UNIX系统安全实用技术指南

●[美] PATRICK H. WOOD, STEPHEN G. KOCHAN 著

●刘昊宇 王 哲 赵爱利 汤连生 译

●刘尊全 审校

電子工業出版社

UNIX 系统安全实用技术指南

[美]PATRICK H.WOOD, STEPHEN G.KOCHAN 著

刘昊宇　王　哲
赵爱利　汤连生　译

刘尊全　审校

电子工业出版社

（京）新登字 055 号

内容提要

本书以 UNIX 系统 V 为背景详尽地介绍了 UNIX 系统的安全内容。全书共分六章，内容包括引言、美国计算机安全中心首席科学家 Robert Morris 在美国众议院就计算机安全问题的一篇证词、用户安全技术、程序员安全技术、管理员安全技术、网络安全技术，附录给出了安全命令和函数、8 个实用的安全程序及一份关于 SUID / SGID 权限的专利报告。

本书是计算机安全方面的一本专著，可作为大学计算机专业高年级学生、研究生的教材，也适用于广大计算机用户、程序员、管理员作为学习计算机安全的参考用书。

UNIX 系统安全实用技术指南

[美]PATRICK H.WOOD, STEPHEN G.KOCHAN 著

刘昊宇　王　哲
赵爱利　汤连生　译

刘尊全　审校

责任编辑　徐云鹏

*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行　各地新华书店经售

北京顺义李史山胶印厂制版印刷

开本：787×1092 毫米 1/16 印张：11.375 字数：366 千字

1992 年 2 月第 1 版　1993 年 2 月第 1 次印刷

印数：8000 册　　定价：8.20 元

ISBN7-5053-1593-5/TP·322

译者的话

《UNIX 系统安全实用技术指南》（原名 UNIX SYSTEM SECURITY）一书是根据美国海登（Hayden）图书公司 1987 年第四版译出的。该书是 UNIX 系统文库丛书之一，PATRICK H.WOOD 和 STEPHEN G.KOCHAN 是这套丛书的著名作者。本书是 UNIX 系统安全的一本专著，从用户、程序员、管理员和网络的各个方面详细讨论了 UNIX 系统的安全问题。在论述有关内容时，介绍了与安全问题有关的 UNIX 命令、例行程序、函数等内容，使读者对 UNIX 系统的环境、安全机制和使用功能有一个全面的了解和认识，具有重要的实用价值。

随着 UNIX 系统日趋广泛的应用和普及，广大计算机专业人员和用户对了解 UNIX 系统安全技术具有紧迫感。本书内容丰富，取材新颖，许多内容（诸如 SUID／SGID 权限、DES 加密算法、安全管理、安全审计、受限环境、计算机犯罪的常用手段、失密的处理、UUCP 的安全等）都反映了当今国际上计算机安全的有关方法和技术。在一定意义上，本书对计算机安全的研究具有指导意义。

本书的第一、二、三章及附录由刘昊宇翻译，第四章由汤连生翻译，第五章由王哲翻译，第六章由赵爱利翻译，全书由刘尊全审校。

限于时间与译校者的水平，缺点错误在所难免，敬请读者批评指正。

译者

1990 年 12 月

目　录

第一章　引言

众所周知，计算机改变了我们工作的方式。当今，在一些典型的公司里，从收发室、打字间，一直到经理办公室，计算机都被广泛地使用着。其结果是那些以前可能被锁在办公室里的各种信息，现在都存储在计算机里。这些信息包括人事档案，新产品和新发明的备忘录，以及记录过去销售情况和将来销售前景的数据等等。

大多数的大中型计算机及目前的小型计算机是为用户共享的。这些计算机在多用户的环境下工作，允许一个以上的用户同时使用计算机。一些大型的计算机系统允许数百名用户同时使用计算机。今天，许多计算机还经常通过数据网络互相进行访问，以文件形式存储在计算机里的数据潜在地可以为使用该计算机的任何用户访问。由于这种开放性，现在计算机安全毫无疑问地应该受到足够的重视。

