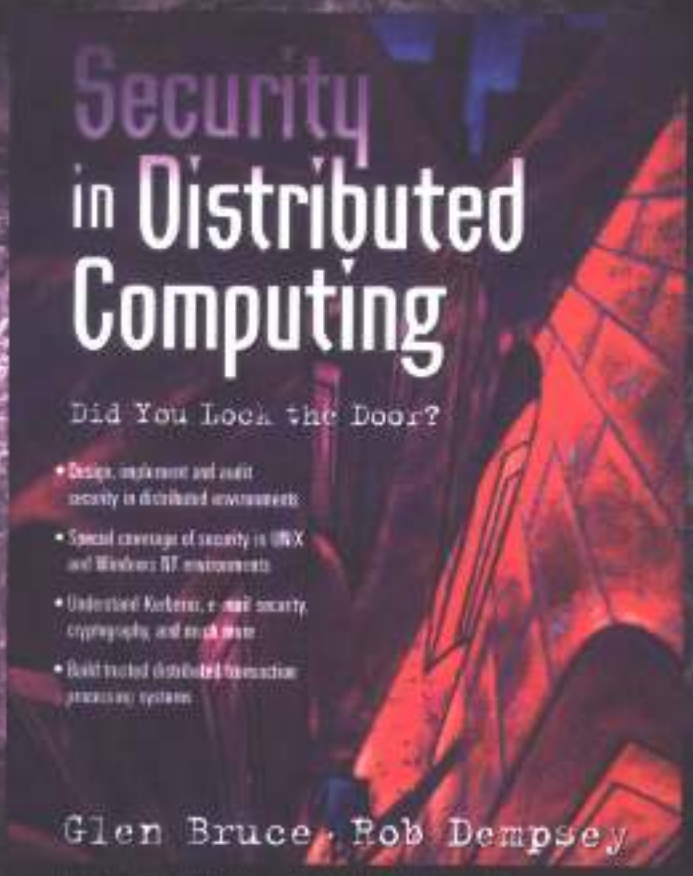
计 算 机 科 学 丛 书

分布式计算的安全原理

(美) Glen Bruce Rob Dempsey 著 李如韵 刚冬梅 等译



Security in Distributed Computing

机械工业出版社
China Machine Press
PH PTR

从核心意义上讲，计算安全不仅仅是一个技术问题，它是一个基本的业务挑战。管理人员有大量可选的安全方案，但是很少有关于如何进行选择的实际指导。尤其在今天，分布式的、多厂商的、因特网的环境变化以往拥有更多的不安全因素。

本书为分布式系统的管理人员解决分布式计算安全问题提供了一个完整的、通用的框架。主要内容如下：

- 开发更加安全的分布式系统体系结构和方法
- 构建可信的、基于开放式系统的分布式事务处理系统
- 评估代价与风险：什么值得去保护，价格如何？
- 考虑人和组织因素，从而做到在提高安全性的同时把对人和业务的影响降至最低

ISBN 7-111-10827-2
9 787111 108276

华章图书

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037
购书热线：(010)68995259，8006100280（北京地区）
总编信箱：chiefeditor@hzbook.com

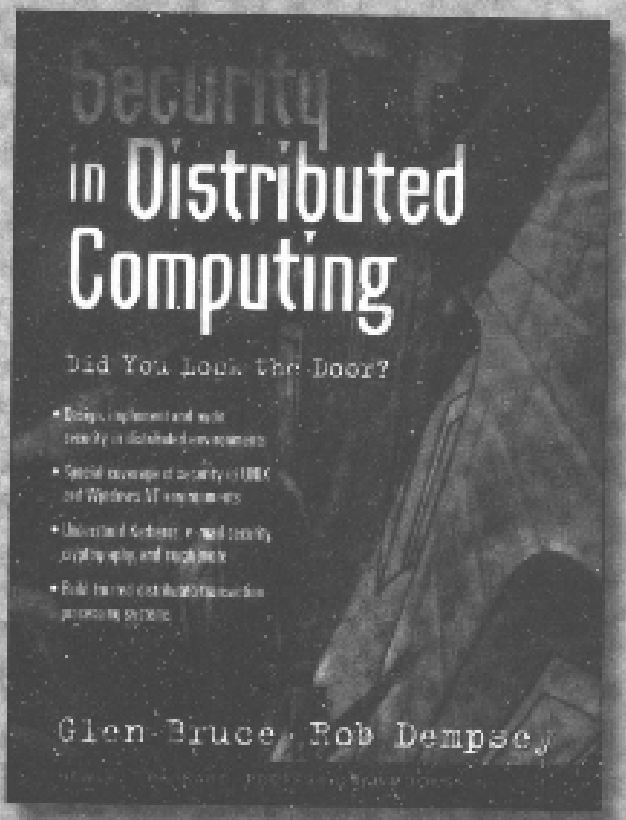
ISBN 7-111-10827-2/TP · 2575
定价：35.00 元

注意防止涂改

计 算 机 科 学 丛 书

分布式计算的安全原理

(美) Glen Bruce Rob Dempsey 著 李如钧 刚冬梅 等译



Security in Distributed Computing

机械工业出版社
China Machine Press

本书为分布式系统的管理人员解决分布式计算安全问题提供了一个完整的，合理的框架。全书包括4个部分，共25个章节，主要讲述了如何开发更加安全的分布式系统体系结构和方法；构建可信的、基于开放式系统的分布式事务处理系统；评估成本与风险；考虑人和组织因素，从而做到在提高安全性的同时把对人和过程的影响降至最低。本书探讨了分布式系统中的很多关键风险领域，其中包括网络、操作系统、应用程序、中间件及因特网。并为如何设计和实现安全策略提供了有价值、广泛的建议。

本书适合于广大的计算机系统安全保障人员阅读，可以作为大专院校的计算机教材或辅助教材。

Glen Bruce & Rob Dempsey: Security in Distributed Computing.
Authorized translation from the English language edition published by Prentice Hall PTR.
Copyright © 1997 Hewlett-Packard Company.
All rights reserved.
Chinese simplified language edition published by China Machine Press.
Copyright © 2002 by China Machine Press.

本书中文简体字版由美国Prentice Hall公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。
版权所有，侵权必究。

本书版权登记号：图字：01-2001-4767

图书在版编目(CIP)数据

分布式计算的安全原理/（美）布鲁斯（Bruce, G.），（美）邓普赛（Dempsey, R.）著；李知韵等译. -北京：机械工业出版社，2002.9
（计算机科学丛书）
书名原文：Security In Distributed Computing
ISBN 7-111-10827-2
Ⅰ.分… Ⅱ.①布… ②邓… ③李… Ⅲ.分布式计算机系统-安全技术 Ⅳ.TP338.8
中国版本图书馆CIP数据核字（2002）第062697号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：张金梅
北京第二外国语学院印刷厂印刷·新华书店北京发行所发行
2002年9月第1版第1次印刷
787mm×1092mm 1/16·18印张
印数：0 001-4 000册
定价：35.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译 者 序

本书是一本关于分布式计算安全原理的权威书籍。作为惠普公司的专业顾问，两位作者对分布式系统安全所涉及到的概念、问题、技术、标准、方案以及发展做了详尽而权威的探讨。全书的组织符合“提出问题、分析问题、解决问题”的一般思路。本书首先是一本安全技术书籍。全书所涉及到技术全都是与分布式计算安全问题密切相关的，这包括操作系统、网络、应用程序、数据库、中间件、联机事务处理等领域；另外作为纵向对比，本书始终关注分布式环境与大型机主机环境之间的对比，从而能够让读者明白在分布式环境的安全问题中，困难在那里，以及应该怎么解决。这对于不同领域的技术人员来说是非常重要的知识。

书中提到，安全问题并不仅仅是一个技术问题。为了向读者灌输解决安全问题的正确思路，本书始终在强调安全问题超越技术问题的复杂性。事实上，目前在国内，安全事故的发生和安全技术的缺乏通常都是因为人们安全意识的缺乏所致。本书的重点不仅仅放在技术上，并且也放在永远存在的人为和组织因素上。技术会改变，但是人们对安全问题的重视、管理部门对解决安全问题的承诺、员工需要具有的安全意识和责任意识，却一直是真正保证公司计算安全的决定性因素。因此，本书在安全问题中关于人和组织的解决方案提出了思路和步骤，它能够使管理人员掌握如何开发安全政策和策略，如何制定安全基础，如何保持安全、审计、技术以及用户部门之间的良好关系。译者相信，这些知识要比确定的技术知识更为重要，它们都是作者在多年经验的基础上累积的实用实践经验。这正是本书对IT专家和管理人员的意义所在。

本书可以说是一本迟到的好书。不可避免地，书中有些知识点在今天已经有了很大的发展，如Java。但是本书仍然具有权威性、实用性。希望读者能够在技术和管理两个方面去把握本书，并把书里的知识同具体的环境结合起来，从而建立真正安全的分布式计算环境。

本书由李如豹、刚冬梅组织翻译，朱冬东、杨启奕、吕俊辉等同志参与完成了本书的翻译、校对、录入等工作。由于译者水平有限，译文中不当之处在所难免，敬请读者批评指正。

译 者

2002年4月

序

“有些东西，只有当失去后才知道它的可贵。”你可能不止一次地思索过这句话，或者因别人的此种人生经历而记起这句话，才会认识到它的价值！

想像一下。假设你是一个进口旅行饰品的销售商，经过种种努力以后，你的事业终于走上了正轨，并且蒸蒸日上。你建立了一个尽管很小但非常忠诚的客户基础。就在这时候，你最大的客户却打来了一个电话，并生气地说要取消所有的订单。该客户说他刚收到了你的办事处发给他的一封信，信里说他的订单已经推迟了，因而不能在圣诞节前及时交货。你试图解释这不是真的，你从来没有给他发过这样的信。但事情已经发生。因此你迅速地检查你的计算机系统，结果发现你的客户信息和订单文件已经全被删除了。电话铃声又响了，这次是另外一个客户，但说的是一样的信。很快，你苦心建立的客户信任就一下子全没了。你的业务受到了致命的威胁。导致这一切的原因是什么呢？可能就是一个员工的误操作，或者商业间谍干的！无论哪种原因，结果都是一样的，即业务陷入混乱，你个人信誉受损。

今天的技术为我们提供了很多激动人心的机会。跟原来一样，我们需要个人信息和业务信息都应该是立即可访问的和可用的。只有认识到需要依靠这些为我们带来价值的信息，我们才能真正意识到它们的安全性是多么的重要！

本书打开了计算机安全领域的大门。对信息安全的日益增长的需要为我们提出了很多业务和技术问题，而Glen Bruce和Rob Dempsey所写的这本书可以很好地在这些方面教育读者。并且，他们还给读者提供了有关如何设计和实现安全策略的实用和广泛的建议。IT专家和业务专业人员面对使用信息技术的挑战，需要以一种安全简便的方式处理业务，对他们来说，本书是一本极好的读物。

阅读本书并不保证就可以实现一个安全的解决方案，但是本书为你提供了一个基础，在此基础上，你可以建立和改进自己的特殊策略。当然，你仍然可以什么也不做，但是你敢冒这个风险吗？

Glenn Osaka
惠普公司业务部，总经理

前　言

对于许多组织来说，保护企业计算资源不被滥用是一个非常复杂的问题。从最小的私人企业到世界上最大的金融公司，它们的计算系统都曾遭到过攻击或者发生过安全问题。

计算机安全事故一直是各种媒体争相报导的重点之一，它们提高了公众对计算安全问题的意识。然而，管理部门对该问题的认识以及解决问题的承诺，却还没有得到提高。现在，市场上出现了很多新的商业安全解决方案，这些方案脱胎于国防工业的先进技术。公司花费在这些技术上的经费大幅攀升。

