第4章 口令破解与防御技术

国家计算机网络入侵防范中心

张玉清

内容安排

- **4.1** 口令的历史与现状
- **4.2** 口令破解方式
- **4.3** 典型的口令破解工具
- **4.4** 口令攻击的综合应用
- **4.5** 口令攻击的防御
- **4.6** 小结

4.1 口令的历史与现状

- **20**世纪**80**年代，当计算机开始在公司里广泛应用时，人们很快就意识到需要保护计算机中的信息。
- 如果仅仅使用一个**userID**来标识自己，由于别人很容易得到这个**userID**，几乎无法阻止某些人冒名登录。基于这一考虑，用户登录时不仅要提供**userID**来标识自己是谁，还要提供只有自己才知道的口令来向系统证明自己的身份。

4.1 口令的历史与现状

- 口令的作用就是向系统提供唯一标识个体身份的机制，只给个体所需信息的访问权，从而达到保护敏感信息和个人隐私的作用。
- 虽然口令的出现使登陆系统时的安全性大大提高，但是这又产生了一个很大的问题。
- 如果口令过于简单，容易被人猜解出来；如果过于复杂，用户往往需要把它写下来以防忘记，这种做法也会增加口令的不安全性。
- 当前，计算机用户的口令现状是令人担忧的。

4.1 口令的历史与现状

- 另外一个和口令有关的问题是多数系统和软件有默认口令和内建帐号，而且很少有人去改动它们，主要是因为：
  - 不知道有默认口令和帐号的存在，并不能禁用他们；
  - 出于防止故障以防万一的观点，希望在产生重大问题时，商家能访问系统，因此不想改口令而将商家拒之门外；
  - 多数管理员想保证他们自己不被锁在系统之外：一种方法就是创建一个口令非常容易记忆的帐号；另一种方法就是和别人共享口令或者把它写下来。而以上两种都会给系统带来重大安全漏洞。

4.2 口令破解方式

- 4.2.1 口令破解方式概述
- 4.2.2 词典攻击
- 4.2.3 强行攻击
- 4.2.4 组合攻击
- 4.2.5 常见攻击方式的比较
- 4.2.6 其它的攻击方式

4.2.1 口令破解方式概述

- 口令破解是入侵一个系统比较常用的方法。
- 获得口令的思路：
  - 穷举尝试：最容易想到的方法
  - 设法找到存放口令的文件并破解
  - 通过其它途径如网络嗅探、键盘记录器等获取口令
- 这里所讲的口令破解通常是指通过前两种方式获取口令。这一般又有两种方式：手工破解和自动破解。

4.2.1 口令破解方式概述（2）

- 手工破解的步骤一般为：
  - 产生可能的口令列表
  - 按口令的可能性从高到低排序
  - 依次手动输入每个口令
  - 如果系统允许访问，则成功
  - 如果没有成功，则重试。
  - 注意不要超过口令的限制次数
- 这种方式需要攻击者知道用户的**userID**，并能进入被攻击系统的登陆界面。需要先拟出所有可能的口令列表，并手动输入尝试。
- 思路简单，但是费时间，效率低

4.2.1 口令破解方式概述（3）

□ 自动破解

- 只要得到了加密口令的副本，就可以离线破解。这种破解的方法是需要花一番功夫的，因为要得到加密口令的副本就必须得到系统访问权。
- 但是一旦得到口令文件，口令的破解就会非常的快，而且由于是在脱机的情况下完成的，不易被察觉出来。

4.2.1 口令破解方式概述（4）

- 自动破解的一般过程如下：
  - 找到可用的userID
  - 找到所用的加密算法
  - 获取加密口令
  - 创建可能的口令名单
  - 对每个单词加密
  - 对所有的userID观察是否匹配
  - 重复以上过程，直到找出所有口令为止

4.2.2 词典攻击

- 所谓的词典，实际上是一个单词列表文件。这些单词有的纯粹来自于普通词典中的英文单词，有的则是根据用户的各种信息建立起来的，如用户名字、生日、街道名字、喜欢的动物等。
- 简而言之，词典是根据人们设置自己账号口令的习惯总结出来的常用口令列表文件

4.2.2 词典攻击(2)

- 使用一个或多个词典文件，利用里面的单词列表进行口令猜测的过程，就是词典攻击。
- 多数用户都会根据自己的喜好或自己所熟知的事物来设置口令，因此，口令在词典文件中的可能性很大。而且词典条目相对较少，在破解速度上也远快于穷举法口令攻击。
- 在大多数系统中，和穷举尝试所有的组合相比，词典攻击能在很短的时间内完成。

4.2.2 词典攻击(3)

- 用词典攻击检查系统安全性的好处是能针对特定的用户或者公司制定。
- 如果有一个词很多人都用来作为口令，那么就可以把它添加到词典中。
- 在**Internet**上，有许多已经编好的词典可以用，包括外文词典和针对特定类型公司的词典。
- 例如，在一家公司里有很多体育迷，那么就可以在核心词典中添加一部关于体育名词的词典。

4.2.2 词典攻击(4)

- 经过仔细的研究了解周围的环境，成功破解口令的可能性就会大大的增加。
- 从安全的角度来讲，要求用户不要从周围环境中派生口令是很重要的。

4.2.3 强行攻击

- 很多人认为，如果使用足够长的口令或者使用足够完善的加密模式，就能有一个攻不破的口令。
- 事实上，是没有攻不破的口令的，攻破只是一个时间的问题，哪怕是花上**100**年才能破解一个高级加密方式，但是起码他是可以破解的，而且破解的时间会随着计算机处理速度的提高而减少。**10**年前需要花**100**年才能破解的口令可能现在只要花一星期就可以了。

4.2.3 强行攻击(2)

- 如果有速度足够快的计算机能尝试字母、数字、特殊字符所有的组合，将最终能破解所有的口令。这种攻击方式叫做强行攻击（也叫做暴力破解）。
- 使用强行攻击，先从字母**a**开始，尝试**aa**、**ab**、**ac**等等，然后尝试**aaa**、**aab**、**aac** ……。

4.2.3 强行攻击(3)

- 系统的一些限定条件将有助于强行攻击破解口令。
- 比如攻击者知道系统规定口令长度在6~32位，那么强行攻击就可以从6位字符串开始破解，并不再尝试大于32位的字符串。

4.2.3 强行攻击(4)

- 使用强行攻击，基本上是**CPU**的速度和破解口令的时间上的矛盾。
- 现在的台式机性能增长迅速，口令的破解会随着内存价格的下降和处理器速度的上升而变得越来越容易了。

4.2.3 强行攻击(5)

- 一种新型的强行攻击叫做分布式暴力破解，如果攻击者希望在尽量短的时间内破解口令，他不必购买大批昂贵的计算机，而是把一个大的破解任务分解成许多小任务，然后利用互联网上的计算机资源来完成这些小任务，加快口令破解的进程。

4.2.4 组合攻击

- 词典攻击虽然速度快，但是只能发现词典单词口令；强行攻击能发现所有口令，但是破解的时间长。
- 很多情况下，管理员会要求用户的口令是字母和数字的组合，而这个时候，许多用户就仅仅会在他们的口令后面添加几个数字，例如，把口令从**ericgolf**改成**ericgolf2324**，这样的口令利用组合攻击很有效。
- 组合攻击是在使用词典单词的基础上在单词的后面串接几个字母和数字进行攻击的攻击方式。

4.2.4 组合攻击(2)

- 组合攻击是使用词典中的单词，但是对单词进行了重组，它介于词典攻击和强行攻击之间。

4.2.5 常见攻击方式的比较

	词典攻击	强行攻击	组合攻击
攻击速度	快	慢	中等
破解口令数量	找到所有词典单词	找到所有口令	找到以词典为基础的口令

4.2.6 其它的攻击方式

- 口令安全最容易想到的一个威胁就是口令破解，许多公司因此花费大量功夫加强口令的安全性、牢固性、不可破解性，但即使是看似坚不可摧很难破解的口令，还是有一些其它手段可以获取的，类似大开着的“后门”。
  - 社会工程学
  - 偷窥
  - 搜索垃圾箱
  - 口令蠕虫
  - 特洛伊木马
  - 网络监听
  - 重放

社会工程学

- 社会工程学：是一种让人们顺从你的意愿、满足你的欲望的一门艺术与学问，并不直接运用技术手段，而是一种利用人性的弱点、结合心理学知识，通过对人性的理解和人的心理的了解来获得目标系统敏感信息的技术。简单来说，就是欺骗人们去获得本来无法访问的信息。
- 在多数的公司里，如果得到信任，就会被允许拥有访问这个公司信息的特权，如雇员、合同方。
- 如果攻击者能通过一些方式得到系统可用的访问账号和口令，或使公司内部的某个人确信他是被信任的实体，他就很可能获得系统的访问权限。

社会工程学(2)

- 攻击者：喂，我是大为。我在技术支持中心工作，现在我要对你的系统进行例行维护。
- 受骗者：是吗？我从来没有听说支持中心要对我们的系统进行例行维护。
- 攻击者：嗯，是这样，上个季度我们才开始做这个工作。我们正在为所有远程用户做例行维护。我刚刚做完北区的所有计算机的维护。实际上，绝大多数用户都说，经过这次维护之后，他们的计算机的速度明显加快了。
- 受骗者：是吗？那好，如果其它人的机器速度都加快了，那么我也这样做一下。现在我需要做些什么？
- 攻击者：嗯，你不需要做什么。我可以远程地把一切都为你做好，但为了能够这样做，我需要知道你的**VPN**用户名和口令。
- 受骗者：你真的能够远程地做好这一切？真是太好了。嗯，我的用户名是**abc**，口令是**chaodong**。
- 攻击者：太好了。谢谢你的协助。我将以**VPN**[illegible]算机，并进行例行维护。这只需要几分钟的时间

至此，我们得到了用户名和密码

偷窥

- 得到口令的另一个简单而又可行的方法就是观察别人敲口令，这种办法就叫偷窥。在开放的三维空间，这一点并不难。
- 有人曾经做过一个“偷窥”的实验：在一个纽约的冬天，他在一个公司后门的入口处停下汽车，当他从汽车里出来的时候身穿长大衣，提着一个似乎很重的箱子，跟在一个正要进入大楼的人的身后。询问那个人是否可以帮他把大门打开，而那个人也照做了，而且并没有询问他是否有徽章，第一步成功，成功地进入了建筑楼内；然后他找到了管理员的工作间，因为他想得到管理员的权限，通过桌子上的一份文件，他得到了管理员的名字，当管理员进来以后，他谎称他们公司正在做一个测试，给管理员电子邮件，想知道他是否有收到。当管理员登陆系统的过程中，他就站在管理员的身后，并成功地偷窥到了管理员的口令，也就偷窥到了系统的管理员权限。

