网络管理员有权限访问数据库，并看到所存储的密码；第二，黑客可能入侵到网站的数据库，同时进行拖库。如果密码是明文存储的，黑客就可以得到大量用户的信息，之后又可以进行撞库攻击，利用所得到的用户名和密码去尝试进入其它更有利可图的网站系统。毕竟，人具有懒惰性，能通过记一个密码解决的问题，一般不会再去记第二个。所以通过MD5加密在很大程度上都帮助了密码的隐藏和保护。

主要的操作方法如下：

用户在注册完毕后，系统将密码转换成MD5的值，和用户名一起保存在数据库中。当用户下次需要登录时，系统将登录时输入的密码转换成MD5并与数据库中的进行比对，如果相同，则用户密码正确；否则，判断用户密码输入错误。这就解释了当我们在使用登录系统时，按下忘记密码后并不会返回我们所设置的密码，而是系统随机生成的串。

自己实现了一遍MD5之后，发现实现方法确实有点麻烦和复杂，但实际上，MD5并没有我们所认为的那么安全，毕竟随着计算机的运算速度和计算能力的飞速增长，人们已经可以暴力破解MD5，逆向工程出原始的数据。

2004年8月17日的美国[加州](http://baike.baidu.com/view/526914.htm)圣巴巴拉的国际[密码学](http://baike.baidu.com/view/25311.htm)会议（Crypto’2004）上，来自中国山东大学的[王小云](http://baike.baidu.com/view/350813.htm)教授做了破译MD5、HAVAL-128、 MD4和RIPEMD算法的报告，公布了MD系列算法的破解结果。宣告了固若金汤的世界通行密码标准MD5的堡垒轰然倒塌，引发了密码学界的轩然大波。虽然这并非是真正的破解，但是加速了杂凑冲撞。[2]

2010年，卡耐基梅隆大学宣布MD5在密码学意义上已经不在适合用作加密，现在美国政府使用的软件都使用SHA-2进行加密。

同时，有些人提出如果对密码先进行MD5加密之后在进行SHA-1加密应该能提高安全性。但是这种组合方法不能增加用户输入的可能性，另外虽然SHA1生成的是160位的Hash，但是由于输入是一个128位的MD5，所以输出也至多只可能有2^128种可能，猜测的范围也没有缩小。黑客依然可以通过构建彩虹表（存储了MD5值和对应的密码的数据库），反查出密码来。  
 因此，现在比较常用的方法是加盐法。生成一个随机数，我们称之为salt，然后在数据库中记录salt和h=hash(pwd + salt)，查询的时候，得到用户的口令p，然后从数据库中查出salt，计算hash(p+salt)，看是不是等于h，等于就是对的，不等于就是不对的。[3]

总的说来，随着计算机的不断发展，MD5已经不足以应对庞大的计算力所带来的破解了。所以我们还需要更加高级的加密方法来保证数据的安全。

引用：

[1] 维基百科MD5 <https://zh.wikipedia.org/zh-cn/MD5>

[2]百度百科MD5 <http://baike.baidu.com/link?url=aX4NrkYM7-XWBkgd4ZPqs7Tuddae0dVmE7eBy5L8PCC2kCEH0_-yctjUYzZrfu6VSggabwsHzc2wVf4iaNlKFq>

[3]知乎 余天升 <http://www.zhihu.com/question/21668719/answer/18937470>