大多数组织都已经认识到了安全问题的存在，并且已经采取了积极的措施以解决该问题。但是事故和攻击却仍旧不断地被报导，几乎是每天都有。不幸的是，计算机业界强烈地感到问题会变得越来越严重。那么，为什么计算安全问题没有得到解决呢？

答案在于如下事实，即计算安全是一个涉及到很多复杂方面的业务问题。它是不能只靠技术上的解决方案来解决的。事实上，不协调的购买和使用各种技术解决方案加重了问题的严重性。本书的目的就是让读者能够了解计算安全问题的所有方面。它将引导读者从各种问题和那些显得混乱的解决方案中寻求答案。

如果仔细分析，我们可以在计算机安全和家庭安全之间发现很多类似之处。家庭安全的基本预防措施就是锁门。虽然这一招并不能让房子绝对安全，但是它为窃贼带来困难。同家庭安全一样，为计算资产“锁上门”也是非常重要的。

我们需要仔细权衡安全问题的解决途径。如果后门不上锁的话，那么即使在前门上装一个世界上最好的锁，也是无济于事的——只锁一个门毫无意义！

我们还需要权衡购买和使用安全解决方案所付出的成本。如果家里的东西全加起来也只值5 000美元，那么谁也不会花上10 000美元来保护家的安全。特别是，如果邻居在过去5年内从没有遇到过任何失窃事情，那么就更没有必要大破费了。安全的成本必须要同预期的损失及相关的风险取得一致。

另外，我们应该把重点放在最有可能的安全问题上。小偷通常不会带着梯子去作案，因此我们应该把钱首先花在为低位置的窗户加置窗户网上。

不幸的是，绝对的安全是不能只靠钱来买的。如果人们没有意识到自己所承担的责任，那么再好的技术也没有用。如果在你不在家的时候你的孩子没锁门就出去玩了，那么即使你在门上装了世界上再坚固的锁，也毫无用处。安全不能看做是孤立于环境的。家庭的安全是同邻居的安全直接相关的。不能顾此失彼。

分布式客户机-服务器技术的出现极大地改变了许多组织中的计算环境。大型机环境中的复杂系统具有很高的操作可信性。大型机安全解决方案，例如IBM和Computer Associates开发的那些产品，允许用户实施强大的集中式控制。然而，分布式客户机-服务器环境的安全问题就复杂多了。与大型机不同，分布式环境中的控制和安全功能分布在几个平台上，并且通常不受任何单独处理器的控制。所以挑战就是保证分布式控制能够一起工作以完成一个共同目标。

我们将明确和说明计算机安全中的各种关键问题。如果要解决计算安全的总体业务问题，那么这些问题必须要先解决。这些关键问题包括需要对用户的安全认证和对用户操作的授权。网络使得全球计算业界可以使用以前没有过的方式来进行相互通信和合作，但是它也使得公司网络和计算系统可以为外部人员所访问。有效地使用技术解决计算安全是另一个关键问题。

说明计算安全所涉及到的技术是本书的一个关键重点。本书将介绍各种安全技术的细节。我们的目的并不是简单地讨论技术，而是让读者能够掌握如何使用技术来解决关键安全问题。

当一个认证过程通过网络通信时，如何信任该过程的完整性？这是关键问题的一个例子。大多数的网络流量，包括用户标识和认证密码在内，当前都是以明文形式在网络上传输的。通过监视网络流量，发现密码并使用它们来破坏安全性是可能的。

Kerberos模型（委托第三方认证）可用来解决认证过程安全性的维护问题。“Kerberos”这个名字来源于Cerberus——一只守卫地狱之门的三头狗，该模型为进行异构技术中的认证提供了一个方法。Kerberos假设网络是不可信的，并且其上的任何流量都有可能被截获。Kerberos的设计宗旨就是要解决这种威胁。我们将在介绍OSF/DCE（Open Software Foundation/Distributed Computing Environment，开放软件基金会的分布式计算环境，Kerberos的一种实现）时详细解释这种认证模型。了解了Kerberos的脆弱性和能力以后，读者可以判断OSF/DCE是如何有效地解决分布式计算安全问题的。

传统上，联机事务处理（Online transaction processing，OLTP）是为基于大型机的系统或者专门的事务处理系统而开发的。用户需要大型机具有连网能力、中心控制以及强大的处理能力，这些需求都是进行事务处理和维护共享数据库控制所必需的。OLTP系统处理事务，收集或者检查业务系统的信息，并把变化加进组织的共享数据库。把事务迁移到分布式服务器和桌面系统上会不可避免地带来安全问题，中心主机式OLTP系统上的保护措施和工具不能用在新的分布式OLTP系统上。若要有效地实现分布式OLTP系统，我们必须首先解决系统管理和安全挑战。

如果要在“开放式系统”平台上提供事务处理系统，那么我们面临两个需求，第一个是在非大型机平台上提供一个健壮的处理环境，同时还要保持同大型机一样的功能和能力；第二个是提供一种分布式处理能力，从而使事务可以在多个操作平台上执行功能并访问数据。Transarc公司的Encina技术是用来解决UNIX平台上的事务处理环境的。IBM的事务监视器CICS，已经移植到了IBM和惠普的UNIX环境上。这些事务监视器，当结合了OSF的DCE组件并启用Encina以后，可以提供分布式事务处理能力。通过DCE，我们可以使用这些技术来提供一个可信的事务环境。

我们还要探讨分布式系统上的集中式控制管理。使用高级网络和系统管理技术，可以确认已建立安全控制并仍旧在位。也可用网络警报来提供对非法行为的早期指示。我们将介绍动态警报技术的使用，并且为实现各种监测机制提供建议。

计算安全问题的解决不能只依靠技术。我们将花大量的篇幅来讨论人和组织所起的作用。我们要全面分析计算安全策略的形成、它所覆盖的区域，以及如何在策略与用户之间进行最好的通信。安全策略概括了组织在安全方面的决策，并且提供了一个基础——组织可以在该基础之上建立安全程序。为了认识这些重要行为的好处，管理部门对安全意识程序的承诺是必需的。

体系结构是一种描述各种组件功能的结构化方法。它以一种容易理解的方式描述了复杂

组件之间的关系。我们也可以把这种方法用在计算安全领域，从而更好地描述各种组件及其相互关系。安全体系结构包括了各种元素，这些元素能够保证信息的机密性，并且保证所有对计算资源的访问都是经过授权和认证的。体系结构的总体目标是可以信任分布式环境。我们需要能够信任所有的地方，或者具有补偿控制——用户访问各种系统而不是只把信任置于信息和工具所驻留的地方。安全体系结构是由很多构建部件组成的，这些组件一起定义了用于全面解决方案的框架。我们将探讨一个安全体系结构，并概述如何把体系结构用做企业安全解决方案的基础。

审计是计算安全中的另一个非技术领域。我们将介绍计算审计的目的、重要的原因以及如何为审计检查做最好的准备。我们还要探讨审计部门同公司其他部门之间的关系，并提供能够建立良好关系的建议。

使用一个结构化的方法对于解决计算安全问题来说是非常重要的。安全策略是组织采取的一系列具体步骤——通过这些步骤，组织可以把现有的安全水平从一个基础级提高到一个更安全的级别。策略方法会让一个组织通过一个已组织好的过程，评估当前计算环境所处的位置，定义组织希望处于的位置，并计划达到预期位置所需的步骤。使用一个已定义的方法，可以保证所有的窗户和门都关好了。当修建房子时，我们就应该考虑使用安全门窗。这种方法已经成功地用来解决了各种组织中的很多问题。

本书适用于任何对计算安全领域感兴趣的读者。通过本书，系统管理员和系统分析员能够了解一些核心技术的知识，例如Kerberos和公开/私有密钥加密。应用开发人员和体系结构设计员通过本书可以了解应该如何把多个安全组件集成进系统设计中。安全必须要被设计进而不是添加到系统中。

对于那些负责安全管理或者审计分布式计算应用的人来说，本书将提供对客户机-服务器计算中的核心安全问题所做的深入探讨。本书也将对那些关心计算安全问题的高级管理人员指引一个解决问题的方法。

计算安全不但是一个技术问题，而且还是一个业务问题。它是一个需要解决多方面问题的复杂问题。各种复杂的技术可用于解决不同安全问题。然而，组织必须以一种有计划的并且良好协调的方式来使用这些技术。另外，组织需要开发一个安全策略和体系结构。本书能够让你更加熟悉计算安全问题以及对它们的解决方案。我们希望，当你在通往分布式客户机-服务器计算之路上已经走了很远的时候，你不会再说“该不会没锁门吧！”

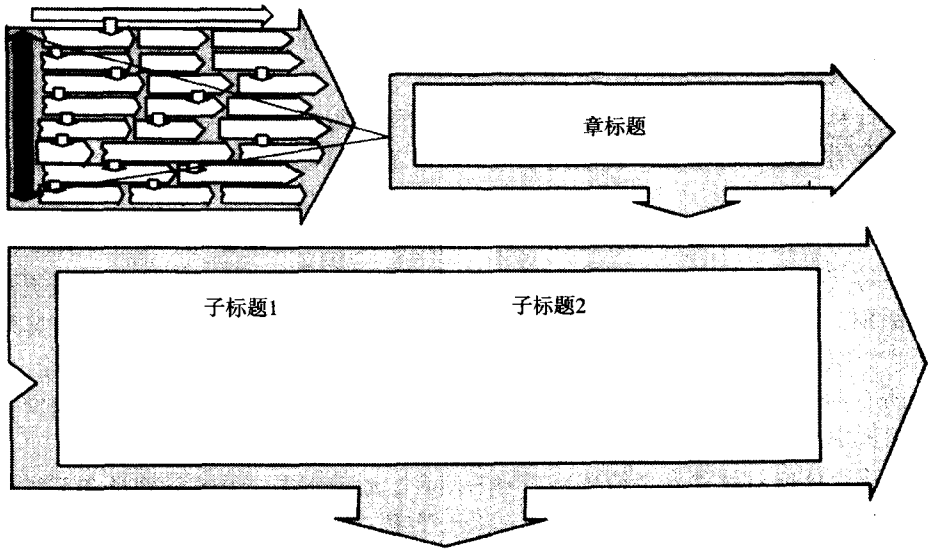
本书意在让读者能够明白在分布式计算中获得安全性会有哪些挑战，其目的是描述整体问题，并提供一些关于如何解决问题的思路。我们希望把重点放在一些能够让读者明白对付这些挑战需要进行工作的领域上，而不是提供一个有关计算安全的百科全书。因此，我们对所选择的技术专题的讨论是非常有限的。

例如，我们没有对个人计算机在分布式系统中的作用进行广泛的讨论。这是因为，运行DOS和Windows的个人计算机几乎没有什么安全机制。解决该问题的方案是在个人计算机上添加第三方安全软件或者硬件产品。我们的总体目标是说明分布式计算安全所带来的业务挑战，而讨论各种厂商的产品对于该目标来说并没有什么用处。我们的讨论重点是分布式客户机系统（包括个人计算机在内）所面临的问题，而不是个人计算机本身。