搜索垃圾箱

- 有许多的人在丢弃垃圾的时候甚至不把电子邮件、文档、计划和口令撕成两半就丢弃了，更别说粉碎后再丢弃。
- 而且许多公司的垃圾都是丢到一个垃圾箱里，大多数清洁工都是在晚上打扫办公室，如果凌晨2点到一些垃圾箱去找找，会很容易就找出一些相当有用的资料。

口令蠕虫

- **2003年，“口令蠕虫”突袭我国互联网，它通过一个名为dvldr32.exe的可执行程序，实施发包进行网络感染操作。数以万计的国内服务器被感染并自动与境外服务器进行连接。**
- **该“口令蠕虫”的特点如下：**
  - 自带一份口令字典，对网上主机超级用户口令进行基于字典的猜测。
  - 一旦猜测口令成功，该蠕虫植入7个与远程控制和传染相关的程序，立即主动向国外的几个特定服务器联系，并可被远程控制。
  - 可以实现大规模的自动化口令攻击，扫描流量极大，容易造成网络严重拥塞。

口令蠕虫（2）

- 口令攻击是常见的黑客攻击方式，但像“口令蠕虫”这样形成大面积、大规模自动化的网上口令攻击，并致使被攻击系统与国外特定服务器进行连接，是一种新的网络攻击方式。
- 与以往利用操作系统或应用系统的技术漏洞进行攻击不同的是，“口令蠕虫”所利用的是网上用户对口令等管理的弱点进行攻击。

特洛伊木马

- 特洛伊木马程序可以直接侵入用户的电脑并进行破坏，它常被伪装成工具程序或者游戏等诱使用户打开带有特洛伊木马程序的邮件附件或从网上直接下载，一旦用户打开了这些邮件的附件或者执行了这些程序之后，就在计算机系统中隐藏一个可以在**OS**启动时悄悄执行的程序。
- 当连接到因特网上时，这个程序就会通知攻击者。攻击者利用这个潜伏在其中的程序，可以任意地窥视你整个硬盘中的内容，监听键盘敲击行为等，从而悄无声息地盗走用户的口令。

网络监听

- 如果口令在网络上明文传输，那么很容易通过网络监听来得到网络上传输的口令。
  - 如果是在共享是局域网内，用普通的Sniffer工具就可以嗅探到整个局域网内的数据包。
  - 如果是在交换式局域网中，可以用ARP欺骗来监听整个局域网内的数据。
  - 还可以在网关或者路由器上安装监听软件，从而监听通过网关或者路由器的所有数据包。

重放

- 为了防止传输过程中口令被监听，系统可能会对口令进行加密，黑客即使监听到了口令密文，也无法知道口令明文。
- 但是黑客可以把截取到的认证信息重放，从而完成用户登陆。

4.3典型的口令破解工具

- 4.3.1 口令破解器
- 4.3.2 候选口令产生器
- 4.3.3 操作系统的口令文件
- 4.3.4 口令破解工具
- 4.3.5 工具运用实例

4.3.1 口令破解器

- 口令破解器是一个程序，它能将口令解译出来，或者让口令保护失效。
- 事实上，很多加密算法是不可逆的，因此大多数的口令破解器一般并不是真正的去解码，而是通过尝试一个一个的单词，用知道的加密算法来加密这些单词，直到发现一个单词经过加密的结果和要解密的数据一样，那就认为这个单词就是要找到的密码了。

4.3.1 口令破解器(2)

- 由于许多人在选择口令时，技巧性都不是很好。一些人认为他的私人数据没有放在互联网上，口令选择比较随便，往往都是一些有意义的单词或者干脆就是用户名本身。这些使得口令破解器尝试的次数大为降低，口令破解比想象的有效的多。
- 另外，很多加密算法在选择密钥的时候都是通过随机数的方法产生的，但这种随机数往往都是伪随机数，并不是真正意义上的随机数，这为解密提供了一系列的方便。
- 从理论上来讲，任何口令都是可以破解的，只是一个时间的问题罢了。对于一些安全性较低的系统，破解的速度通常会很快。

4.3.2 候选口令产生器

- 口令破解器通常由候选口令产生器、口令加密模块和口令比较模块组成。
- 候选口令产生器用来产生认为可能是口令的单词。
- 在口令加密模块，使用知道的加密算法对候选口令加密，将加密后的候选口令密文与实际口令的密文一起送到口令比较模块进行比较，如果一致，那么，当前候选口令发生器中送出来的单词就是要寻找的口令，如果不一致，那么候选口令产生器再生成下一个候选口令。

4.3.2 候选口令产生器（2）

- 根据攻击方式的不同，产生候选口令有三种方法：
  - 一种是从字典里面读出一个单词，使用这种方法的原因是许多用户取密码并不是很明智，比如用一个很好记的单词。所以攻击者通常就会将这些单词收集到一个文件里，叫做字典。在破解密码时，就会从字典里选出候选密码。

字典攻击

4.3.2 候选口令产生器（3）

- 第二种方法是用枚举的方式来产生这样的单词。通常是从一个字母开始，一直增加，直到破解密码为止。这时，通常需要指定组成密码的字符值，比如从0~9、A~Z等等。为了便于协同破解密码，常常要为密码产生器指定产生密码的范围。例如……, aaa, aab, aac, …, aba, abb, ……

4.3.2 候选口令产生器（4）

- 第三种方法综合运用了前两种方法，它以字典为基础，对字典中的每个单词进行重新组合，如在单词后面接上数字、把两个单词拼在一起、在两个单词中间插入生日等等：security123, securitycomputer, security19811229computer。

组合攻击

4.3.3 操作系统的口令文件

- Unix类系统口令文件
- Windows系统口令文件

Unix的口令文件

- **UNIX**系统用户的口令，本来是经过加密后保存在一个文本文件**passwd**中的，一般存放在**/etc**目录下，后来由于安全的需要，把**passwd**文件中与用户口令相关的域提取出来，组织成文件**shadow**，并规定只有超级用户才能读取。这种分离工作也称为**shadow**变换。
- 因此，在破解口令时，需要做**UnShadow**变换，将**/etc/passwd**与**/etc/shadow**合并起来。在此基础上才开始进行口令的破解。

Unix的口令文件

- **/etc/shadow**文件包含用户的加密后口令相关的信息。每个用户一条记录。
- 记录的格式如下：**username:passwd:lastchg:min:max:warn:inactive:expire:flag**
  - username：登录名。
  - passwd：经过加密后的口令。
  - lastchg：表示从1970年1月1日起到上次更改口令所经过的天数。
  - min：表示两次修改口令之间至少要经过的天数。
  - max：表示口令的有效期，如为99999，则表示永不过期。
  - warn：表示口令失效前多少天内系统向用户发出警告。
  - inactive：表示禁止登录之前该用户名尚有效的天数。
  - expire：表示用户被禁止登录的天数。
  - flag：未使用。

Windows的口令文件

- Windows对用户账户的安全管理使用了安全账号管理器(Security Account Manager，简称SAM)的机制。
- SAM数据库在磁盘上保存在%systemroot%system32\config\目录下的sam文件中。
- SAM数据库中包含所有组、帐户的信息，包括密码的HASH、帐户的SID等。

Windows的口令文件(2)

- 黑客在攻入系统后往往渴望知道更多的秘密，而所有的用户信息都是保存在**SAM**文件中，这样，破解**SAM**也就是黑客接下来要做的。

Windows的口令文件(3)

- 在对**SAM**破解之前，我们首先要获取**SAM**文件，登陆**Windows**系统后**SAM**是被锁死的，我们可以用以下方法获取**SAM**文件：
  - 引导另一个操作系统：利用NTFS DOS的系统驱动来获得对NTFS硬盘的访问权限，抓出SAM。
  - 获取备份SAM：Windows会在%system root%\repair目录中备份一个SAM文件，多数管理员都会忘记删这些文件。

Windows的口令文件(4)

- Windows NT对同一用户口令采用两套单向散列函数进行运算，即单向LM散列算法和单向NT散列算法，两种算法的结果都保存在SAM文件中。
- 单向LM散列函数对口令的处理上存在缺陷。

Windows的口令文件(5)

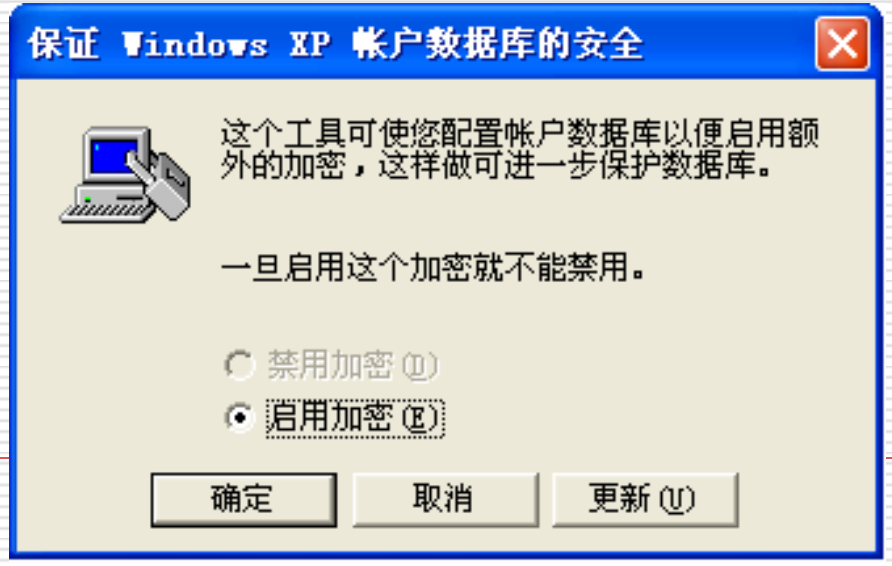
- LM对口令的处理
  - 首先，将用户口令中的字母都转换成大写字母。如果口令不足14位，则以0补足；如果超过14位，则通过截尾变成14位。
  - 然后，将其平均分成两组，每组7位，分别生成一个奇校验DES加密字。
  - 最后，利用一个固定值（已被破解出，以16进制表示为0x4b4753214023 2425）分别加密这两组DES加密字，将两组结果连接起来形成一个散列函数值。
- 如果一个用户口令为空，则经过这番运算，得到的LM散列值为：AAD3B435B51404EEAAD3B435B51404EE