UNIX 操作系统是贝尔实验室于六十年代末研制的。近几年来，它以令人吃惊的速度得到广泛应用，看起来它终将成为工业标准操作系统。UNIX 系统提供了一个“友好的(friendly)”环境，鼓励各种系统的用户共享他们的文件。这并不意味着过去所强调的安全问题已被置之度外，实际上，UNIX 系统（如同任何其它的多用户系统一样）设计中一个重要方面就是使用户之间在逻辑上保持独立。也就是说，一个用户所运行的进程不能干扰其它用户的进程。同样，一个用户所拥有的文件也会同其它用户的文件很容易地区分开来。

与大多数人的看法相反，UNIX 系统被设计成非常安全的系统。事实上，UNIX 系统安全性最好的特征之一就是让系统管理员有权决定系统的安全程度。当然，一个不安全的系统必须同其它安全的系统保持隔离。正如谚语所说：“堡垒最易从内部攻破”。

作为一个系统管理员，最难作出的一个选择是既要限制用户在完全安全的计算机环境当中，又要允许（甚至鼓励）用户与其它用户共享数据。如果安全性被限制得过于严格，会产生不利于用户的创造性和生产率的影响。例如，AT&T 贝尔实验室（Bell Labs）很多程序设计人员的工作一直依赖于对 UNIX 操作系统源代码的自由操作。这使得他们可以很容易地在现有程序的基础上从事开发工作，加以改进，发现其中的错误，并在某些情况下修改这些错误。如果限制对源代码进行操作，可能会使许多程序员在他们需要写一个程序时感到无从下手，或者由于太费时间，人们干脆加以放弃。此外，许多程序的设计思想正是当程序设计人员在浏览现存程序并看到具体问题怎样得以解决时实现的。当然，在另一个方面，允许自由操作源代码，显然使得某些人更容易窃取它了。

谈到计算机安全（Computer Security），一个关键词是安全审计（Security Audits）。安全审计是管理员在考虑如何建立并保持系统的安全，如何对用户进行安全教育，如何定期进行安全检查和如何建立安全标准并加以实施。安全审计也是用户在考虑以下情况时所必需的：如文件存取权限的作用并使自己的文件不被其他用户读出或销毁，如何通过网络传送文件并使得其他用户不能读出这些文件，如何编制安全的程序，如何选择好的口令并作好口令的保密，以及了解不要开着终端而无人看管。安全审计也是用户和系统管理员的

管理机构所必需的，应建立公司范围内的安全政策并让雇员花费一些时间来执行这些政策。

本书的目的是向 UNIX 系统的用户和管理员讲授这些安全意识。

首先我们让 AT&T 贝尔实验室的罗伯特·莫里斯（Robert Morris）先生对计算机安全问题作一个剖析开篇。第二章的内容就是莫里斯先生在美国众议院运输、航空和材料的小组会议上所作书面证词的全文。

第三章从用户的角度讨论 UNIX 系统的安全性。它包括对文件存取权限、口令、设置用户标识和组标识、文件加密、特洛伊木马（Trojan horses）及其它安全问题的详细讨论。这一章将教给你怎样使自己的登录（login）保密。

第四章是为程序员写的。它从用户和系统管理员的角度，讲述如何编写安全的 C 和 Shell 程序。

第五章的内容是管理员所需的安全技术。读者将学会如何实现系统的安全，以及如何通过定期的安全检查来保持它。此外还介绍了如何建立一个严格控制的环境来防范潜在的破坏者。鉴于越来越多的人购买了供办公室使用的小型 UNIX 系统并成为它的管理员，所以这一章以一个关于小型系统安全的讨论结束。

最后一章详述了网络安全。特别提到 UUCP 数据网络，它是世界上最大的计算机网络，读者将学到老的 UUCP 和新的 HONEYDANBER UUCP 网络的安全技术。

由于我们把本书设计成为 UNIX 系统安全的实用指南，所以本书的附录提供了八个有关安全技术的源程序，其中包括了口令管理、安全审计、文件存取权限检查、终端安全、DES 数据加密算法的使用及受限环境的建立。我们希望读完本书时，读者将乐于把这些程序输入自己的系统并使用它们。这些程序全部在 UNIX 系统 V，系统 V Release 2，UTS 和 Xenix3.0 上验证过。

除非特别说明，本书所有的讨论只涉及系统 V。

第二章　安全问题剖析

罗伯特·莫里斯先生在美国众议院科学技术会议运输、航空、材料专业委员会上的证词

(1983 年 10 月 24 日)