同样，我们没有过多地描述有关网络和系统远程访问的解决方案。远程访问带来了新的安全挑战，但是市场上有很多该问题的解决方案。讨论这些解决方案无助于对关键问题的探

讨，即如何在不可信的网络上认证个体？

我们知道，尽管有许多人可能会通读全书，但是还有一些人可能只对本书的某些章节感兴趣。为此，我们使用了一个如下所示的指示图，它们可以帮助读者快速找到特定章节的位置：



我们希望这种布局能够为所有读者都起到帮助作用。

目　录

译者序
序
前言

第一部分　理解问题

第1章　计算安全——一个业务问题 ……2
1.1　业务推动 ……6
1.1.1　网络的优越性 ……7
1.1.2　业务环境 ……7
1.1.3　分布式环境是容易的 ……8
1.1.4　客户或者公众的认识 ……8
1.1.5　技术羡慕 ……8
1.2　业务问题 ……8
1.2.1　把安全当成宗教 ……9
1.2.2　成本与风险 ……9
1.2.3　公司承诺 ……9
1.2.4　选择和技术 ……10
1.3　小结 ……10
第2章　分布式安全的挑战 ……11
2.1　几个故事 ……11
2.1.1　杜鹃鸟的蛋 ……11
2.1.2　网上的蠕虫 ……12
2.1.3　物理安全 ……12
2.1.4　密码窃贼 ……12
2.1.5　遗漏的错误 ……12
2.1.6　丢失了什么 ……12
2.1.7　抓住罪犯 ……13
2.2　安全问题 ……13
2.2.1　计算增长 ……13
2.2.2　认识问题 ……13
2.2.3　风险分析 ……14
2.2.4　数据分类 ……14
2.2.5　单一登录 ……14
2.2.6　有人闯入吗 ……14
2.2.7　远程访问 ……15
2.2.8　安全问题来自何方 ……15
2.2.9　网络连接 ……15
2.3　十大问题 ……16
2.4　结论 ……17

第二部分　基　础

第3章　计算安全基础 ……20
3.1　安全的概念 ……20
3.1.1　标识 ……21
3.1.2　认证 ……21
3.1.3　授权 ……21
3.1.4　机密性 ……22
3.1.5　完整性 ……22
3.1.6　认可 ……22
3.1.7　审计和审计跟踪 ……23
3.1.8　安全过程 ……23
3.2　信任——信任的概念 ……24
3.2.1　可用性 ……25
3.2.2　性能 ……26
3.2.3　信任边界 ……26
3.3　信任——需要信任的原因 ……26
3.4　小结 ……27
第4章　安全体系结构 ……28
4.1　基础 ……30
4.1.1　策略 ……30
4.1.2　原则 ……30
4.1.3　规范和标准 ……31
4.1.4　教育 ……31
4.2　信任 ……31
4.2.1　安全 ……31
4.2.2　可用性 ……33
4.2.3　性能 ……33
4.3　控制 ……33

4.3.1 物理访问 ……34
4.3.2 网络访问 ……34
4.3.3 管理 ……34
4.3.4 测量 ……34
4.3.5 监视和探查 ……34
4.3.6 变动管理 ……34
4.3.7 审计 ……35
4.4 小结 ……35
第5章 基础 ……36
5.1 原则 ……37
5.2 安全策略框架 ……38
5.3 安全标准 ……38
5.3.1 标准 ……40
5.3.2 指导原则 ……42
5.4 小结 ……42
第6章 安全策略 ……43
6.1 安全策略框架 ……44
6.1.1 基本的安全元素 ……45
6.1.2 同数据相关的策略 ……46
6.1.3 个人使用策略（通用）……46
6.1.4 安全管理策略 ……47
6.1.5 系统策略 ……48
6.1.6 网络策略 ……48
6.1.7 用户策略 ……49
6.1.8 软件策略 ……49
6.1.9 其他策略 ……50
6.2 策略的例子 ……50
6.3 建立策略的过程 ……51
6.3.1 责任 ……52
6.3.2 安全策略指南 ……52
6.3.3 认识和教育 ……52
6.3.4 策略过程实现 ……52
6.4 小结 ……53

第三部分 技 术

第7章 网络 ……56
7.1 两个网络的故事 ……56
7.2 系统网络体系结构 ……57
7.2.1 体系结构 ……58
7.2.2 高级对等网络 ……60
7.2.3 IBM开放式蓝图 ……60
7.2.4 SNA/APPN安全性 ……61
7.2.5 SNA/APPN小结 ……61
7.3 TCP/IP介绍 ……62
7.3.1 基本的TCP/IP结构 ……63
7.3.2 TCP/IP工作原理 ……64
7.3.3 网际协议是可信任的吗 ……65
7.3.4 提高IP网络的安全性 ……67
7.3.5 将来的开发 ……68
7.4 SNA同TCP/IP的安全性比较 ……69
7.5 结论 ……69
第8章 网络操作系统 ……70
8.1 网络操作系统的功能 ……71
8.1.1 认证 ……71
8.1.2 授权控制 ……72
8.1.3 审计跟踪 ……73
8.1.4 NOS安全方案 ……73
8.2 有关NOS实现的问题 ……73
8.2.1 物理访问 ……74
8.2.2 特洛伊木马 ……74
8.2.3 LOGIN脚本 ……74
8.2.4 密码攻击 ……74
8.2.5 管理的一致性 ……74
8.2.6 GUEST账号 ……75
8.2.7 病毒防护 ……75
8.2.8 工作组计算 ……75
8.2.9 将来的开发 ……76
8.3 结论 ……76
第9章 客户机-服务器和中间件 ……77
9.1 客户机-服务器 ……78
9.2 中间件 ……80
9.2.1 需要中间件吗 ……80
9.2.2 中间件服务 ……81
9.2.3 中间件模型 ……81
9.3 可用技术 ……82
9.3.1 应用程序通信 ……82
9.3.2 远程过程调用 ……82
9.3.3 socket ……83

9.3.4 IBM MQSeries ……83
9.4 分布式对象 ……83
9.4.1 OMG CURBA ……83
9.4.2 对象请求代理 ……84
9.4.3 COM/OLE ……84
9.4.4 SOM和OpenDoc ……85
9.4.5 分布式对象的安全性考虑 ……85
9.5 密切注意发展趋势 ……85
9.6 小结 ……86
第10章 UNIX安全 ……87
10.1 UNIX安全性名声不好的原因 ……88
10.2 UNIX安全 ……88
10.2.1 物理安全 ……89
10.2.2 UNIX认证 ……89
10.2.3 用户的主环境 ……90
10.2.4 组控制 ……91
10.2.5 UNIX认证中的脆弱性 ……91
10.2.6 资源访问控制 ……92
10.2.7 授权中的局限性 ……92
10.2.8 访问控制列表 ……93
10.2.9 ACL的问题 ……93
10.2.10 超级用户访问 ……94
10.2.11 权力的委托 ……94
10.3 典型性的滥用 ……94
10.4 结论 ……99
第11章 进一步探讨UNIX安全 ……100
11.1 UNIX网络服务 ……101
11.1.1 标准UNIX网络服务的工作原理 …101
11.1.2 远程过程调用 ……102
11.1.3 伯克利服务 ……103
11.1.4 远程执行工具 ……103
11.1.5 Telnet服务 ……104
11.1.6 文件传输协议 ……104
11.1.7 普通文件传输协议 ……105
11.1.8 匿名FTP ……105
11.1.9 sendmail ……106
11.1.10 信息服务 ……106
11.1.11 UUCP服务 ……106
11.1.12 网络文件系统 ……107
11.1.13 网络信息服务 ……109
11.1.14 域名系统 ……111
11.1.15 网络时间协议 ……111
11.2 窃贼的工具 ……111
11.3 结论 ……112
第12章 UNIX解决方案 ……113
12.1 控制监视器 ……119
12.1.1 系统管理 ……121
12.1.2 审计跟踪 ……121
12.1.3 动态警报 ……122
12.1.4 安全警报 ……122
12.2 结论 ……122
第13章 Windows NT安全 ……124
13.1 安全控制 ……127
13.1.1 用户配置文件/登录脚本 ……128
13.1.2 访问控制列表 ……128
13.1.3 NT文件系统 ……129
13.2 连网 ……130
13.2.1 TCP/IP服务 ……130
13.2.2 远程访问 ……130
13.2.3 审计和报警 ……131
13.2.4 Window NT安全吗 ……131
13.3 结论 ……132
第14章 因特网 ……133
14.1 因特网的概念 ……133
14.1.1 因特网服务 ……134
14.1.2 公司使用 ……135
14.1.3 因特网上的业务 ……136
14.1.4 问题 ……137
14.1.5 安全需求 ……137
14.1.6 标准和技术 ……138
14.1.7 Java ……139
14.1.8 因特网PC ……139
14.2 因特网防火墙 ……139
14.2.1 防火墙组件 ……140
14.2.2 典型的防火墙 ……142
14.2.3 构建还是购买 ……144
14.2.4 因特网可接受的使用策略 ……144
14.3 结论 ……145

第15章 密码学 ……146
15.1 私有密钥加密 ……147
15.1.1 DES加密 ……147
15.1.2 自动柜员机 ……148
15.1.3 私有密钥的考虑 ……149
15.2 公开密钥加密 ……149
15.2.1 RSA公开密钥 ……149
15.2.2 公开密钥的考虑 ……151
15.2.3 认证中心 ……151
15.3 加密问题 ……151
15.3.1 加密密钥管理 ……152
15.3.2 出口考虑 ……152
15.3.3 评估考虑 ……152
15.3.4 应该加密的内容 ……153
15.3.5 影响和风险 ……153
15.3.6 Clipper芯片 ……153
15.4 数字签名 ……154
15.5 小结 ……155
第16章 DCE环境 ……156
16.1 DCE的概念 ……156
16.1.1 DCE单元的概念 ……157
16.1.2 线程 ……158
16.1.3 远程过程调用 ……158
16.1.4 目录服务 ……159
16.1.5 安全服务 ……160
16.1.6 定时服务 ……160
16.1.7 分布式文件系统 ……161
16.2 关于DCE的问题 ……162
16.3 结论 ……162
第17章 DCE安全概念 ……163
17.1 DCE认证 ……163
17.1.1 客户机认证 ……164
17.1.2 客户机到服务器认证 ……167
17.1.3 认证外部单元 ……167
17.1.4 扩展注册 ……168
17.1.5 服务器认证 ……168
17.1.6 加密客户机-服务器通信 ……168
17.1.7 认证DCE服务 ……168
17.2 授权 ……169
17.2.1 基于规则的授权 ……169
17.2.2 GSSAPI ……170
17.2.3 双因素认证和智能卡 ……170
17.2.4 审计 ……170
17.2.5 单一登录 ……171
17.3 DCE安全吗 ……171
17.4 结论 ……171
第18章 分布式数据库 ……172
18.1 RDBMS的概念 ……172
18.2 启用应用的不同模型 ……173
18.2.1 用户认证 ……174
18.2.2 操作系统访问 ……174
18.2.3 用户配置文件 ……175
18.2.4 授权控制 ……175
18.2.5 责任分离 ……176
18.2.6 批处理SQL语句 ……176
18.2.7 用户组 ……176
18.2.8 角色 ……176
18.2.9 存储过程 ……176
18.2.10 触发器 ……177
18.2.11 远程过程调用 ……177
18.2.12 审计机制 ……177
18.3 有关RDBMS的问题 ……178
18.3.1 附加的解决方案 ……179
18.3.2 传输安全性 ……179
18.3.3 数据合并 ……180
18.4 数据仓库的概念 ……180
18.5 结论 ……181
第19章 联机事务处理 ……182
19.1 事务的概念 ……183
19.1.1 分布式逻辑工作单元 ……183
19.1.2 分布式数据访问模型 ……184
19.2 事务处理系统的组件 ……185
19.2.1 TP监视器 ……185
19.2.2 TP监视器需求 ……186
19.2.3 OLTP是可信任的 ……186
19.2.4 OLTP与数据库 ……186
19.2.5 分布式OLTP ……187
19.2.6 TP监视器组织 ……189

