**密码存储中MD5算法的安全性分析**

***摘要 - 哈希算法通常用于转换***

***密码进入散列理论上是不可能的***

***解密 本文分析了安全隐患***

***哈希算法MD5密码存储和讨论***

***不同的解决方案，如盐和迭代散列。 我们***

***提出了一种在密码存储中使用MD5的新方法***

***使用外部信息，计算盐和随机密钥***

***在MD5计算之前加密密码。 我们***

***建议使用按键拉伸进行哈希计算***

***较慢并使用XOR密码来制作最终的哈希值不可能在任何标准的彩虹表中找到。***

***关键词组件; MD5; 密码存储安全; 数据安全; 数据攻击 rainbow表;***

***随着计算机技术的出现，它变得越来越多***

***有效地将信息存储在数据库中而不是***

***存放在纸质文件中。 Web应用程序，需要用户***

***身份验证通常通过验证输入的密码***

***将它们与通常的真实密码进行比较***

***存储在公司的私人数据库中。如果数据库***

***因此这些密码将被破坏，***

***攻击者可以无限制地访问这些用户***

***个人。现在，数据库使用哈希算法***

***保护存储的密码，但仍然有安全性***

***违规最近在2012年，俄罗斯黑客发布了一个大的***

***来自知名社会的破解密码列表***

***网站包括LinkedIn。这些袭击是***

***发现由于使用弱散号而成功***

***算法。***

***填充后的消息长度（以位为单位）***

***三，***

***MD5算法在密码中的应用***

***存储安全***

***在密码中存储密码是非常不安全的***

***数据库。为了增加密码的安全性，***

***MD5算法可用于哈希原始密码***

***并且哈希值，而不是明文存储在***

***数据库。在认证期间，输入密码也是***

***MD5以类似的方式散列，结果哈希值为***

***与数据库中的哈希值相比较***

***特定用户。***

***二，***

***哈希功能***

***四，***

***MD5安全分析***

***散列函数是单向加密功能***

***采用可变大小的输入明文m，***

***大小哈希输出。在计算上难以破译***

***哈希和任何破解它的尝试几乎是不可行的。一个***

***“安全”哈希函数应该能够抵制前图像***

***攻击和碰撞攻击。由于鸽子的原理***

***和生日悖论，将会有一些投入***

***产生相同的哈希结果。输出长度固定***

***大小128位，共计2128个可能的输出哈希***

***价值观。这个值可能看起来很大，并不是无限的，***

***从而导致碰撞。***

***MD5算法容易出现两种主要的攻击类型：***

***字典攻击和彩虹表。***

***一个。***

***字典攻击***

***在字典攻击中，攻击者尽可能地尝试***

***名为字典的详尽列表中的密码。的***

***攻击者从字典中删除每个密码***

***对受损的散列进行二进制搜索***

***密码这种方法可以通过预处理快得多，***

***计算这些可能密码的哈希值***

***将它们存储在哈希表中。***

***一个。***

***MD5算法***

***B.***

***彩虹桌***

***彩虹表由哈希链组成，更多它们优于哈希表，优化存储***

***要求，虽然查找做得稍慢一些。***

***彩虹表与哈希表不同***

***使用还原和哈希函数创建。减少***

***函数将哈希值转换为明文。明文是***

***不是哈希值所在的原始明文***

***生成，但另一个。通过交替散列函数***

***具有缩减功能，交替密码链***

***并形成哈希值。只有第一（连锁起点）***

***生成的最后一个明文（链的终点）存储在***

***桌子。要解密散列密码，我们先进行处理***

***通过减少功能的哈希密码直到我们***

***找到一个匹配链的终点。然后我们拿这条链***

***相应的起点和重新生成哈希链和***

***找到散列密码的原始明文。彩虹***

***桌子现在很容易上网。有许多***

***密码破解系统和使用彩虹的网站***

***表也​​是例如OphCrack。