我是罗伯特·莫里斯（Robert Morris）[1]。我是 AT&T 贝尔实验室的计算机科学家，在我的职业生涯中有二十年时间主要从事计算机方面的研究工作——包括计算机设计、计算机安全、密码学及其相关领域。

我很高兴能有此机会来介绍我们在贝尔实验室工作范围内所涉及的计算机安全问题，并且希望我的观点能有助于你们对这个复杂问题的认识和了解。

计算机安全是一个当前人们普遍关注的课题，并且它的重要性对于美国社会来说日益增强。计算机犯罪通过实际案件的新闻报道和类似电影中杜撰的战争游戏等（War Games）虚构事件，逐渐为公众所熟悉。随着时间的流逝，实际和虚构的界限变得模糊了。现在为使公众澄清认识，我要在专业委员会议上专门阐述计算机安全的复杂性及其有关的差异问题。

我作为 AT&T 贝尔实验室的计算机科学家，将首先向大家提供一个大型的计算机环境（Computer Environment）的剖析。其次，我将给出计算机安全关键问题的本质和范围的定义。接下去，我要阐述我们应该重视计算机安全问题并且讨论解决这类问题的通用方法。最后，我将就如何解决安全问题，进行一些广泛的探讨。

我们在贝尔实验室专门从事计算机和计算机软件方面的研究，现在半数以上的雇员从事软件的开发和支持工作，而在 1974 年这方面人员只占总数的百分之十五左右。今天，我们已经拥有 1800 台主计算机和比技术人员数量还多的计算机终端。我们支持贝尔计算机系统上投入运行的大约 3 亿 5 千万条代码，它使得我们成为世界上最大的软件企业之一。

我们的计算机环境包括集中式管理的各地区的计算机中心。另外，我们公司的许多部门都有计算机，通常为小型计算机、专业用工作站（Professional workstations）和微型计算机，提供给公司的技术人员使用。

这些计算机采用多种方式相联。例如，我们有一种使用高速的计算机通讯线路的局部网络。此外，技术人员可以使用其它类型的网络与工作有关的计算机进行通讯。我们在相当多的地区使用遍及全国范围的远程通讯网络。此外，许多雇员可以通过安置在家里的终端访问公司的计算机。

我们重视计算机安全的一个重要原因是，这是可能发生的，至少在概念上外国机构、

---

①他是 1988 年 11 月 2 日美国发生的六千多台计算机遭病毒感染的肇事者 R.T.Morris 的父亲。——译者注

竞争者、黑客（hackers）——虚构中的任何人，能够从与远程通讯网络相联的计算机系统中窃取信息。

如同我们学会了如何高效率地研制大型软件系统和如何保证软件产品的质量一样，我们也建立了有效的软件管理方法。这些方法包括安全性方法的设计，诸如存取控制用于防范或检查非授权访问企图的审计能力。这些方法是可行的，并且已经应用于贝尔实验室的所有部门。我们在公司的范围内有一个软件方面的委员会，以便协调并使我们的知识和处理过程标准化。

从总体上讲，对计算机安全的威胁来自"简单电子入侵"和其它的形式，这些形式包括授权人的委托失误、物理的入侵、专家机构进行的电子侦察，以及对通讯线路的窃听。这里我主要侧重简单电子入侵，这是因为从各种新闻报道和我们所进行的实验表明，它可能是当今对计算机安全的最普遍的威胁。我之所以这样认为，并不是想给出其它形式对计算机安全的威胁不重要的印象，或许其中最令人担忧的是授权人的委托失误，不幸的是从本质上讲计算机安全技术对它起不了多大作用。

计算机安全包括物理安全和逻辑安全两个方面。前者可以由锁好门、设置警卫及采取类似的预防手段来实现；后者由口令、文件授权、审计及类似的措施来实施。这里我将侧重强调逻辑安全性——包括计算机、网络及其相关的软件、用户和管理人员。

在贝尔实验室，我们在计算机安全上的目标是在安全性与通讯的方便之间作出选择。毫无疑问，安全性越强，对通讯的限制和难度也就越大。鉴于技术革新是我们整个企业目标的核心，因此通讯是至关重要的，其中包括组织之间的界限、技术条目与物理装置之间的通讯。