19.2.7 TP监视器 ……190
19.2.8 应用程序设计 ……191
19.2.9 面向对象的事务 ……191
19.3 五大列表 ……192
19.4 小结 ……192

第四部分 解决问题

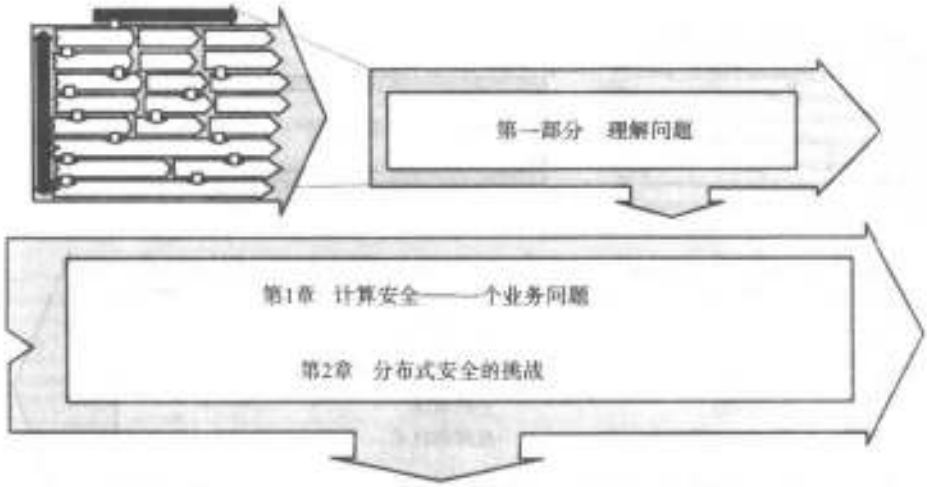
第20章 安全应用程序 ……197
20.1 概念 ……197
20.2 系统开发生命周期 ……198
20.2.1 需求阶段 ……199
20.2.2 设计和分析 ……199
20.2.3 应用开发和测试 ……200
20.2.4 实现 ……200
20.2.5 维护 ……201
20.2.6 认可 ……201
20.2.7 GSSAPI ……201
20.2.8 对象 ……202
20.2.9 基于角色和基于规则的安全性 ……202
20.2.10 重新添加安全性 ……203
20.3 小结 ……203
第21章 实现示例 ……204
21.1 电子邮件 ……204
21.1.1 电子邮件安全需求 ……205
21.1.2 标准 ……205
21.1.3 电子数据交换 ……208
21.1.4 保密增强邮件 ……208
21.1.5 良好保密 ……208
21.1.6 同电子邮件有关的问题 ……209
21.2 Lotus Notes ……209
21.2.1 Lotus Notes的安全性 ……210
21.2.2 用户认证 ……211
21.2.3 Lotus Notes邮件 ……211
21.3 下一步的展望 ……211
21.4 小结 ……212
第22章 安全管理 ……213
22.1 系统管理 ……214
22.1.1 访问控制解决方案 ……216
22.1.2 单一登录 ……216
22.1.3 密钥认证 ……217
22.2 网络管理 ……217
22.2.1 什么是SNMP ……218
22.2.2 SNMP足够强大吗 ……219
22.2.3 网络事件管理 ……219
22.2.4 动态监视 ……220
22.2.5 Andromeda ……221
22.2.6 安全忠告 ……222
22.2.7 飞虎队 ……222
22.3 结论 ……223
第23章 开发安全策略 ……224
23.1 安全策略 ……225
23.2 安全策略路线图 ……227
23.2.1 当前评估 ……227
23.2.2 范围和假设 ……230
23.2.3 需求分析 ……230
23.2.4 体系结构 ……230
23.2.5 建议 ……230
23.2.6 候选方案 ……231
23.2.7 成本 ……231
23.2.8 推荐的解决方案 ……231
23.2.9 风险和影响 ……231
23.2.10 战术计划 ……232
23.3 结论 ……232
第24章 审计 ……233
24.1 审计的概念 ……234
24.1.1 审计者 ……234
24.1.2 常见的错误 ……235
24.1.3 计算审计重要的原因何在 ……235
24.2 审计的角色 ……236
24.2.1 关系 ……236
24.2.2 建立正确的标准 ……237
24.3 UNIX审计标准示例 ……237
24.4 计算机审记基础 ……238
24.5 扩大重点 ……238
24.5.1 用户意识 ……239
24.5.2 业务持续计划 ……239
24.5.3 物理控制 ……239
24.5.4 软件许可 ……240

24.5.5 软件开发 ……240
24.6 其他的审计类型 ……240
24.6.1 风险评估 ……240
24.6.2 脆弱性测试 ……241
24.6.3 自评估 ……241
24.6.4 性能审计 ……241
24.6.5 审计时间 ……242
24.6.6 认可 ……242
24.6.7 自评估工具 ……243
24.7 结论 ……243
第25章 未来 ……244

附 录

附录A 强认证 ……249
附录B 智能卡 ……253
附录C 个人计算机的安全 ……256
附录D 远程访问 ……259
词汇表 ……261

第一部分 理解问题

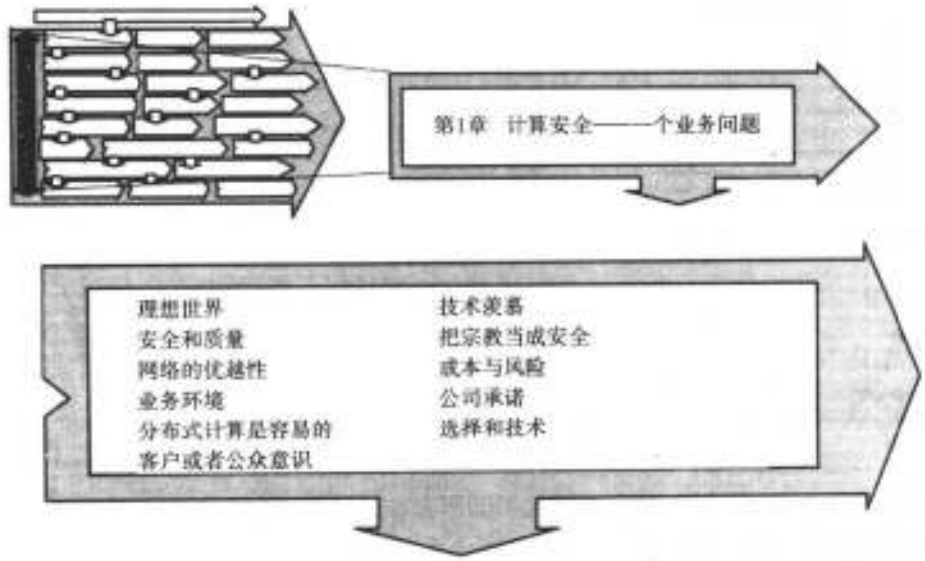


不管处理什么问题，第一步总是要先明确所存在的问题。在本书的前两章里，我们将详细阐述分布式计算安全所包括的问题以及推动其发展的原因。你会看到计算安全是一个非常复杂的问题，它涉及到很多方面，其中包括技术因素、不确定的管理承诺以及用户意识的缺乏等。由于问题的性质，计算安全是不能只靠技术来解决的。事实上，涉及过多的技术反而增加了问题的难度。鉴于它的复杂程度，解决该问题需要有一个战略上的方法。

计算安全问题和家庭安全问题有一些相似之处。大门是进入一个家庭的正道，因此对它们需要多加注意。门并不是非法进入一个家庭仅有的路径。例如，要是没有关上窗户的话，则可以从窗户进入。敲碎窗户玻璃、拉开插销，这对入侵者是轻而易举的。所以如果窗户没关好的话，那么即使是世界上最好的门锁也保护不了你的财产。同家庭安全一样，分布式计算安全也不是“锁上前门”就万事大吉。它具有很多因素，这包括对系统管理的策略、用户意识、重视以及技术上的正确实现。

在第1章中，将对计算安全问题进行总体上的阐述。我们将着重关注分布式计算环境，并且看一下分布式计算是如何增加计算安全问题复杂程度的。我们将对该问题的原因和业务驱动因素进行分析。第2章会激发你的兴趣，它将介绍一些真实的安全事故。当涉及到分布式环境的时候，我们会研究更具体的安全问题。本书的起始部分着眼于介绍同安全问题相关的因素。在后面的各章中，我们会阐述如何采取具体的技术和行动来解决分布式计算中的安全问题。

第1章　计算安全——一个业务问题



最近几年来，任何组织内部所安装的计算机系统的数目正在显著增长。安装和使用计算机应用相对很容易，这促使人们把计算机系统连接在一起并对他们的工作进行分布处理或者共享。这些分布式系统可以更好利用现在可用的处理能力。然而，这些系统能够保持机密数据的安全吗？所有的数据都保持同步吗？你信任这些计算系统所产生的结果吗？你相信一个分布式系统能够安全地为你提供所需要的一切东西吗？

曾经有一段时间，所有的数据及其相关的处理应用都只在一个地方可用，如一台计算机。任何人都能自豪地指着一台计算机说“这就是神奇之所在”。系统的保护是非常简单直接的：首先找一个绝对安全的环境，然后把系统上需要安全保护的任何东西放到该环境中、或者把系统放到一个完全安全的环境中。在今天的处理环境中，通过使用几个不同级别的计算机，其中每个包含一部分数据和应用，就可以使前面的神奇随处可见。对于哪里需要保护、需要什么以及如何实现等问题，是很容易失去线索的。数据或者处理应用实际所在的位置可能并不是很明显。对于只存在一家银行中的钱和分散在自动柜员机网上的钱来说，显然前者更安全。

现在的家庭计算机所具有的处理能力要比几年前一个一般的百万美元级保险公司用来管理其全部业务的计算机所具有的处理能力还要强。处理器的能力在持续增长，同时这种处理能力的单位成本在逐渐下降。廉价处理能力的快速膨胀使得操纵和处理不断增大的数据量并且及时提供结果成为可能。利用唾手可得的廉价计算能力是非常有意义的。随着局域网和其他网络技术的快速发展，这种处理能力可以成为分布式的并且由多人所共享。但是，如果没有足够的处理和控制，那么这种数据和处理的分布化可能会导致更多难以解决的问题。在本章中，我们将看一下什么是分布式系统，并且讲述一些同时带来机会和问题的驱动因素。

理想世界

计算机是一种非常灵活的工具，它几乎能够进行你通知它的任何工作，前提是必须使用一种计算机能够理解的方式。理想的计算机系统能够为你做什么？它能提供对任何所需数据的访问，并且遵照所要求的格式。对数据库的任何更新都将立即可用。对于任何应用来说，不管位于何处都能够执行。系统总是可用的，并且总是能够提供即时响应。可以信赖数据的安全性和完整性，并且对数据的任何更新都会成功完成。如果出现了一个系统问题，并且系统变得不可用，那么系统会立即恢复并且数据会还原到故障出现前的那一时刻，从而使得系统重新变为可用。如果你所需的全部数据和应用都位于自己的机器上，那么前面的各种理想是可能的。如果数据和（或）应用程序分布在两个或多个计算机上，那么若要达到这种理想状态就变得非常复杂了。