当然，使用***

***彩虹桌不能保证100％的成功率***

***不能用来破译盐渍的哈希。但是，两个***

***相同的密码仍然会产生相同的散列。***

***2）***

***不同的随机盐为每个密码和存储***

***数据库内的盐：如果我们使用不同的盐***

***对于每个密码，两个相同的密码将有所不同***

***哈希攻击者必须生成不同的彩虹表***

***对于每个用户来说，使计算更难***

***为攻击者破解散列。这些盐可以储存***

***在数据库中的明文。盐的目的不是***

***是秘密的，但要随意足以打败使用***

***彩虹桌。***

***3）***

***使用现有列值：现有列***

***像用户名一样的值可以用作盐。这个解决方案是***

***类似于上面讨论的第二个解决方案。它击败了***

***使用标准彩虹表，但黑客可能会生成***

***一个彩虹表的特定用户名，例如“root”***

***或“管理员”。***

***4）***

***使用可变的计算盐：盐***

***开裂***

***密码系统但是，越大越好***

***位置可以是前缀（盐附加在密码前面），***

***中缀（盐附加在密码中）或后缀（盐***

***附加在密码的末尾）。如果盐的位置是***

***做随机，然后破解密码变得更加困难。***

***例如，我们可以将盐位置设置为等于***

***密码的长度模数3.盐可以通过计算***

***使用随机字符序列（存储在数据库中）***

***并使用其他字符（嵌入代码中）。对于***

***例如，盐可以作为第一个的组合***

***两个字母的用户名，随机的字符序列***

***最后2个字母的用户名。***

***用于创建彩虹表的字符集和***

***哈希链长度越长，彩虹越大***

***表是。***

***五***

***对策研究***

***一个。***

***信息熵***

***密码强度通常是根据***

***信息熵。简单来说，越高***

***信息熵越大，密码越大***

***更难的是破解它。 6个字符的密码***

***只需要26次尝试耗尽所有可能性***

***一个暴力攻击，一个密码为42个字符***

***将需要242次尝试。可以看出，越长***

***密码和它所在的字符集越大***

***派生，密码越强。作为最佳做法和***

***初步要求，应用要坚持***

***用户在注册期间使用强密码***

***处理。强密码运行的风险较低***

***字典常见的简单密码如“123456”有***

***已被Microsoft Hotmail禁止。***

***5）使用可变的计算盐，包括***

***数据库外的信息和应用程序代码：***

***黑客现在必须打破数据库和服务器***

***包含应用程序代码。他也需要获得***

***破解密码所需的其他信息。***

***C。***

***改进MD5处理***

***可以使用以下方法来改善MD5***

***加工：***

***1）***

***功能被更改，例如使用以下之一***

***功能如（1），（2）和（3）所示：***

***改进的散列函数：哈希计算***

***B.***

***盐***

***成功密码的最常见原因之一***

***像LinkedIn这样针对LinkedIn的破解攻击是因为***

***他们使用无盐哈希。这使得它更容易***

***黑客通过使用彩虹表破解系统，***

***特别是事实许多用户使用非常常见的简单密码，并且这些类似的密码导致不相似的哈希。盐是一个附加到该文本然后散列的一串字符的信息的次要部分。盐渍使密码对彩虹表更具抵抗力，因为盐化散列密码将具有更高的信息熵，因此在预先计算的彩虹表中更不可能存在。通常情况下，一个至少应该是48位。可以使用以下方式实现盐析：hash = Hash（password + salt）（1）（2）（3）hash = Hash（Hash（password）+ salt）hash = Hash（password + salt + key）2）迭代散列：密码多次散列。 MD5是一个快速散列函数，也就是计算速度快。迭代散列使得计算速度较慢，因此计算速度较慢，而且更多的是1）所有密码的单一盐：鉴于盐的使用时间长而复杂，标准的彩虹表，难以破解迭代次数通常可以是***