不言而喻，计算机安全的重要性是显然的，但有时却被人们忽视。如同银行的保管库比木制的门房更为安全一样，计算机安全问题应该考虑到信息本身的价值，应该有一个多层次的安全系统——从最小、中等至最大范围，以便提供文件保护的必要层次。在贝尔实验室里敏感的人事信息（例如工资表的数据内容）是同其它类型的信息严格加以区分的，而且对访问敏感数据（Sensitive data）要加以严格控制。

关于计算机安全的另一个关键问题，涉及我们试图保护的对象——电子信息的本质。一些人可以不必转移物理设备，而从计算机文件中窃取信息，这一个信息特征使得确定窃取信息的工作复杂化，实际问题不断地在变化，从而也使得相应的道德问题趋于复杂化。

最后，我要指出一个公司的最高行政部门必须重视计算机安全问题，否则任何一个实际问题的发生都使系统面临真正的威胁。前面已经提到，在贝尔实验室的软件委员会召集成立了一个计算机安全特别工作组，以便处理有关安全方面的问题并提出改进安全工作的建议。这个特别工作组要向公司高层管理部门报告所发现的情况，以及所采取的相应措施。此外，我们在计算机中心设立了常设的安全委员会，还在保护资产机构里配备了计算机安全专家。

综上所述，我还要补充说明一点，在贝尔实验室安全性相对较弱的系统上电子入侵并未被排除掉。我们的保护资产专家对于跟踪电子入侵者富有经验，并且当工作需要的时候可以得到法律部门——强制机关、地方和联邦法院的支持。鉴于以上的情况及我将叙述的其它步骤，现在我们比过去能更早更快地发现电子入侵者。

尽管如此，我并不想暗示电子入侵问题已经彻底解决了，无论是对于我们或是我所知

道的任何一家大型的高技术公司，在这方面的电子入侵同冒充顾客入店行窃相类似，好的管理方法和严密的安全措施可以控制并且可以使问题减少到最小，但是并不能根除这方面的问题。

贝尔实验室对计算机安全的基本考虑是：我们要保护有价值的信息不被窃取、更改或毁坏，当它们存储在计算机里或通过数据线路进行传输时，我们要注意防止对机时和资源的非授权使用。为此，我们要求计算机用户和管理人员保持高度的安全意识（security awareness）。综上所述，我们对整个计算机环境包括硬件、软件及有关人员等要有一致的和集中的行政控制手段。

从一开始，计算机安全最重要和明显的事情是所涉及的人员：用户和管理人员，也包括他们的主管人。然而人们对安全的最大威胁却往往被忽视了，例如：在登录后使用一台计算机，然后在终端无人看管的情况下离开了；进行计算机存取时与他人共用口令；将敏感信息存储到不适宜的计算机文件中。

为了总结我们研究计算机安全问题的一些方法，让我们来看下面的检查表（check list），我们将它推荐给主管人，以便检验他们企业里计算机的安全程度：

- 你知道谁对你的计算机拥有访问权吗？不要与他人共用口令，即使是你所信赖的人。如果你组里的人员需访问其他雇员的文件，他们应有自己的口令。
- 你的系统能否拒绝远程计算机的非授权请求？如果不能，它应该通过对计算机重新设置许可来加以纠正。
- 你是否有一位系统管理员？在他的工作范畴中是否具有监视并改进安全方面的措施？最安全的系统是那些具有强有力的管理措施的系统。
- 你是否将诸如公司计划或人员评价等秘密信息（private information）存入你的计算机文件？要设想最坏的情况，甚至偶然的浏览者也可能读出你输入的信息并把敏感数据保存到另外的地方。
- 你的组员是否了解计算机安全的重要性？要确保组内的雇员了解计算机安全的重要性，并且他们能够采取必要的措施或手段。

让我们更进一步考察这些观点的有关内容。例如口令，除了“一个人一个口令”（one person，one password）的预防措施外，我们可以采用更加复杂的口令以便提高其安全性。计算机用户经常用他们的姓，或者是配偶或宠物的名字，或是生日来作为口令。不幸得很，在破译口令中一台机器能迅速地扫描一个由姓或英语中 20，000 个最常用单词组成的阵表。一种可供选择的口令的更加复杂的方案是由六个字符组成的口令，字符可以是数字或字母。这样的口令将是极其难以破译的。当然，口令必须不是任何时候都用的，即在用过一定次数以后，口令必须被更改，还要根据系统安全的级别来确定识别的强度。