分布式计算同汽车类似的原因

你知道若要让家庭汽车始终保持最佳形态，那么需要付出什么成本吗？不久前你可能刚把汽车开到街角的服务站，在那里你的汽车受到了仔细的维护，这包括修理扁了的轮胎、更换刹车、安装新电池、修理引擎、校直方向盘、更换机油以及进行其他任何所需的修理工作。然而今天，街角的服务站在什么地方？现在有专门的“润滑油商店”能够在10分钟内完成汽油更换，并且也有专门的发动机修理处，以及消声器、刹车以及减震专业人员。对于汽车维护的每一方面，现在几乎都有专门的商店。在今天，街角服务站只卖汽油、彩票以及一些软饮料。全能服务站几乎已经全部消失了。

不管在什么时候什么地方需要进行汽车维护，作为汽车的主人，你有责任保证所有的维护工作都会完成，然而现在这些工作已经分散到几家专业服务处了。当购买一辆新车的时候，通常你会得到一份保险以及同销售商的一份服务合约。只要销售商提供服务，不必担心你的汽车会怎么奔驰。你可以相信：一切功能正常，汽车会把你送到任何想去的地方。然而，一旦保险合约过期了，那么销售商的继续服务可能会非常昂贵。你可以选择一个相对便宜些的途径，把汽车送到专门的商店进行维护。这样问题接踵而至：需要跟踪和管理所有的服务需求从而使得你的汽车总是可靠的。

把数据和处理分布到多台计算机上同汽车维护非常类似。单厂商大型机环境同从销售商处购买新车类似。在这种情况下，所有的服务和保证都来自同一个地方。如果分布式系统的所有组件也来自单一厂商，那么这也类似于从销售商处购买新车。如果系统组件来自于许多不同的厂商，那么这类似于使用专门的商店来维护汽车。你需要保证全部所需的组件都到位并能一起协同工作。

图1-1说明了集中式或者大型主机系统操作同操作来自一个提供商的或来自多个提供商的分布式系统之间的区别。在分布式多厂商系统中，安装技术基础的成本更低并且灵活性更好，但是其成本是提高了复杂性并降低了管理分布式系统的能力。来自单一提供商的分布式系统则位于全分布式系统的成本和灵活性和集中式（或者大型主机）系统的复杂性之间。在许多情况下，如果考虑所有的处理和管理因素，那么分布式系统的成本并不会更低。

回到汽车例子中。如果使用销售商维护方案，那么成本可能会更大，但是不用担心维护时间安排以及维护工作。如果让专业维修商店来维护汽车，那么会有更多的选择，并且会得到互相竞争的价格：但是可能需要奔波于很多地方才能完成一次维护。也可以选择自己来做一切工作，但是需要掌握复杂的汽车技术。如果你是一个熟练的机修师并且打造了一辆高度

自制的汽车，那么很有可能你是惟一一个能够维护和改进它的人。

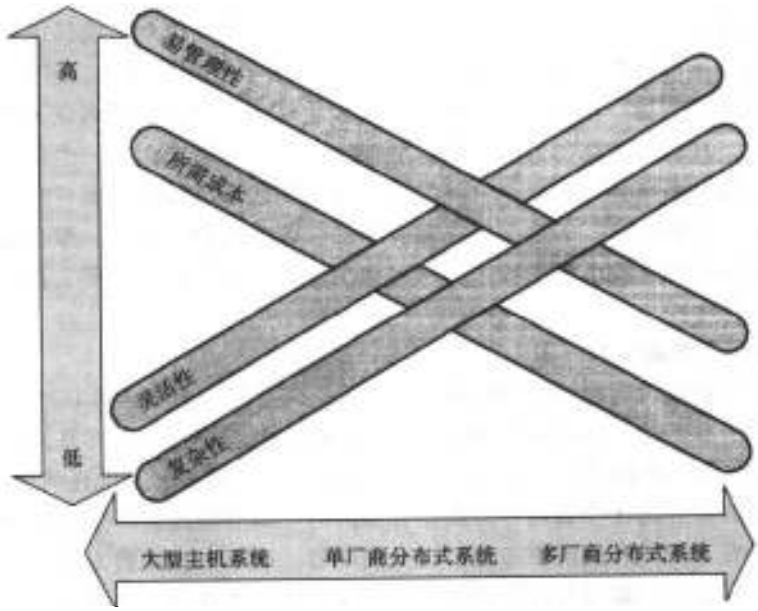


图1-1 大型主机与分布式系统

对于个人计算机来说，电子表格程序的使用可以说是最有价值的创新之一。使用电子表格程序，可以立即做出你的年度部门财政预算来。这些部门预算可以汇总在一起生成分公司预算，然后所有分公司预算可以汇总成总公司预算。然而，用于产生公司预算的数据和处理可能分布在几个不同的地方以及不同的计算机上。当为了满足公司目标而必须对部门预算进行修改的时候，处理过程会发生什么情况呢？你会被通知：削尖铅笔，重新填写你的预算，然后重新提交新的预算单。

上述情况必须使用协调机制来保证当前所有的预算信息都为下一个预算单所使用并且保证机密性。产生公司预算的主要工作是数据的协调而不是数据的产生。如果使用了一个集成的预算软件包或者一个分布式数据库，那么上面的工作大部分可以自动完成。如果该过程不是自动的，那么你可能会为确定是不是包括了最新的电子表格文件或者是不是使用了正确的磁盘而头痛，并且可能还会有很多人会为此饱受折磨。如果产生公司预算所需的数据和过程管理良好，那么你可以相信公司预算的准确性。

分布式计算的概念

对于不同的人来说，分布式计算可能意味着不同的事情。对某些人来说，它意味着客户机-服务器；对于另一些人，它意味着协作处理；而对于其他的人，它还可能意味着使用分布式数据库。我们可能需要一个更详细的解释和定义，从而保证人们会对分布式计算系统有一个共同的见解。当引用一个分布式计算系统的时候，每个用户应该能够看到同一件事物。它应该同人们看一辆汽车类似，大家会看到同样的事物；而不是像艺术品那样需要分开解释。几乎每个人都知道一辆汽车通常看起来是什么样子并且是用来干什么的。但是对于艺术品，人们可能会有多种评论。一个一般的计算机处理应用可以分成多个组件，其中每个组件负责处理过程的不同方面。在这种情况下，我们假设计算机处理过程是一个人通过键盘或者其他

输入设备启动的。图1-2说明了一个一般的计算处理的各个组件。

终端用户设备	终端用户界面	应用程序	数据管理	数 据

图1-2 处理过程组件

每个组件可以是一个独立的单元，也可以聚合成一个处理单元，并且它们可以通过网络连接同其他组件进行通信。这些组件可以通过多种方式来隔离和组合，而这可以使用分布式计算模型来描述。图1-3说明了这些组件的一些常见组合情况。如果终端用户界面、应用程序、数据管理以及数据都包含在一个处理器内，那么我们可以使用通常的中心主机处理模型。使用个人电脑（PC）你可以修改该模型来提供终端用户界面，然后通过该主机来提供应用程序、数据管理以及数据。该模型称做远程表示（remote presentation）。

如果把应用程序移动到本地PC上并且在一个主机计算机上维护数据，那么这通常就称做是客户机-服务器计算。如果把应用程序分成多个运行在主机和本地PC上的组件，那么这就是协作处理。如果本地PC上的应用程序也具有一个能向主机数据库请求服务的数据管理器，那么这就是分布式数据库。如果所有的组件分布到多个不同的处理器上，那么这就是完全分布式系统——除非实现了大量的管理过程，否则它极有可能是非常混乱的。

我们对系统的定义是，由多个互相连接的或者相关的不同元素为执行一个由单个元素所不能执行的功能而组成的集合。一个分布式计算系统是由系统元素所组成的，这些元素分布在通过网络互相连接起来的不同处理平台上。分布式计算系统所需的性质之一是透明性，这是指为了让分布式环境中的系统元素能够按照同单主机环境中一样的方式来交互和操作的一种能力。如果要让不同系统组件之间的交互在多主机分布式环境中与在单系统环境中保持一致，那么透明性是必需的。

客户机-服务器的概念

有很多术语看起来抓住了信息系统（information system，IS）行业的想像力，但是看起来又有多种意义（这取决于谁在下定义）。它的定义范围从简单的PC网络连接到精心制作的多级别多地点应用程序和数据集成。在最简单的形式中，服务器是一种或多种资源的管理者，而客户机是服务器资源的用户。通常客户机也看做是可编程的工作站，但这不是一个限制定义。在简单的术语中，客户机-服务器可看做是分布式计算的一种实现，数据和应用程序驻留于服务器中，而客户机负责提供终端用户界面。不幸的是，一个简单的定义通常意味着术语客户机-服务器可使用到几乎任何牵扯到多个计算机的行为中。当一个客户机-服务器系统看做是一种最新技术并且用做一个卖点的时候，这一点特别正确。对于解决安全问题的目的来说，客户机-服务器将会做为分布式计算的一种实现来对待。在本书中，当需要时，同客户机-服务器实现相关的内容会被指出来。

在大型处理系统中，安全是一个相对简单的问题。在这种系统中，所有的处理都是在一个中央计算中心上完成的。你所需要做的就是在这个中心计算机周围挖上一条护城河并且只建造一个吊桥。任何欲进入的人都会一视同仁，出示相应的许可证。这样只需要一组可以识别进入者的守卫即可。当未经授权的行为发生时，出现什么情况是很明显的。现在有许多计算机链接到了吊桥上。我们有许多护城河吗？谁负责维护护城河？如何保证所有的护城河都是同一大小并能进行同一工作？

图1-3 分布式处理模型

安全和质量

安全在今天的业务系统中不仅仅是信息在技术层面上的保护，并且自动业务的每一个方面也必须考虑和包括。质量程序近来变得非常流行，它使得公司可以提高他们的产品及其开发过程的质量，从而保持或提高公司在市场上的位置。作为向其客户（或者在大的方面说向全世界）做出质量保证的方式之一，许多公司都已经实现了那些广泛接受的标准，如ISO 9000系列等。成功的公司质量程序已经从起始于上层决策者的公司承诺中产生。

我们相信计算安全同质量一样，也必须被公司接纳并且要包括在业务操作的各个方面。客户和公司要评估他们的提供商和其他业务关系对安全标准的承诺和实现情况。如果你认为一台ATM（automated teller machine，自动柜员机）或者它所连接的网络所提供的安全性没有另一台ATM所提供的安全性高，那么你可能不会使用前者。

比较一家公司同另一家公司的操作需要安全标准。公司间在线事务处理（电子数据交换和电子商务）的实现需要合适的标准。对于一个组织来说，大型业务合同的签订已经开始取决于交付和操作一个安全计算方案的能力。你可能必须证明它有多么安全。安全的正确实现可能是一个有竞争力的优势，而缺乏足够的安全一定是一个竞争力的劣势。

1.1 业务推动

是什么真正推动了对安全的需要？对我们的工作来说，下面这个事实是非常明显的，即安全计算环境是业务问题而不是技术问题。现在有大量可用的技术处理方案，甚至更糟的是，有大量的基于技术的标准。问题不在于应用什么技术，而是在于应该把技术应用到什么上以及如

何应用。太多的技术以及太多的标准，但缺乏计划，也缺乏管理，这会带来安全风险，甚至带来灾难。缺少管理能够为分布式环境带来漏洞和缺口，也同大门敞着、窗户不关没什么区别。如果缺乏对这种情况的认识，那么你就会冒更大的风险，并且有更多潜在的不利条件。