***等于1000。***

***3）***

***钥匙拉伸：这使得密码更具抵抗力***

***通过使攻击工作量进行预计算攻击***

***大。使用迭代散列，在那里提供弱键***

***进入哈希算法，输出结果更强***

***键。有3个按键拉伸方法取决于***

***用于迭代散列的输入：***

***a）简单按键拉伸：只有键被散列***

***反复地，如（4）。不涉及盐***

***图2.通过柱状转置生成复杂的密码***

***第四，一个128的附加随机信息字符串***

***为每个用户生成位并存储在外部文件中，***

***例如在闪存驱动器。***

***key =哈希（键）***

***（4）***

***b）密码键拉伸：密码随附***

***关键都在循环中使用。***

***c）盐密钥拉伸：密钥，密码和盐***

***在循环中使用。这种方法是三大关键之一***

***拉伸方法。***

***在哈希前转换密码：之前***

***计算密码的MD5哈希值，后者是***

***使用简单的密码方法进行转换。***

***最后，使用基于公式的哈希密码***

***钥匙拉伸。散列过程类似于密码***

***块链接方法，其中一轮的输出为***

***用于另一轮的输入，如图1所示。来自***

***一次计算哈希值的异或结果***

***与前一轮的一个，由此产生的散列值***

***在任何标准的彩虹表中都不可能找到。***

***然后将最终散列密码存储在数据库中。***

***系统通过计算哈希值对用户进行认证***

***（随机密钥从数据库检索以供使用），***

***然后将其与存储的散列密码进行比较。一个***

***哈希密码如何出现的例子***

***数据库如图1所示。 4。***

***4）***

***5）***

***链接方法和异或（异或）密码：If***

***使用迭代散列，从当前的哈希输出***

***迭代用于下一次迭代的输入。我们用一个***

***简单的XOR密码通过“XORing”计算最终的散列***

***来自所有迭代的哈希输出值。一个简单的异或***

***密码通常是（5）所示的形式。如果键是***

***finalHash = HV0 ^ HV1 ^ ...... ^ HVN，***

***做得足够随便***

***不可能破解***

***的***

***密文***

***将***

***几乎***

***HV0 =哈希（CpxPassword，additionalInfo）;***

***HV1 =哈希（CpxPassword，HV0，盐）;***

***HVN = Hash（CpxPassword，HVN-1，盐）;***

***N是迭代次数，^是XOR。***

***HV：哈希值和***

***明文XOR key =密文***

***（5）***

***D.***

***改进的MD5处理的例子***

***我们现在将演示如何在其中填入密码***

***数据库使用MD5的改进版本。有五个***

***主要步骤涉及***

***CpxPassword：复杂密码***

***图3.使用改进的数据库中的散列密码示例***

***MD5***

***首先，随机长度的随机密钥串是第一个***

***生成。其字符集为{0-9，a-z，A-Z}***

***密钥字符串用于生成复杂的密码***

***也存储在数据库中供以后在密码期间使用***

***认证。***

***其次，密码被转换成一个复杂的***

***密码通过柱状转置密码。假设***

***随机密钥是“YDCiA”，密码是***

***“crazyrichard”，密码首先转换为矩阵***

***5列（与随机键的长度相同）和空白***

***电池交替填充“\*”和“@”，如图所示***

***图2使用柱状转置密码，复合体***

***生成的密码是“ya \* ac \* ridcrrzh @”。***

***第三，通过找到XOR值计算盐***

***的随机密钥字符串与复合密码，行***

***行。在我们的例子中，盐是“％i \ b \”s6 \* \ r \“”。***

***图4.使用改进的数据库中的散列密码示例***

***MD5***

***总体算法可以总结在图1中。 5.***

***这里使用的初始化向量是附加信息，***

***由亚特兰蒂斯出版社出版，巴黎，法国。***

***©作者，2013***

***0708***

***第二届计算机，通信，控制与自动化国际研讨会论文集（ISCCCA-13）***

***它是128位的随机串。每个用户都有一个***

***不同的初始化向量值。***

***初始化向量和盐对于每个用户是不同的。对于***

***两个用户具有相同的密码，最终的哈希值会***

***完全不一样***

***七，***

***结论***

***密码存储安全是数据的一个重要方面***

***安全性，因为大多数系统现在需要验证***

***方法使用密码。哈希算法如MD5***

***通常用于加密明文密码***

***理论上不能被黑客破坏的字符串***

***由于它们的单向加密功能。但随着时间的流逝，***

***攻击通过使用字典表和彩虹表成为可能。在本文中，我们讨论了阻止这些攻击的不同方法：（1）使用强密码来减少在词典中存在的概率，（2）使用盐，（3）密钥拉伸和迭代哈希，使MD5计算更慢， （4）链接方法，其中一次迭代的输出在下一次迭代中被使用，并且对于每个密码使用不同的初始化向量，（5）加密之前的密码和（6）XOR密码以使得最终哈希值不可能找到任何彩虹表。所提出的方法的实现使用C＃编程语言和Microsoft SQL Server as数据库进行。通过我们提出的方法，攻击者现在将要进入数据库，服务器包含应用程序代码以及外部文件。图5改进的MD5处理（IV：初始化向量）VI.PERFORMANCE ANALYSIS首先，使用标准彩虹表将失败，因为存储在数据库中的散列密码不是十六进制形式，因此不会在任何标准彩虹表中出现。我们尝试了一些onlineMD5解密工具，如http://www.md5decrypter.co.uk/和下载的软件工具，如Cain和Abel。但是，由于所有这些都使用彩色表格，MD5以十六进制形式显示，我们存储的哈希值为ASCII码，所以从一开始就将失败，如图所示。另外，通过对来自每个文件的输出哈希值进行异或，几乎不可能在第一轮找到原始的输出。一般来说，如果我们使用一个密钥来XER平面文本来计算密文，那么我们可以通过使用密钥对密文进行异或来简化明文。然而，如果密钥是随机的，那么几乎不可能得到明文。在我们的例子中，key是完全随机的，因为我们在哈希输出值之间进行散列，从而给出了最终的哈希值，这是不可能解密的。本文所报道的工作得到了北京市自然科学基金资助（Grant No.4112037）的支持。作者要感谢并承认亲戚朋友和同事的支持和帮助.REFERENCES [1] [2] Rivest，R. MD5消息摘要算法。 RFC 1321,37（1992年4月）。张少兰，邢国波，杨一贤，MD5改进与安全分析[J]。计算机应用，2009，vol。 29（4）：947-949。 [3] [4]郑晓玲，吉东金，密码认证中MD5算法的应用与安全研究，第9届国际模糊系统与知识发现会议，H. Mirvaziri，Kasmiran Jumari，Mahamod Ismail，Z.Mohd Hanapi，基于现有摘要算法组合的新哈希函数，第五届学术研究与发展大会 - 2007年12月11 - 12日，马来西亚SCOReD。 [5] Md。 Didarul Alam Chawdhury和A.H.M. Ashfak Habib，通过使用TCP头的未使用的比特来安全地增强MD5密码密码，第11届国际计算机与信息技术会议论文集（ICCIT 2008）2008年12月25 - 27日，孟加拉国库尔纳图6. MD5中无效的字符Cain和Abel中的哈希另外，即使两个用户有两个相同的密码，用于加密密码的随机密钥将会不同，导致不同的复杂密码。而且，***