这里还要强调指出责任的重要性，确定计算机系统所涉及的人——用户、管理员和主管人员。例如，所有的计算机使用都要有主管人的授权，以便决定管理的责任，掌握由谁和出于何种目的使用计算机。最后，每台计算机都应该有授权用户表，此外，即使是普通的使用还应有一个公司可以访问的计算机日录，其中包括与特殊计算机、电话号码、系统管理员和对有关部门的识别。

在这个计算机安全检查表中，最后强调的一点是专用软件包的使用，通过限制对独立用户文件的一般性访问以提高安全程度。我们可以通过限制特殊文件及其建立的软件访

问，来保护计算机文件中的信息，直到准许其他人的访问。另一种限制访问的方法是使用拦截访问的硬件，要求核实口令，并且可以收回日录的授权用户成员表。

除此之外，其它的安全软件包可以跟踪非授权的可疑访问。例如，重复尝试登录或对他人文件提出请求。为此，我们可以采取限制企图登录的次数或设置登录时间的限制，当超过时间限制后将会被自动地切断。

现在拥有的技术可以提供一个高水平的计算机安全手段。例如，我完全相信可以在军事安全（Military security）标准下来控制我们的计算机运行。我要补充一点，用于控制加密信息破译的计算机安全费用高得惊人。因此，为了保持一个高水平的计算机安全环境，通常对计算机采用隔离法和设置物理进入系统的障碍。

随着计算机安全技术的进展，可以有助于减少这种隔离措施。换句话说，可以通过建立全面的安全控制，对合法用户（legitimate users）提供更多的透明度。实际上，我是说——正如我们致力于开发的计算机拥有更多的“用户友好”（user friendly）界面——我们在发展或提高计算机系统安全性能时，也应该把用户的需要牢记在心头。

对通过通讯线路入侵计算机的防范和检测技术与系统识别电话号码一样简单，这种技术一方面可以允许授权电话号码表的安全性检查，另一方面也能够提供非授权访问企图的跟踪记录。这种技术日前已经应用于计算机化的商业通讯系统中，通常是用于这些公司提供系统服务的线路，这种技术正在向全国范围的远程通讯线路方面推广。

此外，在我们称做“信用卡终端”（Credit-card terminals）上同样存在使用这种技术的可能性，防止篡改意味着要比日前的口令提供更高水平的识别技术，设置识别硬件（identification hardware）卡也将是有益的。这是一个可以产生信号的简单硬件，它可以防止软件的复制，这样就能够增强网络地址（network addresses）的安全能力。

日前可以采用的计算机安全的最基本方法是加强人们的安全意识，我们需要全国性的努力来提高人们的计算机安全意识——法律和道德的教育。我们希望各级的教育部门能够提供更大的支持。作为开端，我们要去掉计算机黑客的假象，他们不是什么绿林好汉，而是电子偷视者（Peeping Toms）、非法入侵者和窃贼。我们有必要来澄清涉及破坏计算机安全的一些模糊问题，从而使得没有人再以声称“不知道做了错事”来为这类行为进行辩解。

我们不应该忘记计算机安全方面的受害者。我们也必须在那些掌握电子信息的人们中间增强安全意识。同电子入侵者和窃贼一样，那些忘记了“锁上他们的门”（lock their doors）的人也应该承担他们的信息遭到破坏或丢失的责任。

第三章　用户安全技术

在这一章里我们从用户的角度来讨论 UNIX 的安全技术。我们将涉及口令、文件保护、目录保护、UNIX 系统关于用户程序的一些特点以及使用 crypt 命令加密。我们也将讨论一些重要的安全技巧，它们将对你的登录安全有帮助。