Windows的口令文件(6)

- 考虑这样一个口令：**Af32mRbi9**，这个口令包含了大写字母、小写字母和数字，并且无规律，可以认为是符合安全要求的一个口令。但经过**LM**的处理后，**Af32mRbi9**就变成**AF32MRB**和**I900000**两部分，**LM**接下来将对这两部分分别进行加密处理。但这样一来，对口令破解程序来说，难度和运算时间就大大降低了，因为它只要破解两个**7**字符的口令，而且不需要测试小写字符情况。
- 对**Af32mRbi9**这个口令而言，原有的**9**位口令分成了两组，一组**7**位，一组**2**位，其穷举法组合以数量级的形式减少了！问题的关键点就仅在第一组的**7**位字符上了。这对攻击者来说，是再好不过的事情了。

Windows的口令文件(7)

- 微软在**win NT4**的**SP3**之后，提供**syskey.exe**来进一步加强**NT**的口令。
- 当**syskey**被激活，口令信息在存入注册表之前还会进行一次加密处理，以防止轻易破解口令。
- 在命令提示行下输入**syskey**即可配置：



4.3.4 口令破解工具

- Windows口令破解程序
- UNIX口令破解程序

Windows口令破解程序

- L0phtcrack
- NTSweep
- NTCrack
- PWDump2

L0phtcrack

- **L0phtcrack**是一个**Windows**口令审计工具，能根据操作系统中存储的加密哈希来计算**Windows**口令，功能非常强大、丰富，是目前市面上最好的**Windows**口令破解程序之一。
- 它可以从本地系统、其它文件系统、系统备份中获取**SAM**文件，从而破解密码。
- 它有四种方式可以破解口令：快速口令破解、普通口令破解、复杂口令破解、自定义口令破解。

快速口令破解

- 仅仅把字典中的每个单词和口令进行简单的对照尝试破解。只有字典中包含的密码才能被破解。

普通口令破解

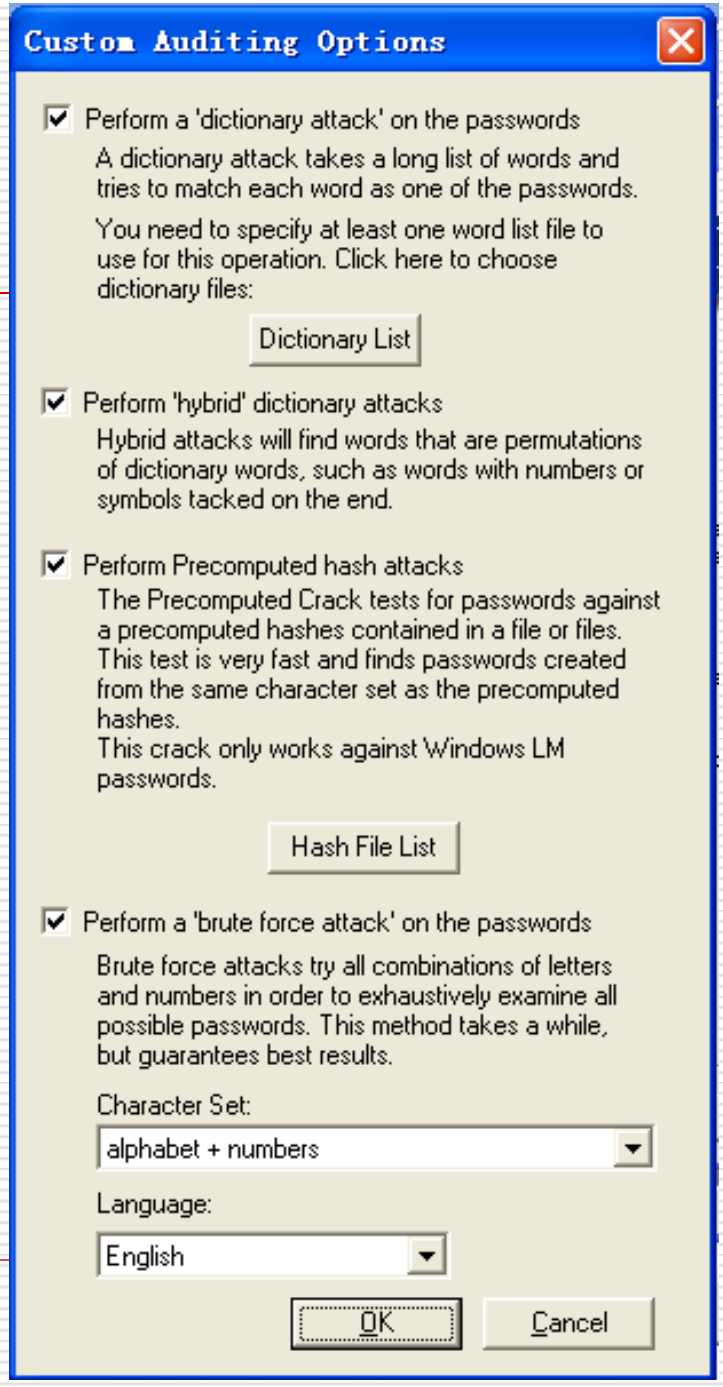
- 使用字典中的单词进行普通的破解，并把字典中的单词进行修正破解。

复杂口令破解

- 使用字典中的单词进行普通的破解，并把字典中的单词进行修正破解，并且执行暴力破解，把字典中的字、数字、符号进行尽可能的组合。

自定义口令破解

- 自定义的口令破解可以自己设置口令破解方式。
- 如右图所示，有四个选项可供选择。
- 下面介绍这四个选项。



自定义口令破解(2)

- 字典攻击（**dictionary attack**）可以选择字典列表进行破解；
- 混合破解（**hybrid attack**）把单词数字或符号进行组合破解；
- 预定散列（**precomputed hash attack**）利用预先生成的口令散列值与**SAM**中的散列值进行匹配；
- 暴力破解（**brute force attack**）可以设置为“字母＋数字”、“字母＋数字＋普通符号”、“字母＋数字＋全部符号”

NTSweep

- **NTSweep**是利用了**Microsoft**允许一个用户改变其它用户口令的机制，它使用的方法和其他口令破解程序不同，不是下载口令并离线破解。
- **NTSweep**首先取定一个单词，使用这个单词作为帐号的原始口令并试图把用户的口令改为此单词。如果主域控制机器返回失败信息，就可知道这不是原来的口令，接着取下一个单词测试。反之如果返回成功信息，就说明这一定是帐号的口令。
- 此方法很隐蔽，因为成功地把口令改成原来的值，用户不会知道口令曾经被人破解过。

NTCrack

- NTCrack是UNIX破解程序的一部分，但是在NT环境下破解。
- 它不象其他程序一样提取口令哈希，它和NTSweep的工作原理类似。必须给NTCrack一个user id和要测试的口令组合，然后程序会告诉用户是否成功。

PWDump2

- **WDump2**不是一个口令破解程序，但是它能用来从**SAM**数据库中提取口令哈希。

Unix口令破解程序

- Crack
- John the Ripper
- XIT
- Slurpie

Crack

- **Crack**是最著名的**Unix**系统上破解**UNIX**口令的工具之一。
- **Crack**是一个旨在快速定位**UNIX**口令弱点的口令破解程序。**Crack**使用标准的猜测技术确定口令。
- 它检查口令是否为如下情况之一：和**user id**相同、单词**password**、数字串、字母串。

Crack工具介绍

- **Crack** 的工作原理很简单。我们知道**Unix**加密口令是不会被解开的，这是因为加密算法是不可逆的。所以，一般的口令入侵是通过生成口令进行加密去匹配原口令密码，或直接从网上截获明文口令。
- **Crack** 程序中包含了几个很大的字典库，进行解破时它会按照一定的规则将字词进行组合，然后对之进行加密，再与要解破的加密口令匹配。所以运行**Crack**通常要占用大量的**CPU**，并要运行相当长的时间才结束。

Crack工具介绍(2)

以**Crack5.0**为例，**Crack5.0**的安装和使用比较简单，可执行下列几步：

- 修改Crack, 修改CC之类的参数。
- 执行 Crack –makeonly 生成可执行代码。
- 执行 Crack –makedict 生成字典。
- 执行 scripts/shadmrg.sv > passwd 将/etc/passwd 和/etc/shadow 文件合并。
- 执行Crack passwd 解破passwd中的口令。
- 执行 ./Reporter 查看解破结果。

John The Ripper

- John The Ripper这个软件是由著名的黑客组织UCF编写的，它支持UNIX、DOS、Windows。
- 对于老式的passwd文档（没有shadow），John可以直接读取并用字典穷举破解。
- 对于现代UNIX/Linux的passwd+shadow的方式，John提供了unshadow程序可以直接把两者合成出老式的passwd文件。

John The Ripper(2)

- **John The Ripper有四种破解模式:**
  - “字典文件”破解模式（Worldlist Mode）
  - “简单”破解模式（Single Creck）
  - “增强”破解模式（Incremental Mode）
  - “外挂模块”破解模式（External Mode）

“字典文件”破解模式

- 这是**John**所支持的破解模式中的最简单的一种，你要做的唯一的工作就是告诉**John**字典文件在哪，好让它取出破解。
- 在“字典文件”破解模式里可以使用“字词变化”功能，让这些规则自动的套用在每个读入的单词中，以增加破解的几率。
- 如字典中有单词**cool**，则**John**还会尝试使用**cooler**，**coOl,Cool**等单词进行解密。

“简单”破解模式

- “简单”破解模式是专门针对“使用账号做密码”的懒人所设计的。
- 所谓“使用账号当做密码”的意思是，如果一个使用者的账号是“**John The Ripper**”，那么他的密码也会取为“**John The Ripper**”。
- 在“简单”破解模式里**John**会拿密码文件内的“账号”字段等相关信息来破解密码，并且使用多种“字词变化”的规则套用到“账号”内，以增加破解的几率。
- 如账号“**John**”,他会尝试使用**John**、**John0**、**njoh**、**j0hn**等规则变化来尝试密码的可能性。