1.1.1 网络的优越性

一个主要的业务推动是计算机网络所带来的冲击，即计算机进行通信的能力。几乎每个有多台计算机的公司都会有某种形式的计算机网络。并且，几乎所有困难的网络问题都已经解决了，从而使得计算机的互连变得相对很容易。网络技术是容易实现容易使用的。公司网络已经扩展到包括业务伙伴、客户、合约关系以及进行家庭办公的员工。

在某些情况下，公司网络甚至已经扩展到了直接的竞争对手。例如，连接到工业市场管理组织的网络，或者支持具有多个相似组件提供商的大型工程的网络。你可能希望能够在晚上休息的时候也能对公司的所有资产了如指掌，并且公司的信息资产能够受到保护而不会被窃取或者被破坏。尽管大部分的网络技术问题已经解决了，但是计算机通信的容易性为安全维护问题带来了新挑战。你不仅仅要关心自己的机器的安全性，也需要关注同其相连的所有其他计算机的安全性。

1.1.2 业务环境

最近的情况显示，公司开发计算机系统来对其业务处理进行自动化，并且力求完美。相对于传统的手工业务处理来说，这种电子化方式确实大大提高了工作效率。自动化产品的使用累积了大量有价值的数据。当人们开始解决如何访问这些有价值数据的时候，业务处理的另一波浪潮又到了。这种对数据的访问类型通常同其他自动化处理过程有冲突。在公司里，信息技术（Information Technology，IT）部门掌握着计算处理如何工作以及如何访问数据这些知识。安全问题由IT部门的某个人员负责，并且是很容易实现的。如果计算机系统不可用，业务能够仍旧进行，但可能有点别扭。

现在，自动业务处理的当前状态是完全不同的。几个主要的推动力量正在汇集起来改变计算技术对业务提供支持的方式。技术上的许多进步使得安装的计算能力几乎成指数增长，并且其价格不变甚至低于原来的系统。更多更好的自动工具正用来支持业务。视频、图像、声音以及语音记录等都可以是获得或者提供信息的方式，它们正在改变着计算机对业务提供支持的方式。业务决策过程已经从根本上改变了，并且我们希望能够更好。我们越依靠系统生成和提供的数据，对这些数据就需要越多的信任。

业务处理过程是由经济上的需要决定的，而一个更大的改变已经重构了业务处理本身。仅仅对业务处理进行自动化是不够的，它们必须在计算机技术的支持下重新从头构建。计算机不再仅仅是一个工具，现在它们已经成为操作业务的一个集成的重要部分。现在，业务功能非常依赖计算机，所以你不得不信任它们。对某些非常依赖计算机系统的业务，即使他们的系统出现故障只是很短的时间，那么这些业务也可能会停止进行，至少是运行非常不便。

这种业务的再建工程也对系统用户的需求带来了根本性的改变。业务自动化的第一次浪潮根据系统输入使用计算机系统来分析数据、创建报表以及提供决策建议等。用户提供的输入，并按系统给出的决策建议采取行动。系统的第二次浪潮把数据的输入移到了产生数据的地方，从而避免用户干预数据的处理。银行系统引导着这一波浪潮。

现在，我们使用的系统已经发展成能够支持知识工人，这些知识工人在他们的系统配置中有许多应用程序，并且他们有能力访问几乎所有为提高工作效率所需的数据。他们访问的应用程序和数据可能分布在世界上的任何地方。并且完成任务所需的应用程序和数据可能并不总是可预测的。我们已经取代了汇编行，我们需要知道具有高度知识的用户进行数据处理的方法以及访问他们所需的几乎任何计算机资源和数据的能力。对机密信息进行保护的机制也必须能够跟上新的业务处理方式。

1.1.3 分布式计算是容易的

分布式系统可以很容易地组织在一起，这是另一项推动力。由于某些原因，人们更多关心和注意的似乎是某些困难并且结果不明确的事情，而不是那些容易并且结果可预测的事情。如果分布式系统容易实现，那么安全问题似乎更少有人关心。计算成本同样如此。如果分布式解决方案是便宜的，那么计算成本问题恐怕没人过问。几乎任何人都能够获得并实现一个可以参与分布式计算系统的系统，但是保护机制必须加进来并且要能够跟上这种扩展的使用要求。因为分布式计算是容易的，所以你最好能够信任它。

1.1.4 客户或者公众的认识

当从自动柜员机中提钱的时候，或者更重要在存钱的时候，我们需要保证事务是绝对安全和机密的。我们相信刚完成的事务是对我们的银行账户所进行的同一次事务，并且其他任何人对该项事务都一无所知。如果有人使用这些机器来破坏或者窃取银行的信息，并且我们的事务也受到牵连，那么我们会很快停止对那些机器的使用，甚至有可能不再光顾该银行。安全性不仅仅应该是严格的，它也应该被广泛地理解。由于越来越多的组织使用计算技术和网络来处理组织间或者同客户的业务，所以处理事务时的安全性必须被认为是强有力的。安全性可用做一个卖点。尽管从看它可能不是一个大问题，但是如果一个组织的计算环境比其竞争对手的计算环境更安全，那么前者就有了一项很有竞争力的优势。

1.1.5 技术羡慕

隐藏在许多新计算机系统之后的另一项推动力可以称做技术羡慕。进行信息处理的很多人某种情况下都需要最新最先进的技术。人们总是需要拥有最新的技术以及最快、最强大的系统。信息处理最初用做业务处理自动化工具。需要实现新一代自动化系统是因为技术的发展而不是因为业务需求的改变。在许多情况下，这节省了对数据处理硬件和软件的预算。然而，操作业务的成本并不总是降低。现在，技术过时是按照月而不是年来算的。

有时候，应用信息处理技术的决策和策略是建立在最流行的技术而不是业务需求的基础上的。一篇发表在航空公司杂志上的优秀文章可能会对一家公司的策略产生重大的影响。人们使用新技术通常只不过是因为怕落伍。事实上，许多公司已经停止把他们的计算进行分布化，但是他们根本没有认识到他们的方向，或者根本没有理解所牵扯的业务问题。安全考虑通常滞后于对最新最先进技术的需要。

1.2 业务问题

前面所提到的业务推动力点明了分布式计算是业务的一个重要部分的原因。然而，分布

式环境带来了许多重要的问题。这种环境要求考虑更周密，并且在大的范围内制定计划。对于那些曾经使用简单的技术实现来保护资产的地方，现在则需要提供所有那些分布式元素能够采纳并且必须采纳的规则和结构。

1.2.1 把安全当成宗教

有时候，安全方法更多地是建立在同宗教信念类似的基础上，而不是建立在所定义的需求和原则的基础上。在分布式环境中有两个或者更多的“宗教”。一个分布式环境中的安全方法可能完全不同于另一个分布式环境中的安全方法。如果不同环境的大门需要不同的钥匙，那么跨环境的应用和数据共享是非常困难的。安全太多和安全不够同样都是问题。如果安全机制变成了用户工作效率的一个负担，那么用户可能需要进行大量的工作来绕开它们。

公司管理对安全有不同级别的需要，并且其受安全影响的程度也不同。这些不同使得存在着不同的观点和意见。某种管理级别可能特别重视员工工作效率，它把严格的安全机制看做是对完成业务任务的障碍。更高级别的管理更重视业务的操作和知识产权的保护。有很多这样的例子，公司秘密被窃取并卖给其竞争对手，然后其竞争对手使用这些信息来生产出更具竞争力的产品，从而挤垮受害公司。

1.2.2 成本与风险

平衡安全解决方案的成本和控制这些方案所带来的风险是一个重大挑战。提供一个安全解决方案的成本应该同安全机制所保护的资产的价值成比例。该成本也必须要与同任何潜在损失或者资产损失相关的风险取得平衡。即使资产被看做是很有价值的，为防范所有可能出现的风险所花费的成本也可能是不合算的。必须要在安全成本和潜在风险之间达成平衡。

风险可以看做是业务系统的危险程度、机密信息的暴露危险或者通过欺骗引起的潜在损失。一个敏感的风险分析也应该包括对员工和客户安全的考虑。在考虑方案之前，务实的分析应该完全掌握所存在的风险以及这些风险所带来的影响。然而，客户的资产需要受到保护。

风险的本性同技术一样，会不断变化。由于网络连接的相对缺乏，来自外部的风险曾经看做是非常有限的，但现在这种风险在增长。企业间和企业内部对通信的需求以及通信能力都在持续增长。将来，任何公司都可以通过数据高速公路同客户、其他公司、提供商以及政府进行电子通信。这提高了对组织内部管理控制的要求，为防止信息暴露，不仅要对付内部欺骗同时要对付可能的外部攻击。

1.2.3 公司承诺

前面，我们将公司质量承诺和安全承诺做了对比。若能够提供分布式环境的安全性，公司领导层需要具有安全意识，并且把安全应用到所有的分布领域中。方法和实现的任何差异必须要在企业中达成共识。许多组织实现了安全处理，但没有提供足够的基础（包括原则和政策），而这些基础可以使得安全程序能够更加有效。安全是一个公司问题，应该作为公司问题来对待。公司可能需要一个良好定义的安全策略。

公司承诺的另一个重要方面是对员工安全意识的培养和教育。只有员工认识到安全的重要性以及每个人所承担的责任，公司资产的安全性才是真正有作用的。这种意识必须扩展到组织的所有地方。系统的设计和开发必须把安全需求牢记在心，这应该同把系统和网络的操

作技巧牢记在心一样。如果人们不知道安全对于组织是非常重要的，并且也不知道他们应该为此付出什么，那么他们不会注意安全问题。

1.2.4 选择和技术

不幸的是，没有能够解决所有的分布式安全问题的“魔杖”。解决这一问题没有一个合适的方法或者技术方案。对于信息的重要性和分布式环境中涉及的相关技术风险，经常存在相互冲突的目标，或者受到不同程度的关注。对于什么是安全或者需要什么安全这个问题，通常会有互相冲突的答案。在信息访问的方便性和安全性之间往往有一个平衡。关于安全方案的任何决定都只能在全盘考虑以后才能完成。

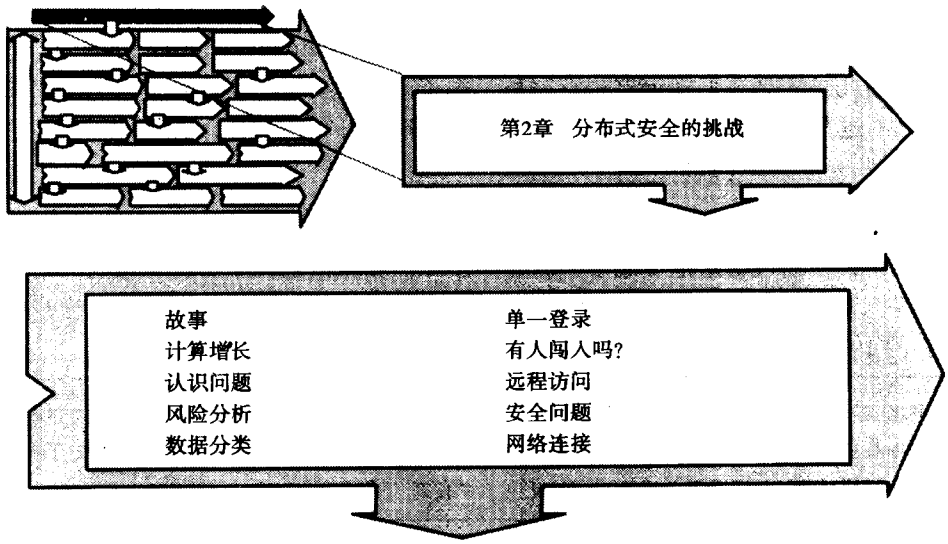
对于安全问题，人们通常急切地需要找到一个技术方案。有许多不同的安全方案可用，这使得问题变得更加复杂。从前面可知，分布式计算安全是一个业务问题而不是一个技术问题。在真正了解需要保护什么以及为什么需要保护它们之前，寻找一个技术方案是没有多大意义的。跟上所有的分布式和安全标准以及它们可能对组织所产生的影响是非常困难的。许多标准现在仍然在发展。