3.1　口令安全

UNIX 系统中有一个叫 / etc / passwd 的文件，它包含了系统所需知道的每个用户的信息，包括口令。不管你相信与否，这个文件在系统上能被任何人打印出来。为什么系统这么自信？因为这些口令是编码方案加密（以后会涉及到）的，它使得要想破译某个人的口令是非常困难的。一段从 / etc / passwd 文件典型的摘录如下：

```
S cat / etc / passwd
root: xyDfccTrt180x, M.y8:0:0: admin: / : / bin / sh
console: lolndT0cc0Mzp, M.y8:1:1:admin: / : / bin / sh
pat: XmotTvoyUmjis: 127: 10000:p wood: / usr / pat: / bin / sh
steve: J9cxpd97Ftlbn, M.z8:201:10000:s kochan: / usr / steve: / bin / sh
restrict: PomJk109JkY41,. / :116:116:: / usr / restrict: / bin / rsh
S
```

本书例子中的黑体字将代表用户的输入，登录名首先被列出来，接着是冒号(:)，接着是加密的口令，接着是另一个冒号，两个数字，接着是更多的信息（将在第五章中讨论）。那两个数字分别是用户标识和组标识。它们被 UNIX 系统用作对用户和组进行唯一的鉴别和用作确定是否一个进程能够访问一个文件（以后会涉及到）。

无论你什么时候登录，你在终端上键入的口令被加密并且同 / etc / passwd 中你的登录的加密口令做一核对。如果它们完全一致，你将被允许使用系统；如果它们不一致，将给出 Login incorrect 的信息，你必须重新键入。

如果想要改变你的口令，你不能直接修改 / etc / passwd，不允许那样做。如果能直接修改，迟早有人会进入并改变所有的口令；这样就没有人能够登录了。作为替代，你可以使用 passwd 命令。你需要做的只是键入 passwd，它将提示你下一步做什么：

```
S passwd
Changing password for pat
Old password: wizardl                    Not printed
New password: wom2 bat                   Not printed
Re-enter new password: wom2 bat          Not printed
S
```

在允许你改变口令之前，passwd 命令要求你键入原来的口令，这样做只不过是确信真的是你，而不是别人在当你不在时使用了你的终端。如果你在键入原来口令时出错，系统回答 Sorry.，意味着未做出任何修改并要求你重新键入口令。如果原来的口令正确，passwd 命令会让你键入新口令。由于这些口令不在屏幕上显示，命令让你第二次键入新口令以免你不小心出现错误。如果两次输入不一致，passwd 命令将再要求你键入新口令两次：

```
S passwd
Old password: wom2bat                    Not printed
New password: wizzardl                   Not printed
Re-enter new password: wizrdl            Not printed
They don't match; try again.
New password: wizzardl                   Not printed
Re-enter new password: wizzardl          Not printed
S
```

3.2 选择好的口令

尽管许多 UNIX 系统不对口令进行限制（有些根本不要求），但如果你想使登录安全，你应该使用不那么显而易见的口令。名和姓、姓名起首的字母、出生日期以及类似的都是不好的口令。把登录名倒着拼写也是不好的。给几个周末的上机时间和一部联机字典，由普通英语词汇产生的口令就可能被破解，即使你在词的末尾加上一位数字。不要用房间里看得见的东西上的字作口令（如一个由航海广告产生的口令——hobiel），因为任何到过办公室的人可能看到同样的标题。

“好的”口令至少要六个字母长，不是基于个人信息，并且中间有非字母的字符（特别是控制字符）：4score, adv8ance, my_name, sistem (sic), bonljour, 及 alb2c3 要被破译是足够困难了，但并不是完全不可能。

更好一些的口令是 dg7m33ex, mintlpen, 及 luv2run. 这些几乎是不可能破译的。不幸的是，第一个几乎不可能被记住。如果一个口令太难以理解以致于你不得不把它写下来，它就不是一个好口令——把口令写下来不是一个好的保密习惯。（你同样不能将登录名或口令存储于终端的功能键或智能调制解调器的存储区中）。选择口令的一个好办法是挑两个不相干的单词，例如，foot 和 house，中间用一个数或控制字符将它们联起来，再将字符串截短到八个字符。

你不应该在不同的机器上使用同样的口令作为登录名，如果其中的一个被破解了，将危及所有的登录名。如果你使用保密程度不同的机器，即使一个坏家伙在一保密程度差的系统上发现你的口令，他也不能用它去接近其它的、更为安全的系统，这一点就特别重要。