“增强”破解模式

- 这是**John**里面功能最强大的破解模式，他会自动尝试所有的可能字符组合，然后当做密码来破解。
- 这个破解模式所需要的时间非常的冗长，因为要尝试全部组合字符是非常耗费时间的，所以**John**才会定义一些“字符频率表”（**character frequency tables**）来帮助破解。
- 简言之，这个破解方法就是暴力法，把所有的密码组合都试一次，来得到正确的结果。

“外挂模块”破解模式

- 这个破解模式是让使用者可以自己用C编写一些“破解模块程序”，然后，挂在John里面来使用。
- 其实所谓的“破解模块程序”就是一些用C语言设计好的函数。他的功能就是产生一些单词来让John尝试破解。而在执行John程序时，他在加载这些“破解模块程序”时会自动编译这些C函数来使用。

XIT

- XIT是一个执行词典攻击的UNIX口令破解程序。XIT的功能有限，因为它只能运行词典攻击，但程序很小、运行很快。

Slurpie

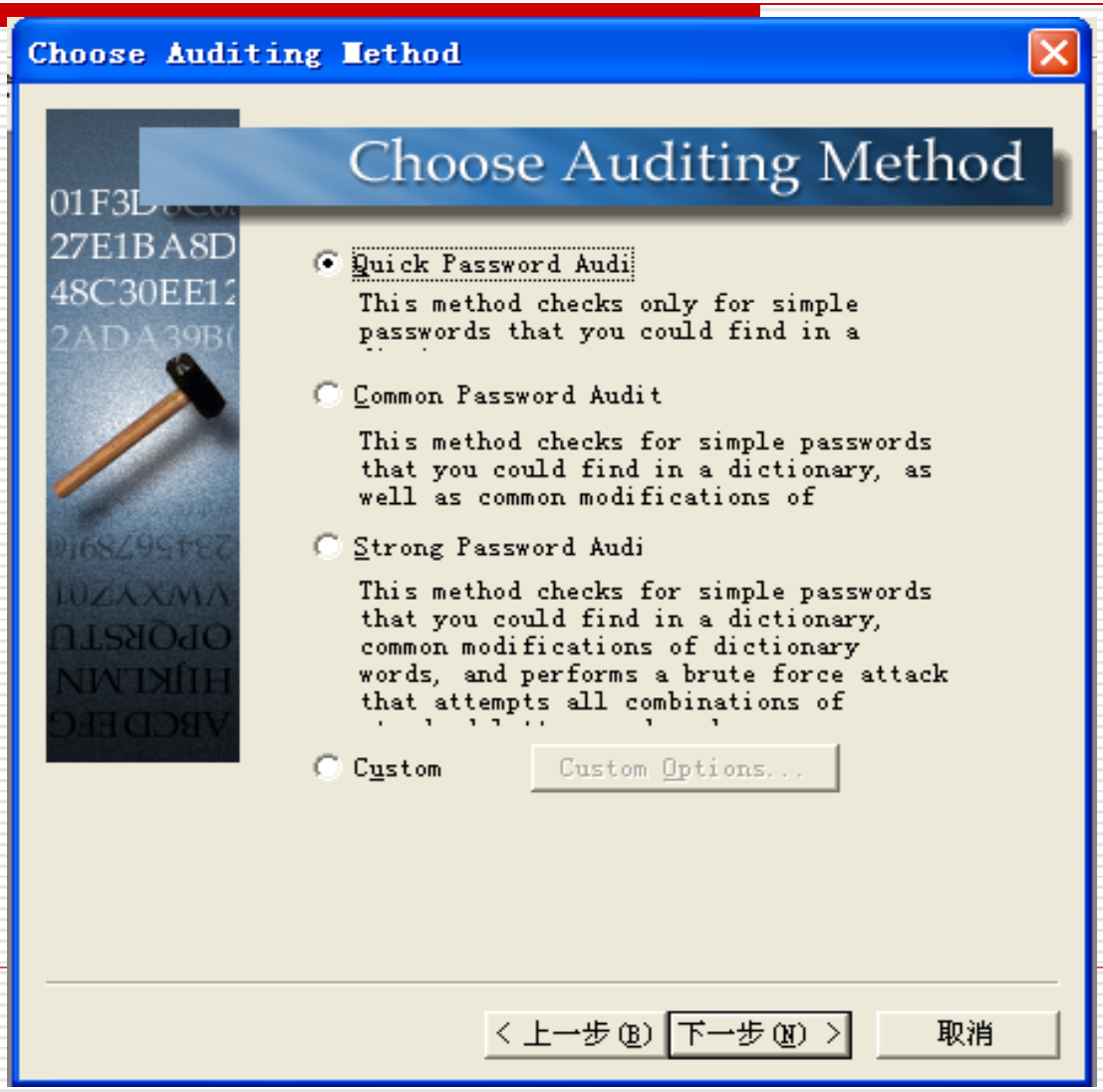
- Slurpie能执行词典攻击和定制的强行攻击，要规定所需要使用的字符数目和字符类型。

4.3.5 工具运用实例

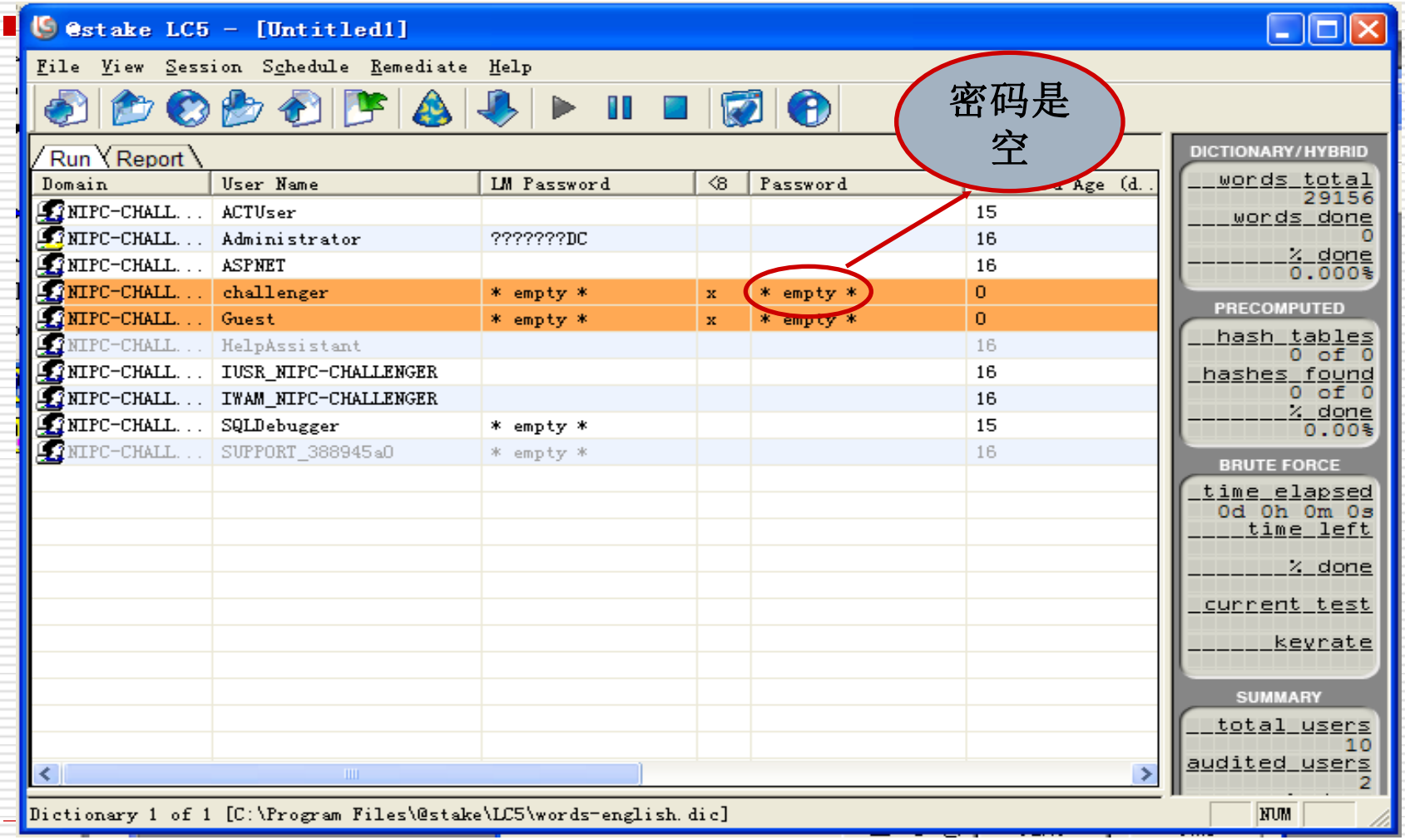
- 用**L0phtCrack V5.02(LC5)**破解计算机内**challenger**用户的口令：
  - 事先在主机内建立用户名challenger，密码依次设置为空密码、security、security123进行测试。
  - 本实例在window xp sp2上测试通过。