1.3 小结

在当前的自动系统中，分布式计算是一个非常重要的因素。有很多原因可以解释这一点。若要保证分布式环境中的信息资产受到保护，那么公司或组织需要考虑很多方面和很多问题。分布式计算安全是可能的吗？值得为它付出的成本吗？提供分布式系统安全性的挑战在其获得更好的解决之前可能会变得更严重。然而，如果正确处理了安全需求，那么分布式系统的好处要远远超过其风险。我们将试图提供一些帮助从而使得人们可以更好地理解问题，并且提供如何解决这些问题的建议。

在下面几章中，我们将试图更详细地解释所存在的挑战，并且试图帮助你着手对付这些挑战。当你掌握了问题及其解决方法以后，分布式系统安全就是可以达到的。在第2章中，我们将继续评论分布式环境所带来的更具体的挑战。

第2章　分布式安全的挑战



分布式计算中的安全性不容易得到。同1849年穿过大平原到达加利福尼亚州和俄勒冈州的开拓者们一样，这里也有许多危险需要战胜。希玛和俄勒冈的足迹表明那些开拓者们跨越了许多宽阔的河流和沙漠。天气是常年的敌人，并且存在着迷路的可能性。但是通过使用其他人的经验，就能认识到危险并且能够对付。拓荒故事千奇百怪，安全问题同样如此。没有任何一个方案或者技术能够解决我们可能遇到的所有问题。

2.1　几个故事

本章介绍一些危险和应该考虑的问题。就如同那些穿过大平原的人们所做的那样，我们也首先从几个故事开始……。

2.1.1　杜鹃鸟的蛋

在1986年8月，Clifford Stoll，一个天文学家，后来成为加利福尼亚大学劳伦斯伯克利实验室中的计算系统管理员，发现每月计算机账单记录中有一个75分的差额。研究人员开始对该差额进行跟踪，结果发现了一个德国黑客组织，该组织已经渗透到了世界上许多政府和军事网络里。使用相对并不复杂的技术来攻击脆弱的安全实现，他们能够穿透许多计算机站点。在对其行为进行了数个月的跟踪以后，这些黑客们最终被逮捕归案。在跟踪过程中，跟踪人员试图得到几个对该问题感兴趣的不同代理。Clifford Stoll在《The Cuckoo's Egg》[Stoll,1989]一书中完整地介绍了这个故事，这本书变成了一本流行的计算机黑客读物。

2.1.2 网上的蠕虫

在1988年11月2号，上千台连接到互联网上的VAX和UNIX系统开始神秘地瘫痪。它们感染了一种称做蠕虫的特殊计算机病毒。该蠕虫启动一个又一个的进程，从而迅速消耗掉受感染计算机的资源。最终，受感染的计算机会瘫痪掉。

蠕虫也找到了网上的新受害者。利用操作系统中鲜为人知的安全漏洞，蠕虫在整个互联网上迅速地繁殖，最终使上千台计算机的速度慢得像爬行一样，或者使得计算机脱离了网络。该蠕虫的作者是Robert T. Morris，它是美国国家安全局（National Security Agency）的首席计算机科学家的儿子。当人们知道这一点以后，都被震惊了。Morris是一个大学毕业生，他宣称该蠕虫只不过是用来测试计算机安全性的一个没有成功的实验而已。

2.1.3 物理安全

在1969年，一场教学示范变成了一场财产破坏，其原因是位于蒙特利尔的乔治威廉姆斯大学的计算机系统受到了攻击。该计算机系统被占领了，计算机的打印输出从窗户中飘出来并散布在下面的大街上。

> “我还记得当时的情形，”我的祖父说，“那是在安大略省的一个寒夜，我们的马拉雪橇被一群狼包围了。我拿着枪跳到了狼群中，但枪却哑火了。”然后我的祖父叹了一声气。我用害怕的嗓音问他，“后来呢？”他说道：“我死了”。

2.1.4 密码窃贼

在1994年2月，卡内基梅隆大学的计算机应急小组（Computer Emergency Response Team，CERT）发布了一个重要报告。报告发出警告，指出入侵者们使用受害系统在互联网上窃取密码。据报告，遭到窃取的密码已经有数万之多。黑客使用诊断软件来截获发送给他们已获得访问权的系统的流量。大多数受害系统都没有注意到这种偷窃行为。

2.1.5 遗漏的错误

许多计算机厂商向客户交付系统时会附带一些诊断工具，这些工具是由一个单独的用户账号所支持的。这些账号通常有一个共同的密码，如support。联机文档指出一旦系统正确安装以后，这些密码应该马上改变。如果安装后没有改变密码，那么就意味着用户的系统敞开着大门，而该信息会通过BBS（bulletin board system）快速传播开。

2.1.6 丢失了什么

有一位制造商在经过一个长周末后返回办公室发现其办公室被闯入了。经过一番仔细搜查以后，发现什么也没有丢。只不过后来，当这家公司丢失了一个同竞争对手的重要合约时，他们才发现入侵者把含有同这份合约有关信息的计算机文件给拷走了。这种信息盗窃可能不易发现，但是它是非常有害的。

2.1.7 抓住罪犯

一家软件公司逮住了一个非法使用其产品的用户。该公司提供其新产品线的免费演示软件。该演示软件装到客户的PC里。除了进行演示以外，该软件还会在硬盘上搜索对该公司软件的非法拷贝。该软件公司深受盗版者青睐。如果发现一次非法拷贝，那么邀请用户打印、填写并回寄一个用于免费产品手册的凭单。几百个用户都这么做了，但是他们没有收到手册，取而代之的是收到了来自该公司律师的一封非法软件索赔信。

2.2 安全问题

今天的业务处理方式受到计算机的自动支持。我们把自己公司的业务处理进行自动化，并且希望其他公司的业务处理也会有足够的自动化，从而能够在不减损我们的自动化系统的情况下同他们进行业务来往。为了保持业务往来的效率，这些关系的信任方面必须要得到保证。现在我们同客户、提供商以及伙伴有各种不同的关系，这把我们带入了一个同这些实体的信任关系网内。为了这种信任，我们必须要有一个坚固的基础。为了建立和维护这种信任，我们需要处理许多问题。在本章中，我们将对信任所需的一些具体方面进行说明。

2.2.1 计算增长

在第1章中，提到了促进了分布式计算系统的实现的业务推动力之一是它的相对容易性。其他业务问题刺激了分布式系统的增长，以及对安全性的需要。过去的小型部门打印服务器现在已经发展成全功能微型数据中心。在许多情况下，它们仍然只像打印服务器一样操作和保护。在公司小型化的世界中，运作部门的分布式计算系统一般没有补充信息系统专业人员。公司需要对安全功能进行集中式管理，但是用于这方面的工具仍旧不成熟或者不可用。通常，复杂性的增长和系统数量的增大使得提供安全环境变得非常困难。

2.2.2 认识问题

把集中式数据中心的计算能力分散到不同部门中也导致了系统数量的增大。把应用从大型主机移到开放式平台上是个很大的工程。在许多情况下，这意味着用几台计算机来代替一台大型机器。有效安排多台计算机及位置所带来的挑战为许多公司的防卫系统带来了问题。在许多情况下，管理部门没有认识到安全责任的改变以及这种新型分布式环境所带来的挑战。

必须承认，在分布式环境中问题的明确是非常困难的。一方面，分布式平台的安全性并不非得同集中式系统完全一样；另一方面，“大场面”的认识，其中包括所有的分布式平台，可能不是很明显。我们需要分析整个企业所受的影响，而不仅仅局限于当前的近邻。也需要认识什么时候处理陌生人，什么时候处理近邻。通常，一个公共信任是同近邻形成的，但是我们也需要考虑陌生人访问的需求。

对于许多组织来说，计算机病毒正成为一个严重的问题。事实上，在最近一次调查中，超过2/3的组织报告说他们受到了计算机病毒的影响。这种影响的范围小到简单的恶作剧病毒，大到对企业数据的真正破坏。病毒可以通过多种途径进入组织。员工的家庭计算机中可能有病毒，如果把软件或者共享软件带到办公室，那么病毒就有可能也带到办公室。甚至，来自厂商的许可发布软件也可能会带有病毒。从系统中清除病毒的成本是很大的。该问题的影响

必须要认识到，并且需要采取措施以维护一个“干净的”环境。如果不进行杀毒工作，那么更多的病毒变种会大量繁殖。

2.2.3 风险分析

任何资产保护都需要包括风险分析。你的资产需要进行什么程度的保护？采取控制可以减少风险，但这是需要成本的。引入了越多的安全性，控制的成本就越高，同时希望预期损失的成本就越低。如果你对风险、需要保护的资产以及控制的成本有清晰的认识，那么就可以得到一个平衡的安全系统。不幸的是，在计算机系统中，控制的成本通常要比风险以及需要保护的信息的价值更容易确定。

对于分布式处理来说，重要的问题是不能明确什么是有价值的或者哪里有价值。安全系统的目标是提供安全控制，但是其成本应该同没有控制的情况下的预期损失的价值相同。任何安全问题的缺乏都不会表明风险的缺乏。许多组织在明确他们需要的保护的对象以及这种保护所避免的风险之前，就把重点放在解决安全问题的技术上。你可以通过实现控制、把风险分给其他人或者自己假设风险来避免风险。这里的底线是在实施保护之前一定要首先明确需要保护的对象。

2.2.4 数据分类

你清楚地知道什么业务信息应该是秘密的以及什么不应该是秘密的吗？不同的信息有不同的机密性和可靠性需求，这是构建安全系统需要解决的一个主要问题。许多组织没有花费大量的时间来确定各种信息对于业务的重要性。这会影响到实现的安全度量或者如何应用安全。如果没有一个分类系统，那么限制安全机制可能会应用到一切事物上，可能限制工作效率；也可能是使用了非常少的安全措施，从而增大了风险。在实现安全机制之前，我们必须要明确对于组织来说什么是重要的以及为什么重要。数据的安全性和可用性将决定需要什么样的安全度量以及这些度量应该如何广泛地使用。

2.2.5 单一登录

单一登录的问题是客户最常提到的问题。登录到计算机系统上的用户需要使用用户ID和密码作为识别和认证形式。如果要访问几个系统，那么可能需要登录到许多不同的系统上，其中每个都需要用户ID和密码。事实上，安全系统越复杂，如何记住通过它来访问真正的应用问题就越大。用户会绕过该问题，或者找到可以提供附加访问的替代方法。

即使最尽责的人也可能会无意地导致最健壮的安全技术毫无用处。在有些办公室中，终端或者PC上会贴着几个黄色的小便签，上面写着各种登录顺序、系统ID以及应用密码，这样做是为了避免用户忘记这些信息。你注意到过这种情况吗？如果用户忘记或者混淆了密码，那么工作效率会受到影响。这可以通过向系统管理员求助以重设密码。这里的挑战是如何在不需要用户必须通过几道关口才能访问分布式应用的前提下提供所需的安全性。