你应该定期更换口令，这样即使有人已经发现了它，任何非特许使用也会被消除。更换口令的周期依赖于登录名要保密到什么程度；不管怎样，你应该至少六个月换一次。在

第五章里讨论如何使 UNIX 系统要求用户定期更换他们的口令。这其中的逻辑是一个被窃取的口令只能被用到该口令的使用期满。如果你的系统要求你定期更换口令，就不要再用原来的；挑一个你从未使用过的口令。通常，人们总是交替使用两个口令，这意味着如果旧的口令被窃取了，当你又轮到它时它可能被再次非法使用。

当你登录时，运行 su 命令（以后会涉及到）及更换口令时，你都需要键入口令。这几次口令不显示出来是为了防止被偷看。尽管如此，一些眼睛锐利、记忆力好的人还是能通过观察你手指在键盘上击键来得到你的口令。所以当键入口令时，你必须确信周围没有人。

3.3 文件存取许可

文件存取许可的作用是确定谁能够访问一个文件以及访问该文件时能够做些什么。例如，你可以要求一个包含一些敏感信息的文件不能被其他用户读出，但是你也可以要求另一个文件可被任何人阅读。UNIX 系统允许你从你的需要改变一个文件的存取许可。这些许可确定谁能读一个文件、谁能对它进行写操作以及谁能执行它，如果它正好是一个程序。

文件属性

具有-l 选择项的 ls 命令打印出一个文件的详细目录：

```
S ls -l zombie
-rwxrwxrwx 1 pat CS440 70 Jul 28 21:12 zombie
S
```

现在跳过文件存取许可（-rwxrwxrwx）和紧接着的计数（1），你可以看见与文件有关的拥有者名字（pat）和组名（CS440）。每个文件都有一个与之相关的拥有者或组。文件的拥有者是一个用户，通常是创建文件的人；组是几个用户被逻辑地组合在一起并命名的一种标识。例如，几个进行同一项目的人通常被分为一组，所以在限制外人（非本组用户）的同时，他们可以自由存取组内其他人的文件。每个用户是在一个组里，即使里面只有他一个人。一个用户属于一个以上的组也是可能的（例如；假设他工作于不同的项目）。

回到-rwxrwxrwx 上来，你会看到开始的-。它说明这是一个正常的文件并且不是目录。在-后是 rwxrwxrwx，它被称作文件的模式或存取许可。它告诉你能对 zombie 做些什么。你会注意到这里模式 rwx 重复了三次。这三个模式中的每一个告诉你用户处理文件的一种特殊形式。第一个 rwx 告诉你 zombie 的拥有者（在这种情况是用户 pat）能够读出（r）、写入（w）及执行（x）文件。第二个 rwx 告诉你 CS440 组的任一成员也能对该文件进行上述操作。第三个 rwx 说任何用户可以进行上述操作（图 3-1）。如果其中之一被拒绝，-显示出来代替了与之相当的字母。例如，rw-表示可以读和写，但不能执行；r-x 表示可以读和执行，但不能写。

有时你可能看见除了 x 或-以外的字符出现在存取许可的执行位置。其它可能出现的字母是 s、S、t 和 T。s 和 S 出现在用户模式和组模式中，与一些特殊许可有关，我们将

在本章的后面去讨论它们；t 和 T 出现在其它模式中，与和安全性无关的“sticky bit”有关。你所需知道的是小写字母（x、s 或 t）说明允许执行，一个减号或大写字母（-、S 或 T）说明不允许执行。

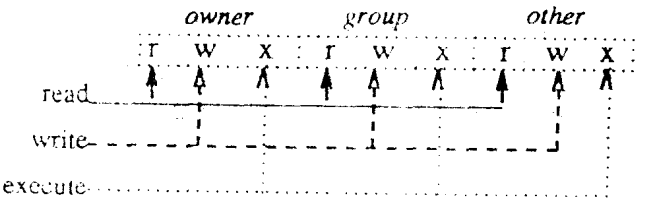

图 3-1　文件模式

改变文件存取许可

现在你得到了一种对于三种模式的用户：拥有者、组和其他用户控制你的文件的可访问性的方法。我们假设你是 pat，文件 zombie 的拥有者，并且你不想要除了 pat 的任何用户写入（从而改变或破坏）zombie。另一方面，由于你感到 zombie 是一个有用的程序，你想让其他用户都能检验并执行它。所以你要把 zombie 的新模式变成 rwxr-xr-x，即允许组内用户和其他用户读和执行此文件，但不能篡改它。