选择导入加密口令的办法—从本地导入

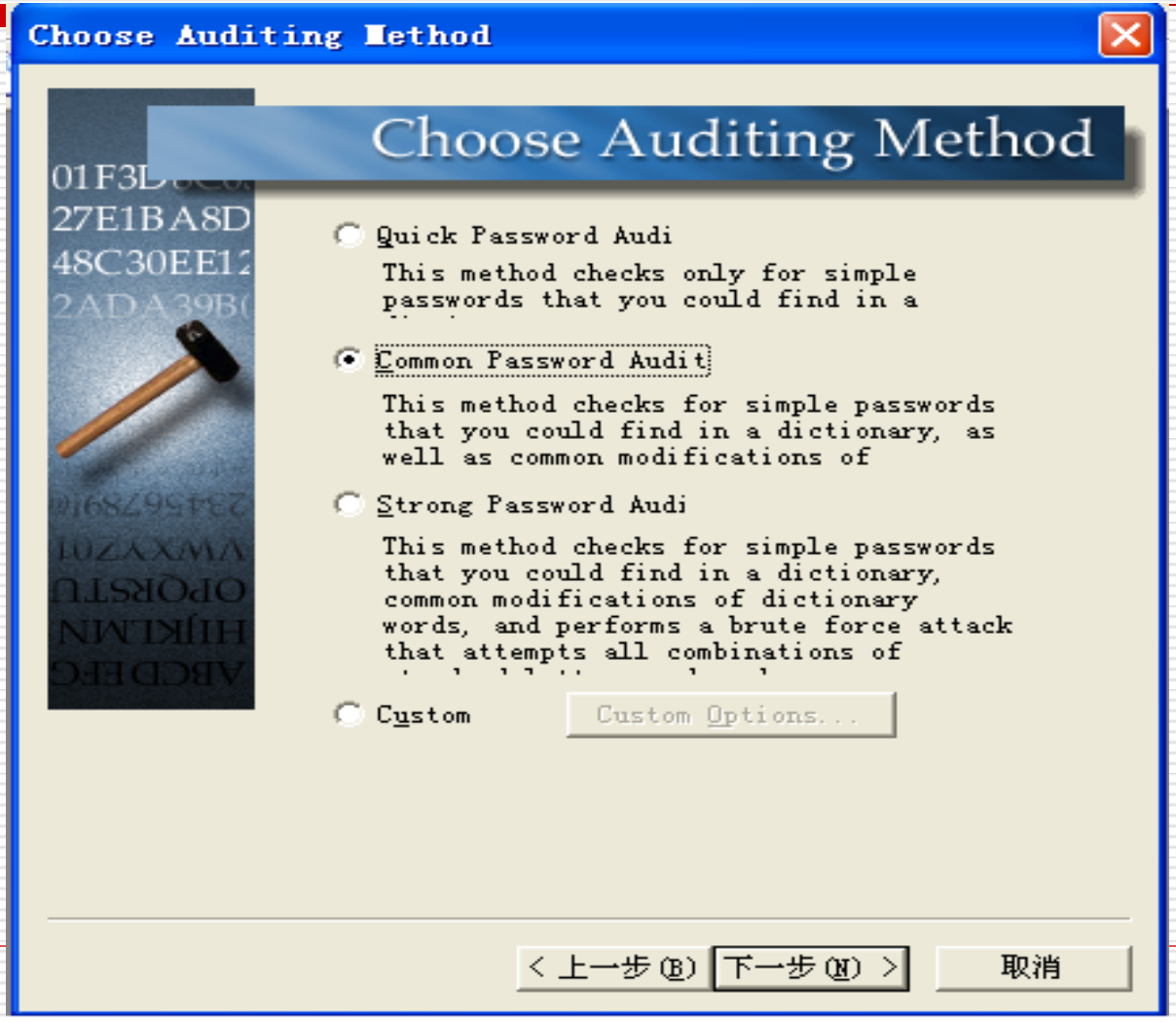
选择破解办法为快速口令破解



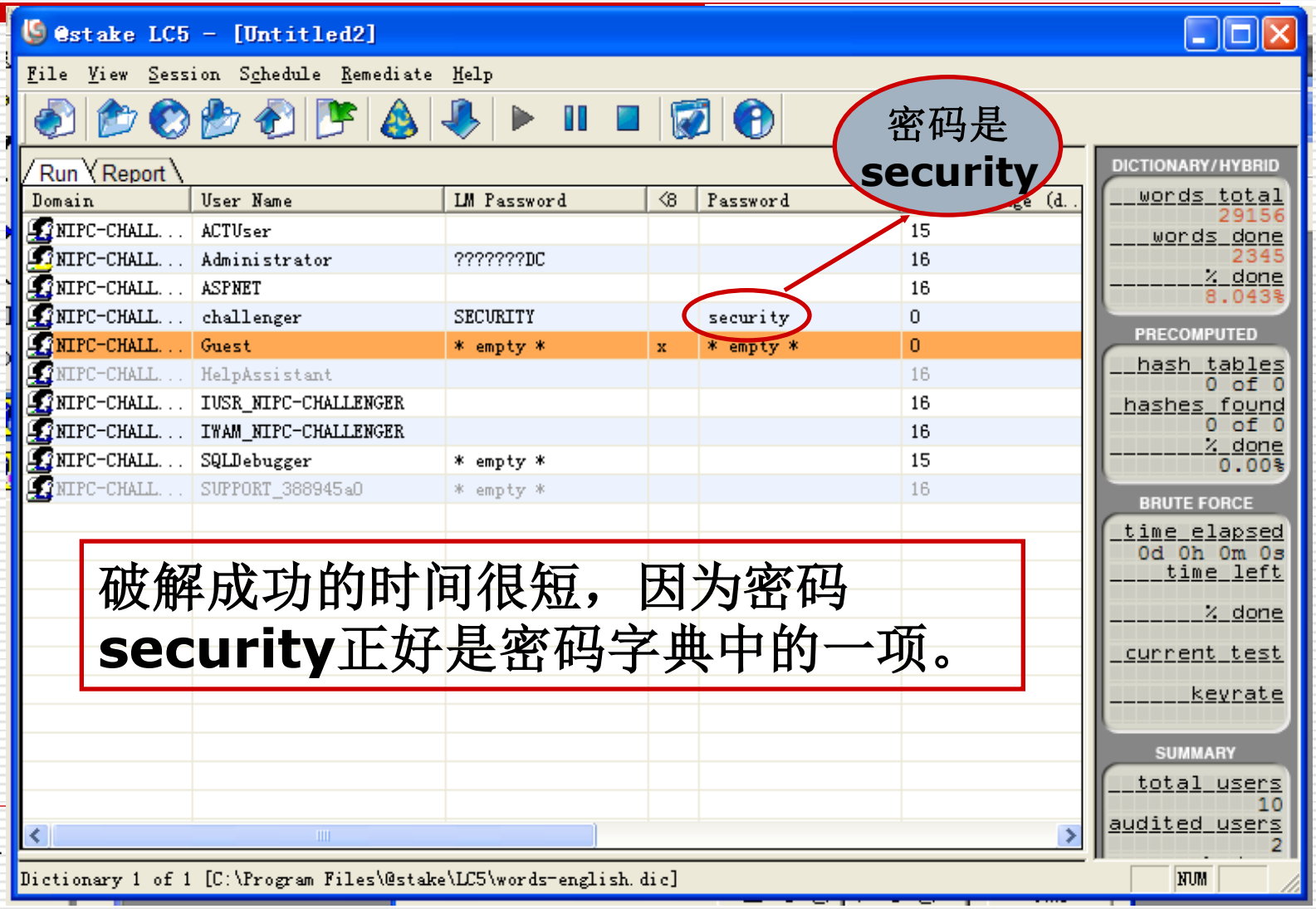
密码为空的破解结果


@stake LC5 - [Untitled1]
File View Session Schedule Remediate Help
密码是空
Run Report
Domain | User Name | LM Password | <8 | Password | Age (d..
NIPC-CHALL... | ACTUser | | | | 15
NIPC-CHALL... | Administrator | ???????DC | | | 16
NIPC-CHALL... | ASPNET | | | | 16
NIPC-CHALL... | challenger | * empty * | x | * empty * | 0
NIPC-CHALL... | Guest | * empty * | x | * empty * | 0
NIPC-CHALL... | HelpAssistant | | | | 16
NIPC-CHALL... | IUSR_NIPC-CHALLENGER | | | | 16
NIPC-CHALL... | IWAM_NIPC-CHALLENGER | | | | 16
NIPC-CHALL... | SQLDebugger | * empty * | | | 15
NIPC-CHALL... | SUPPORT_388945a0 | * empty * | | | 16
DICTIONARY/HYBRID
words total 29156
words done 0
% done 0.000%
PRECOMPUTED
hash tables 0 of 0
hashes found 0 of 0
% done 0.00%
BRUTE FORCE
time elapsed 0d 0h 0m 0s
time left
% done
current test
keyrate
SUMMARY
total users 10
audited users 2
Dictionary 1 of 1 [C:\Program Files\@stake\LC5\words-english.dic]
NUM