为了更改 zombie 的模式，你必须使用 chmod 命令建立新模式并将文件名作为参量。新模式不是对 chmod 指定 rwxr-xr-x，而是一个三位的八进制数，它是由被指定许可的数值相加计算得出的（图 3-2）。

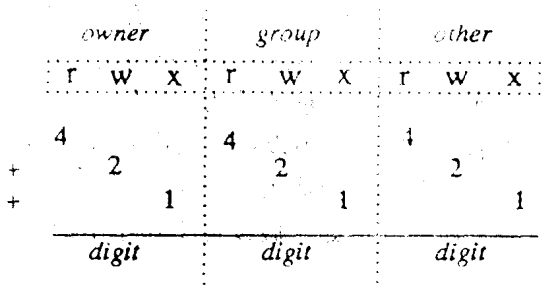

图 3-2　计算新的存取许可

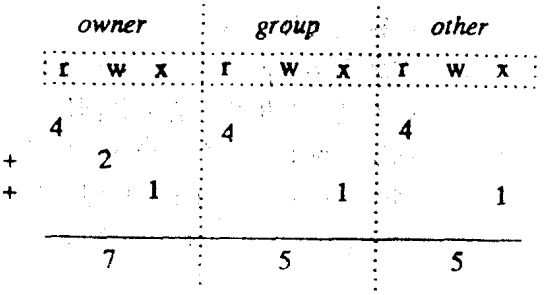

图 3-3　计算 rwxr-xr-x

在这种情况下，新模式为 755，7 代表拥有者许可 rwx，第一个 5 代表组许可 r-x，第二个 5 代表其他用户许可 r-x（图 3-3）。

为了把 zombie 的模式变为 rwxr-xr-x，你对 chmod 命令给出了三位数，后面是要改变的文件：

```
S chmod 755 zombie
S
```

如果你看 zombie 现在的模式，你会发现它的确改变了。

```
S ls -l zombie
```

```
-rwxr- xr- x 1 pat CS440 70 Jul 28 21:12 zombie
S
```

下面将看到使用合成模式的 chmod 的其它一些例子。注意只有拥有者或上级用户能改变一特殊文件的模式；因此，如果其他用户不能访问该文件，那么他们就不能使用 chmod 来获得访问权。

```
S chmod 700 zombie
S ls -l zombie
-rwx------ 1 pat CS440 70 Jul 28 21:12 zombie
S
```

现在只有 pat 被允许对 zombie 做任一种访问。

```
S chmod 711 zombie
S ls -l zombie
-rwx--x--x 1 pat CS440 70 Jul 28 21:12 zombie
S
```

现在组内成员和其他用户被允许执行 zombie，但不能检查或改写它。

```
S chmod 771 zombie
S ls -l zombie
-rwxrwx--x 1 pat CS440 70 Jul 28 21:12 zombie
S
```

现在 CS440 组内用户被允许拥有同 pat 一样的特权，但其他用户还是只能执行 zombie。

存取许可被使用在防止意外改写或删除一个重要文件。你所需做的是把模式改为 r-xr-xr-x，这样即使你是拥有者也不能写入文件。你在更改文件之前不得不改变模式以便允许你自己写入。

```
S chmod 555 zombie
S ls -l zombie
-r-xr-xr-x 1 pat CS440 70 Jul 28 21:12 zombie
S echo hi there > zombie
sh: zombie: cannot create
S
```

这里 shell 不能把 echo 命令的输出重新定向到文件 zombie，因为写入许可未被设置。

即使 rm 也不会立即删除该文件，它会首先要求认可：

```
S ls -l zombie
-r-xr-xr-x 1 pat CS440 70 Jul 28 21:12 zombie
S rm zombie
zombie: 555 mode ?                    Do you really want to remove it?
```

如果当 rm 要求认可时你键入 y，文件将被删除，忽视了模式或拥有者。任何其它输入将使 rm 在不删除文件的情况下退出（你将在后面看到，删除文件的许可与其所在目录