更改破解方法为普通口令破解

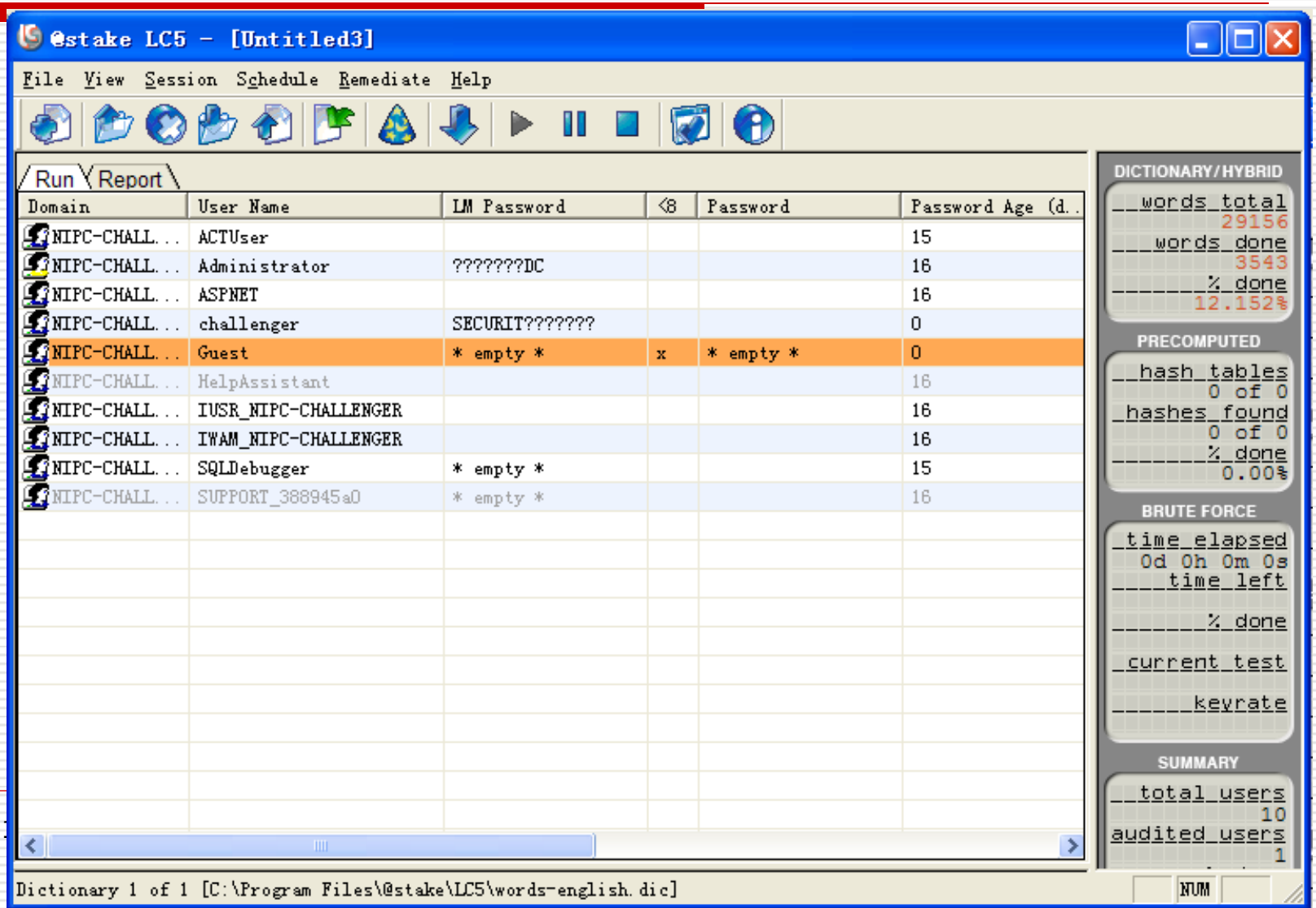


密码为security的破解结果

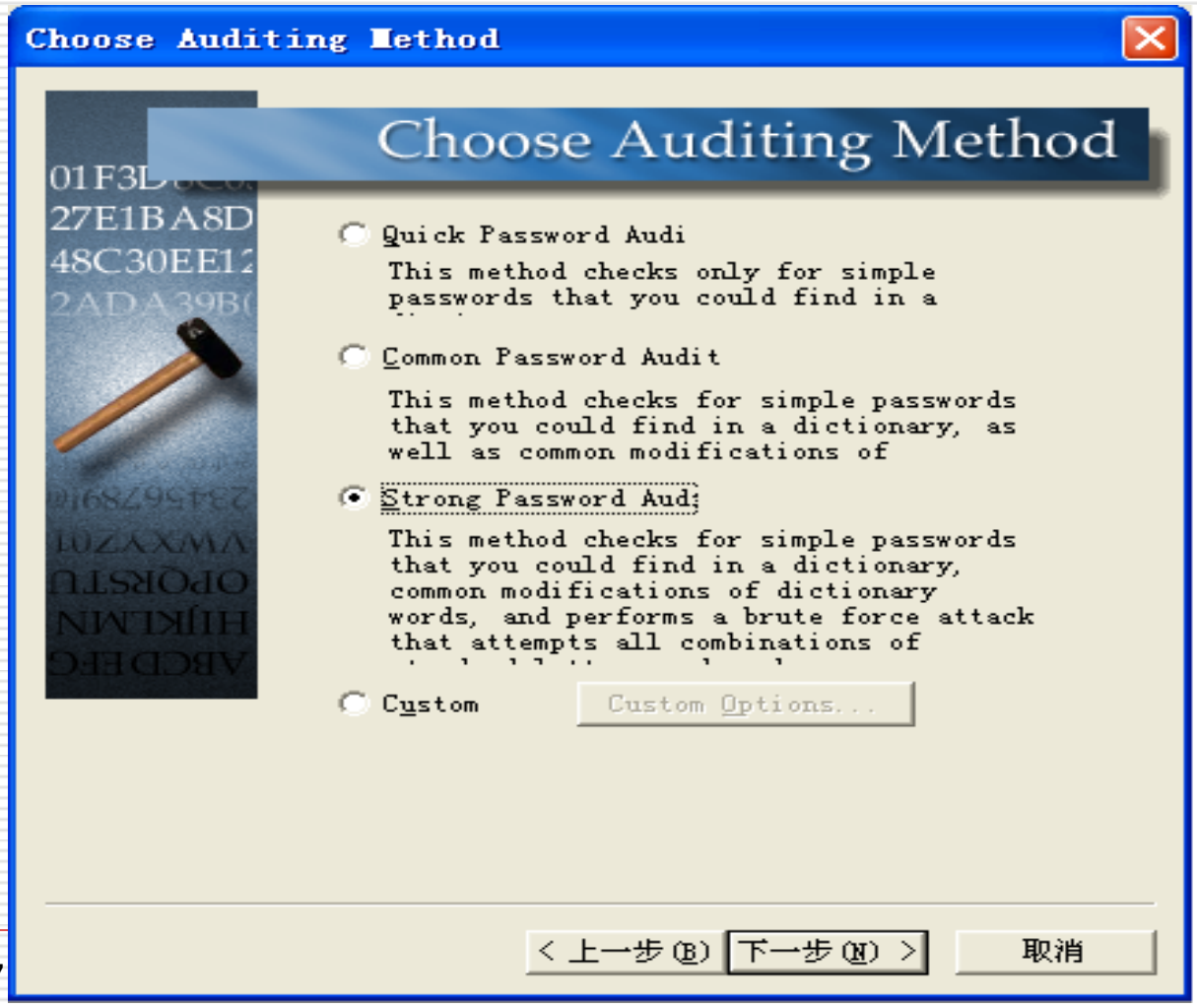


破解成功的时间很短，因为密码**security**正好是密码字典中的一项。

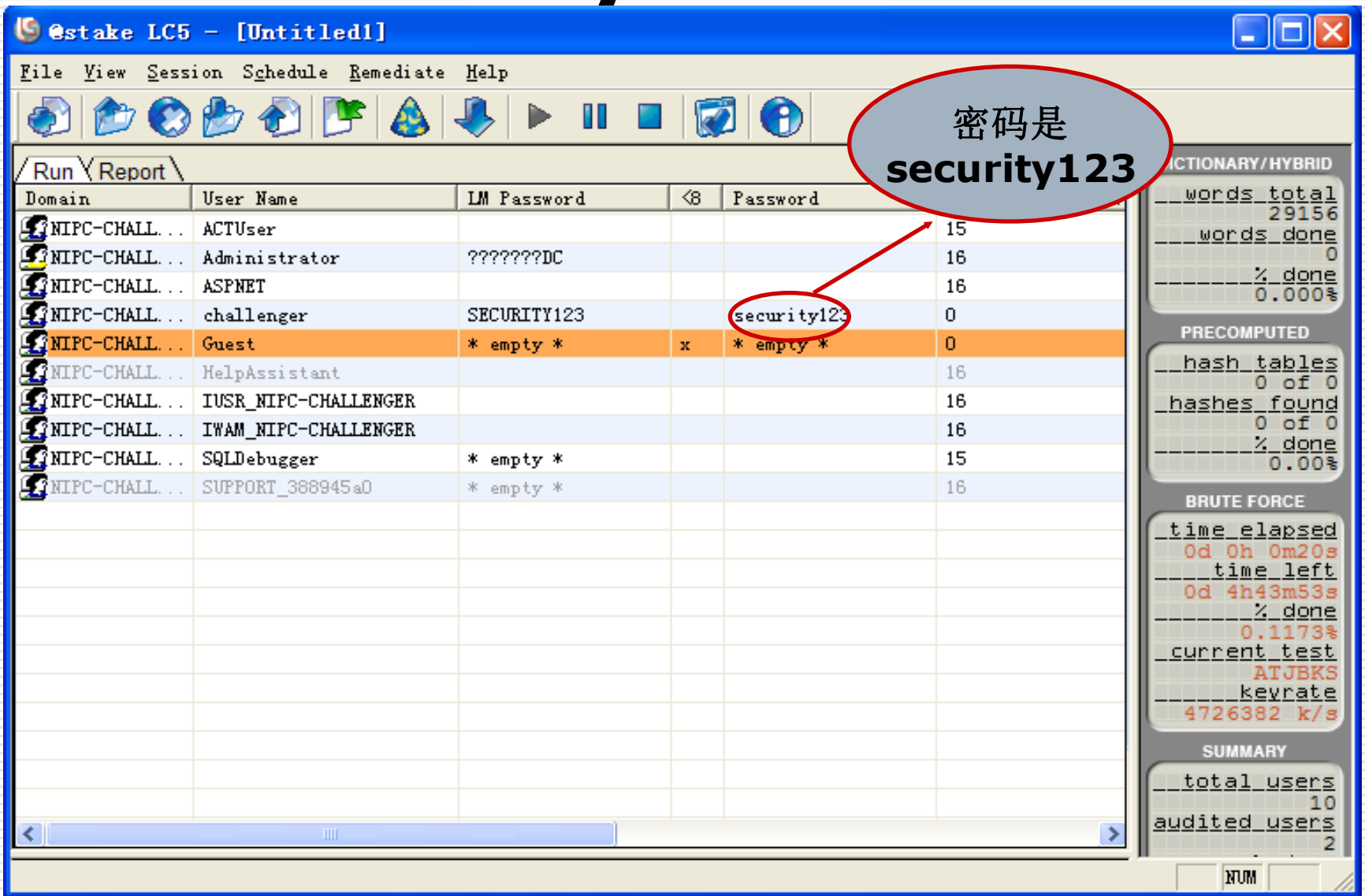
密码为security123的破解结果—无法破解



更改破解方法为复杂口令破解



密码为security123的破解结果



4.4 口令攻击的综合应用

- 本地口令攻击
  - Windows系统口令攻击
  - Unix系统口令攻击
- 远程口令攻击

Windows系统口令攻击

- 背景知识储备
- **Windows NT, 2000**口令攻击
- **Windows XP, 2003**口令攻击

可以在课后试试这里介绍的方法

背景知识储备

- 前面介绍了用**L0phtCrack**破解**Windows**口令的例子，如果你已经是一个计算机的用户，你可以用这个办法得到同一个计算机内其它用户的口令。
- 我们这里将介绍更通用的**Windows**口令攻击方法：即使你不是计算机的用户，也可以获取**Windows**的口令从而进入系统。
- 当你忘记了操作系统的密码时，也可以用这里介绍的方法来进入操作系统。

背景知识储备（2）

- **Windows NT、Windows 2000、Windows XP以及Windows 2003中对用户账户的安全管理使用了安全账号管理器(Security Account Manager，简称SAM)的机制，安全账号管理器对账号的管理是通过安全标识进行的，安全标识在账号创建时就同时创建，一旦账号被删除，安全标识也同时被删除。**
- **安全标识是唯一的，即使是相同的用户名，在每次创建时获得的安全标识都是完全不同的。**

背景知识储备（3）

- 安全账号管理器的具体表现就是**%SystemRoot%\system32\config\sam**文件。**sam**文件就是**Windows**的用户账户数据库，所有**Windows**用户的登录名及口令等相关信息都会保存在这个文件中。
- 如果我们用编辑器打开这些**Windows**的**sam**文件，除了乱码什么也看不到。因为**Windows**系统中将这些资料全部进行了加密处理，一般的编辑器是无法直接读取这些信息的。

Windows NT, 2000口令攻击

- Windows NT和2000的口令攻击很简单。
- 在NT和2000的操作系统中，如果从DOS启动然后删除了sam文件，则当系统重新启动时，会默认生成一个sam文件，并将管理员密码置为空。
- 这样就能轻松登陆系统了。

Windows XP, 2003口令攻击

- 用于NT和2000的方法对XP和2003的系统并不奏效。
- 因为如果不小心删除了sam文件，系统将无法启动，除非将备份的sam文件(在%SystemRoot%\repair目录下)恢复回来。
- 但是我们知道了上述四种版本的Windows系统的账户密码是基于SAM机制的，那么对密码的破解就可以针对SAM文件进行。

Windows XP, 2003口令攻击（2）

- 攻击方法（同样适用于NT和2000系统）
  - 提取SAM文件进行破解
  - 用备份的SAM文件替换当前SAM文件
  - 使用口令修改软件
  - 替换屏保程序

提取SAM文件进行破解

- 用DOS启动盘启动计算机，并把SAM文件拷贝出来，用软件进行分析破解，则可以得到系统的口令。
- 有很多软件都有分析SAM文件的功能，如L0phtCrack，由于L0phcrack功能强大，密码的破译只是一个迟早的事。

用备份的SAM文件替换当前SAM文件

- 攻击者用DOS启动盘启动后，还可以用%SystemRoot%\repair\asm覆盖%SystemRoot%\system32\config\sam，这样系统管理员Administrator的密码就恢复成安装操作系统时的密码了，而大部分人在安装操作系统时都将初始密码设置为空或者很简单。

使用口令修改软件

- 目前有许多相应的软件可以在不登陆系统的情况下修改系统密码，如**Passware Kit Enterprise**这款软件可以自动把**administrator**密码修改为**12345**。
- **Passware Kit Enterprise**可以找回多种办公室应用程序档案失去或忘记的密码，包括**Excel**、**Word**、**Windows 2003/XP/2K/NT**、**Lotus Notes**、**RAR**、 **WinZip**、**Access**、**Outlook**、**Acrobat**、**Quicken**、**QuickBooks**、**WordPerfect**以及**VBA**，在此我们只需要使用其中的**Windows KEY**功能。

使用口令修改软件（2）

- **Windows KEY**运行后，在软驱中放一张空白软盘，生成一张恢复系统管理员的密码盘，其中共**3**个文件，分别是**Txtsetup.oem**、**Winkey.sys**和**Winkey.inf**。
- 现在用**Windows**安装光盘启动电脑，启动过程中按**F6**键让系统采用第三方驱动程序。此时，放入该软盘就会自动跳到**Windows KEY**的界面。这时它会强行把**Administrator**的密码换成“**12345**”。
- 当你重新启动以后，会要求再次修改密码。

替换屏保程序

- 这个方法是利用当系统在长时间没有动作时，启动屏幕保护程序的特点。
- 利用这种方法的前提是我们还是需要有一张启动盘，如果系统盘是**FAT**格式，则普通的**DOS**启动盘就行了；如果是**NTFS**格式，则需要一张能够识别**NTFS**文件的**DOS**启动盘。

替换屏保程序（2）

- 我们可以把“**%systemroot%\system32\logon.scr**”替换为“**cmd.exe**”，然后在系统登陆处等待，过一会，系统就会去运行“**logon.scr**”这个屏保，因为你替换了这个屏保文件，所以实际上运行的是“**cmd.exe**”，并且是“**localsystem**”权限，现在你就可以破解密码了。
- 最简单的就是在“**cmd.exe**”里运行“**net user administrator**”，成功后管理员密码被清空，这样就删除了管理员帐户的密码。

Unix系统口令攻击

- Unix的加密口令是很难逆向破解的，黑客们常用的口令入侵工具所采用的技术是仿真对比，利用与原口令程序相同的方法，通过对比分析，用不同的加密口令去匹配原口令。
- 下面介绍口令破解工具Crack的主要工作流程。

Crack破解Unix口令的流程

- 它采用逆向比较法进行口令破解。
  - ① 准备，对口令文件作**UnShadow**变换。
  - ② 下载或自己生成一个字典文件。
  - ③ 穷举出口令字典中的每个条目，对每个单词运用一系列规则，如大小写交替使用，在单词的开头或结尾加上一些数字。
  - ④ 调用**crypt()**函数对使用规则生成的字符串进行加密变换。
  - ⑤ 取出密文口令，与**crypt()**函数的输出进行比较。
- 循环**3**到**5**步，直到口令破解成功。

远程口令攻击

- 远程口令攻击主要是指网络服务口令攻击，是一种远程在线攻击。
- 许多网络服务，都是通过账号/口令来认证需要访问该服务的用户。
- 如**Email，Telnet，FTP，HTTP**等。
- 可以远程进行穷举字典的方式来猜解口令。
- 破解效率很低，而且容易被记录。

远程口令攻击（2）

- 攻击过程大致如下：
  - ① 建立与目标网络服务的网络连接。
  - ② 选取一个用户列表文件和一个字典文件。
  - ③ 在用户列表文件和一个字典文件中，选取一组用户和口令，按照网络服务协议规定，将用户名和口令发给目标网络服务端口。
  - ④ 检测远程服务返回信息，确定口令尝试是否成功。
- 循环2到4步，直到口令破解成功为止。

4.5 口令攻击的防御

- 4.5.1 口令攻击防御概述
- 4.5.2 强口令的选取方法
- 4.5.3 保护口令的方法
- 4.5.4 一次性口令技术
- 4.5.5 生物技术

4.5.1 口令破解防御概述

- 防范办法很简单，只要使自己的口令不在英语字典中，且不可能被别人猜测出就可以了。
- 一个好的口令应当至少有8个字符长，不要用个人信息（如生日、名字等），口令中要有一些非字母字符（如数字、标点符号、控制字符等），还要好记一些，不能写在纸上或计算机中的文件。

4.5.1 口令破解防御概述

- 保持口令的安全要点如下：
  - 不要将口令写下来
  - 不要将口令存于电脑文件中
  - 不要选取显而易见的信息作口令
  - 不要让别人知道
  - 不要在不同系统上使用同一口令
  - 为了防止眼捷手快的人窃取口令，在输入口令时应当确定无人在身边
  - 定期更换口令，至少6个月要改变一次

4.5.2 强口令的选取方法

- 强口令的定义差别很大，它和单位的业务类型、位置、雇员等等的因素有关。强调这一点是因为会因所处的环境不同而差别很大。定义也会因技术的增强而变化。
- 比如说，五年前曾被认为是强口令，现在很可能就会变成弱口令。导致这种变化的主要原因就是计算机系统比五年前的计算机系统要更快和更便宜。五年前用最快的计算机破解要花几年的时间的口令，现在只要不到一个小时的时间就解开了。

4.5.2 强口令的选取方法

- 什么才是强口令呢？基于目前的技术，强口令必须具备以下的特征：

  1、每**45**天换一次

  2、口令至少包含**10**个字符

  3、必须包含字母、数字、特殊的符号

  4、字母、数字、特殊符号必须混合起来，而不是添加在尾部

  5、不能包含词典单词

  6、不能重复使用以前的五个口令

  7、一定次数登陆失败后，口令在一段时间封闭

4.5.2 强口令的选取方法

- 提议用户用句子而不是用单词作为口令。这就要选取一个容易记忆、不含词典中的单词、含有数字和特殊字符的口令。
- 例如，使用每个单词的第一个字母作为口令。比如说，如果口令**wIsmtIs#￥%*5t**，如果就这样记的话是非常困难的，但是如果你记住这句话**“When I stub my toe I say “#￥%*” 5 times”**（我的脚趾头被绊时我说了**5**次**“#￥%*”**），这样的话口令可能就会被记住了。简单的取每个单词的首字母，就组成了一个口令。

4.5.3 保护口令的方法

- 强口令的选取是从用户的角度来说的，那么，对于系统来说，口令的安全又是如何得到保障的呢？
- 系统中存的任何口令都必须受到保护，防止未授权泄漏、修改和删除。

4.5.3 保护口令的方法

- 未授权泄漏在口令安全中占有重要的地位。如果攻击者能得到口令的副本，则读取口令后，他就能获得系统访问权。
- 这就是为什么强调用户不能将口令写下或者透漏给同事的原因。如果攻击者能得到口令的副本，他会变成合法用户，所做的一切最后都会追踪到那个合法用户身上。

4.5.3 保护口令的方法

- 未授权修改也很重要，因为即使攻击者无法读到口令，但是可用他所知道的单词修改口令，这样你的口令变成了攻击者知道的值，他不需要知道实际口令就能做到这一点。
- 这在各种操作系统中成了主要问题。

4.5.3 保护口令的方法

- 未授权的删除也很重要，因为攻击者删除帐号，或者导致拒绝服务攻击，或者用他知道的口令重新创建该帐号。
- 比方说，攻击者在周末闯入了系统并删除了所有的用户帐号，这就产生了一次拒绝服务攻击，因为星期一所有人都无法登录系统，被系统拒绝访问。

4.5.3 保护口令的方法

- 要保护口令不被未授权泄漏、修改和删除，口令就不能按纯文本方式存放在系统内，如果系统中存放有包含所有口令的文本文件，很容易被某些人读取并获得所有人的口令。
- 保护口令的一个很重要的方法就是加密。加密能隐藏原始文本， 所以如果有人得到了加密口令，也无法确定原始口令。

4.5.3 保护口令的方法

- 密码学最基本的形式使把明文隐藏为密文的过程，目的是使它不可读。在这里，明文是原始消息或者可读口令，密文是加密的或者不可读的版本。
- 下页是一加密的例子。

4.5.3 保护口令的方法

可以看到，加密后消息很难读：

This is a plaintext message. Here is the corresponding message encrypted with Pretty Good Protection (PGP);

QANQRIDBwU4DoGKRq+1ZHbYQB/0dgBvp6axtop9zu2A6Yb964CJcpZ5Ci9N1W/6B
Pbu3qitff/M9I1dSoNtFuMcQMvxK5c7R4+qmPM7pgsXaRYEBjuA9Cdei2qp4bOhl
KJRaM/cCRLBWdBP8UUocfRk3jHxg6cwy9QwVVwCZ7LL+6Rqt9kohdbALVENY/XnL
9wP4QcJ3klyjznxB0t9yF1Dnshpzvs0HcdxK3CT19U1k8n+Sw0J+mv0eOv3uqbRa
Cuyo5Z3zZeyGttfYaDBXDIPq6qouNIaxz+9cRtA7y5jNfLPdYmPzrwVsz0IGfMzA
1Bf3ByMieQt/QSdMFhkihI89AT2qVSeyosIgWpCXFaB468bxCADtN7h6BWaCNEV0hSsJo
6O9uv8v1OlKfxBpdnXvsMZxrA4yTATfO3xnxmRp4kXMlmPElPxSzBId2VqrIJZ/HZfxbyW
KZGSUQuG62228xDPWhYQBeKvyACUXzguHgddTO3+XYFxWgUdV8mNi4twA2hdapu
AUZSyuIsnGa0yhpXPOzEUrYwKV/hxL4cUkzxVzr9Hf9qTbVd/TrFqF0wrbFvb2m65i++H2
w73w3PlnKvKNiPyJ8iFsLLXyfzgmOtF6QYaeBqBIp31Hd3s+GAqJxs07jxm+ba+slJqLzZDJ
pzZDJpc/hyn6dpjyD0Wx6myfGaZuN4a6W3JIr8xlB1O/e+saFwexnyTNwySfcL6sOQQN3Rs
0ucws3ORJKIEqxJnfcXwfoSILZYFwz2ucrTZMShEnETMCuW

4.5.3 保护口令的方法

- 通常有三种加密类型：
  - 对称或单密钥加密
  - 不对称或双密钥加密
  - 哈希（即散列，hash）

对称加密

- 对称加密方法加密和解密都使用同一个密钥。
- 如果我加密一条消息让你来破解，你必须有与我相同的密钥来解密。这和典型的门锁相似。如果我用钥匙锁上门，你必须使用同一把钥匙打开门。
- 对称加密的优点是速度快，缺点是在通信之前用户需要有安全的信道交换密钥。

不对称加密

- 不对称加密使用两个密钥克服了对称加密的缺点：公钥和私钥。
- 私钥仅为所有者所知，不和其他任何人共享；公钥向所有会和用户通信的人公开。
- 用用户的公钥加密的东西只能用用户的私钥解开，所以这种方法相当有效。别人给用户发送用用户的公钥加密的信息，只有拥有私钥的人才能解开。

不对称加密

- 公钥加密的优缺点：
  - 优点：是在通信前用户不需要安全信道交换密钥。
  - 缺点：速度太慢。

不对称加密

- 在安全通信中，多数系统结合使用对称加密和不对称加密，来利用两种方法的优点。
- 可以先使用不对称加密发起会话，交换会话密钥。因为这个会话密钥用公钥加密，并用私钥解密，它是安全的。
- 会话密钥交换后。它就在后面的会话中用于对称加密，因为对称加密快得多。

哈希函数

- 哈希函数被认为是单向函数，因为它们只做信息的单向不可逆变换。
- 给定一个输入字符串，哈希函数产生等长的输出字符串，而且无法从输出串确定原来的输入串。

哈希函数

- 哈希函数似乎是存储口令的最佳选择，因为它们没有什么可担心的因素。同时，因为不可逆，无法得到原始口令。
- 既然不可逆，在人们每次登录时要如何取回原来的口令来验证呢？用户每次登录到系统输入口令时，系统会取出输入的口令文本，计算哈希值，并与存储的哈希值比较。如果相同，用户一定输入了正确的口令。反之，用户则是输入了错误的口令。

4.5.4 一次性口令技术

- 仅从字面上理解，一次性口令技术好像要求用户每次使用时都要输入一个新的口令。
- 但事实正相反，用户所使用的仍然是同一个口令。

4.5.4 一次性口令技术

- 一次性口令技术采用的是挑战—响应机制。
- 一次性口令的工作原理：

  1) 首先，在用户和远程服务器之间建立一个秘密，该秘密在此被称为“通行短语”，相当于传统口令技术当中的“口令”。

  同时，它们之间还具备一种相同的“计算器”，该计算器实际上是某种算法的硬件或软件实现，它的作用是生成一次性口令。

4.5.4 一次性口令技术

2) 当用户向服务器发出连接请求时，服务器向用户提示输入种子值。种子值（**seed**）是分配给用户的在系统内具有唯一性的一个数值，也就是说，一个种子对应于一个用户，同时它是非保密的；

可以把种子值形象地理解为用户名。

4.5.4 一次性口令技术

3) 服务器收到用户名之后，给用户回发一个迭代值做为“挑战”。迭代值（**iteration**）是服务器临时产生的一个数值，与通行短语和种子值不同的是：它总是不断变化的。

可以把迭代值形象地理解为一个随机数。

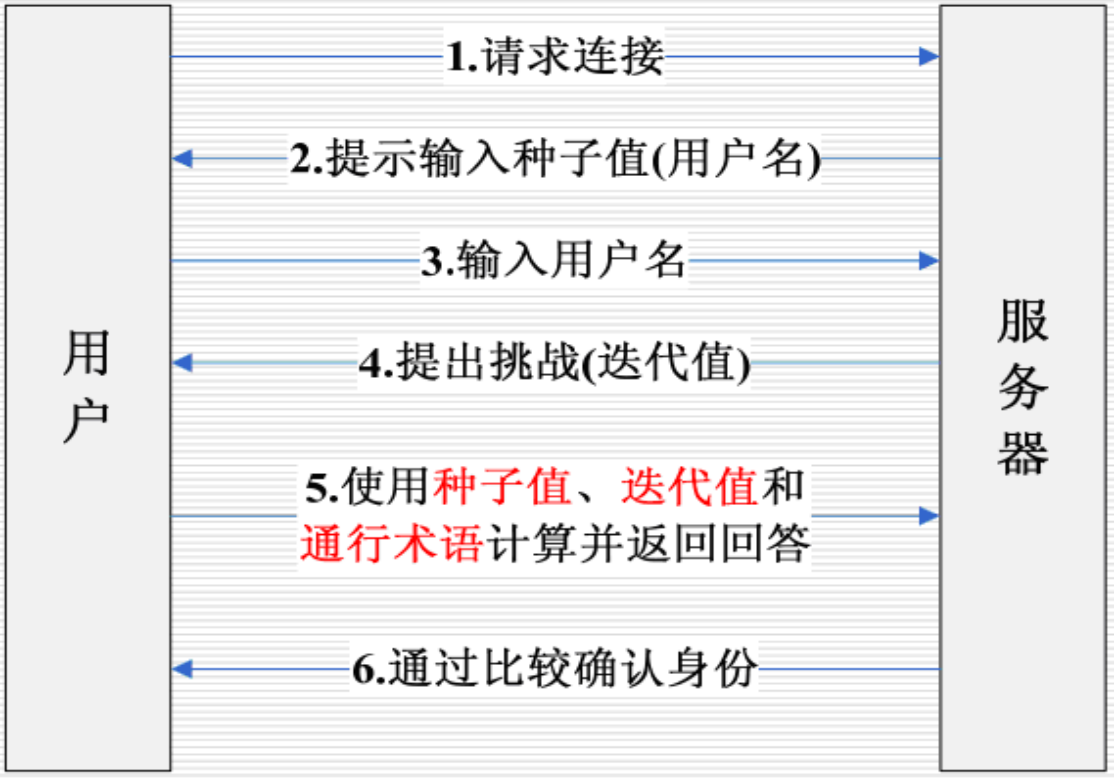
4.5.4 一次性口令技术

4) 用户收到挑战后，将种子值，迭代值和通行短语输入到“计算器”中进行计算，并把结果作为回答返回服务器。

5) 服务器暂存从用户那里收到的回答。因为它也知道用户的通行短语，所以它能计算出用户正确的回答，通过比较就可以核实用户的确切身份。

4.5.4 一次性口令技术

- 以上过程可用下图表示：



4.5.4 一次性口令技术

- 我们可以看出，用户通过网络传给服务器的口令是种子值，迭代值和通行短语在计算器作用下的计算结果，用户本身的通行短语并没有在网上传播。
- 只要计算器足够复杂，就很难从中提取出原始的通行短语，从而有效地抵御了网络监听攻击。又因为迭代值总是不断变化的，比如每当身份认证成功时，将用户的迭代值自动减**1**，这使得下一次用户登录时使用鉴别信息与上次不同（一次性口令技术由此得名），从而有效地阻止了重放攻击。

4.5.4 一次性口令技术

- 总之，与传统口令技术的单因子（口令）鉴别不同，一次性口令技术是一种多因子（种子值，迭代值和通行短语）鉴别技术，其中引入的不确定因子使得它更为安全。

4.5.5 生物技术口令

- 随着生物技术和计算机技术的发展，人们发现人的许多生理特征如指纹、掌纹、面孔、声音、虹膜、视网膜等都具有惟一性和稳定性，每个人的这些特征都与别人不同，且终身不变，也不可能复制。这使得通过识别用户的这些生理特征来认证用户身份的安全性远高于基于口令的认证方式。
- 人们发展了指纹识别、视网膜识别、发音识别等多种生物识别技术，其中以指纹识别技术发展得最为火热。指纹**ATM**、指纹电子商务、指纹网络管理、指纹加密文件，指纹财务管理、指纹**ERP**，就连现在许多笔记本上都使用了指纹识别器。
- 由于计算机在处理指纹时，要将生理特征转化为数据，只涉及了指纹的一些有限信息，而且对比算法不能保证**100%**精确匹配，因此，在应用系统的设计中，要充分考虑识别率（包括漏判和误判）的问题。

4.6 小结

- 本章的主要内容就是关于口令的攻击和防御，介绍了口令的重要性和口令的供给方法和防御。
- 攻击方法主要介绍了词典攻击、强行攻击和组合攻击三种比较典型的攻击方法，还有一些其它的攻击方法。
- 而防御的方法主要是介绍了强口令、加密和一次性口令的方法。

谢谢各位！