2.2.6 有人闯入吗

对于分布式系统来说，最让人头痛的问题之一是探查是否发生了安全问题。如果一个夜贼光临了你家，可以很容易知道。如果你的房子非常安全，那么必然会留下一些强行进入的

痕迹；抽屉可能被翻得乱七八糟，或者房子里丢失了一些东西。不幸的是，如果有人像夜贼那样光临了你的计算机系统，就不容易知道了。这种偷窃行为的目标是你的信息资产，夜贼可能只是弄走了一份拷贝而原件丝毫未损。只有系统能够对访问信息进行审计，或者使用了特殊的检测机制（如房间内安装了摄像机），那么你才可能会有未授权登录的证据。

如果锁紧大门，大多数计算机安全系统都会拒绝非法访问；但是如果发现有人通过窗户进行访问，那么系统就有麻烦了。许多密码和用户ID都被安装在系统中或者网络上的特殊工具截获，但你根本不知道这是谁干的。这类似于在蜂窝电话会议中缺乏安全性。任何人都可以听取会议内容，并且你没有办法知道所发生的情况。蜂窝电话会议安全问题是很容易理解的，但是若要保证计算机通信的安全性则成本很大。尽管如此，还是值得那么去做。

2.2.7 远程访问

廉价的个人计算机的可用性能以及相对高速的通信网络使得许多员工有能力从家中访问公司的计算机系统。在某些情况下，这可以使员工在正常工作时间之后进行工作。这通常意味着员工可以使用某种形式的工具来“拨号”到公司的系统上。然而，允许这种形式的网络入口也会使得其他的非公司员工来尝试进入网络。可以用许多技术方案来提供所需的安全程度，并保证只有授权用户才能通过这种途径来访问公司网络。

通过对公司网络的远程访问，“在路上”的员工也能够访问公司数据库，不在办公室的员工可以通过电子邮件来提高其通信能力和工作效率。许多安全解决方案依靠以下事实，访问的发源地已定义并可预测。而“在路上”进行远程访问的情况不符合该事实，这种情况需要能够提供身份认证的其他安全方案。

对远程访问的需求不仅仅是技术安全问题，它还是政策问题。对公司网络的远程访问也可能会提供进入其他网络的入口。这个事实是非常重要的，可能会成为一个问题，例如，如果你或者你的孩子希望在家中通过公司的访问点来上网的话。为了上网，有些员工可能会在管理人员一无所知的情况下把调制解调器和软件安装在办公室的PC上，从而可以容易地从家里访问公司网络。

2.2.8 安全问题来自何方

回顾安全事故，了解大多数问题来自哪里是非常重要的。大多数人以及很多新闻认为安全事件和损失的来源是罪恶的黑客，但是到目前为止，同安全相关的事件中最多的一部分源自组织内部。超过半数的损失都是由安全问题引起的，罪魁祸首或是现在的或是原来的员工。这样的问题大多数都是由错误或者疏忽引起的，肇事者大多是不满的员工而不是恶意的黑客，但是黑客也是非常值得注意的。管理部门已经逐渐认识到这一事实，并且把内部环境同外部环境一起进行防范。

2.2.9 网络连接

在第1章中，我们提到过对安全问题的推动力之一是计算机网络的可用性以及实现。我们认为，对于将来成功的业务来说，网络的互联是最重要的因素之一，同时它也是最大的潜在安全风险之一。网络正用来支持过去通常在组织之外的访问。工作承包给转包商。公司之间互相进行业务往来。业务处理是由外部公司进行的。在有些情况下，竞争者已经把他们的网

络连接到了公共的第三方，甚至他们自己。

由于这些互联需求的全球性质，问题会继续增多。如果不同的组织之间已经互联了计算系统，那么如何在它们之间提供安全？如何保证所有对网络和系统的访问都是经过授权许可的？你使用什么样的安全基础并且如何对其进行管理？另一个组织的安全实践会对你产生什么影响？所有这些因素迫使提出安全问题并加以解决。

2.3 十大问题

根据我们的观点，下面的十大问题（关于信任和安全）是使用分布式计算的组织需要面对的。但是其排名与其重要性无关，我们认为它们都是非常重要的。

1) **达到正确的平衡**。计算机安全不会为任何人（厂商和顾问除外）创造金钱。它是使用计算机进行业务处理的成本。在安全的需要和系统的有效利用的需要之间必须要取得一个平衡。通常情况下，没有任何商业企业会拥有使其计算环境绝对安全所需的全部资源。

企业的风险必须要同防备这些风险所付出的成本取得平衡。其他的需求，如使用方便性和系统性能，也必须要同系统安全进行权衡。从前面可知，我们不愿意实现一个会降低工作效率的安全系统。我们也不愿意在安全方面投入的金钱比受保护对象的自身价值还要高。

2) **脆弱的认证**。认证是大多数安全机制的基础。我们需要证实用户或者系统确实是他们所说的用户或者系统。当前的认证方法依靠密码或者网络地址来进行认证。密码通常在局域网或者网络上以明文的形式进行传输，任何能够访问该局域网或者网络的人都可以很容易截获密码。网络地址也可以伪造。我们需要更严格的认证方法，它必须能够防止发现明文密码，并且不依靠网络地址。

3) **管理工具**。数据处理的一个重要规则是，如果它执行，那么管理它。对于使用多厂商产品的分布式环境，综合管理工具是非常缺乏的。公共认证、访问控制、审计以及探查需要集中式工具。使用多种产品和技术的环境需要这些工具来支持。

4) **因特网**。地平线上有了新东西。它或者是灿烂的阳光，或者是一团黑暗，这完全取决于你的观点。上网现在是谈论和出版的最多的主题之一，它可能是印刷术问题以来最伟大的事情，也可能是文明结束的开始。通过上网，人们能够访问到大量的信息，并且可以互相进行频繁的接触，这是真正值得注意的。

通过因特网，全世界都可以访问你的系统，而你可能会在不知不觉的情况下受到危害。在本书后面你会发现，网络访问带来了一些安全挑战和麻烦。如果要连接到因特网上，必须要进行仔细的考虑并进行正确的预防工作以保证你的系统安全不会受到侵害。

5) **网络上的最脆弱点**。公司网络上的最脆弱点将会是第一个遭受攻击的地方。一旦一个脆弱的系统被攻破了，那么该系统会被用做对更关键的系统和应用进行侵害的基础。你需要保证所有的系统都符合已定义的标准，并且需要针对所有系统的审计来提高企业的安全级别，或者把脆弱系统同企业网络分离开。网络的安全性只取决于其上最脆弱的链接。

6) **不同的技术**。问题不是我们没有足够的技术，而恰恰相反，是有太多的技术。标准的缺乏阻止了对问题全面解决方案的接受。或者有大量不同的标准来处理安全的不同方面，或者是有多个标准来处理安全的同一方面。没有一个标准能够处理所有的方面，也不是所有的标准都处理同一方面。大多数安全问题的解决都有许多非常先进的技术可用，但是这些技术之间互相难以协调，并且它们都难以管理。

7) **物理访问**。如果一个未经授权的人能够对计算机系统或者网络设备进行物理访问，那么系统就可能会受到非法访问。由于大多数的计算机事故都归因于内部员工，所以物理访问仍旧是一个需要注意的重要问题。对连接在网络中的一台PC上的磁盘的简单访问就足够提供这种容易的访问。

8) **不适当的策略和程序**。许多组织有来自大型主机环境的安全策略和程序，或者根本就没有任何策略。在分布式计算环境中，员工的责任和对信息的访问都要比在大型主机环境中所预想的要多。分布式计算环境需要新的策略和程序以适应环境的改变。

9) **教育**。今天计算机已经渗透到了每一个方面，犯罪技术和安全技术同时都有了很大发展，但不知哪种技术发展得更快。普通的系统管理员可能已经跟不上技术的变化。许多安全漏洞被利用了，只是因为黑客要比组织内的任何人都更明白这些漏洞。现在，许多专家黑客要比一个组织内的任何人都更了解组织内硬件和软件的复杂细节。

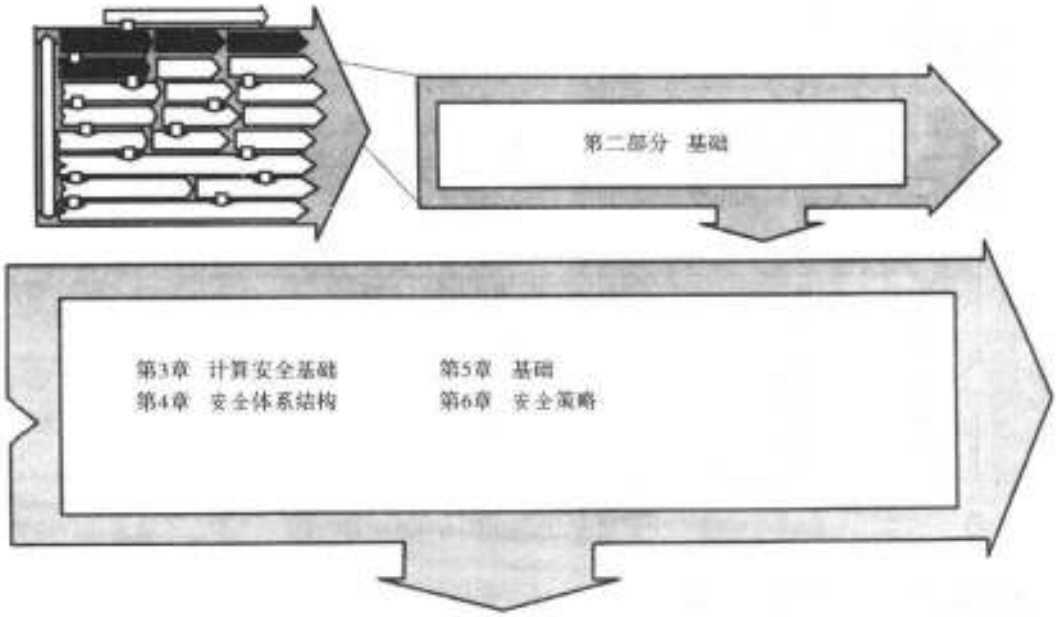
10) **缺乏计划**。对于一个组织来说，移动到分布式计算环境会对当前安全体系结构引起许多改变。可能一个企业所犯的最大错误是对新环境没有制定计划。要知道，根据计算技术界的直接需求来定义环境是非常错误的。计划的缺乏会导致互相冲突的控制和不可管理的解决方案。甚至更糟，很多安全大门可能没有关上。

2.4 结论

当前，对安全问题没有一个万能的解决方案。操作系统和网络系统的本性注定了安全问题的存在。对我们的工作起到帮助的东西也使我们暴露于众。当计算机互相还难以通信时，安全问题几乎没有人关心。在分布式环境中，用于一个系统的安全级别可能不适合另一个系统。当这些系统互连起来以后，会发生什么情况呢？在大的中心系统中，安全保护通常由技术和技术上的含糊性来提供。在现在的分散世界中，我们需要一套一致的法律和一个公共的蓝图来提供同一级别的保护和信用。

全球化商务对组织间的通信提出了新的要求，同时它把众多的内部网络互相链接起来。对计算机系统的攻击还会增加。组织必须要采取措施以保证它们自身不会成为攻击目标。在下一章中，我们将讲述一些安全基础，定义一些术语，并提供一些用于安全系统构建的基础。

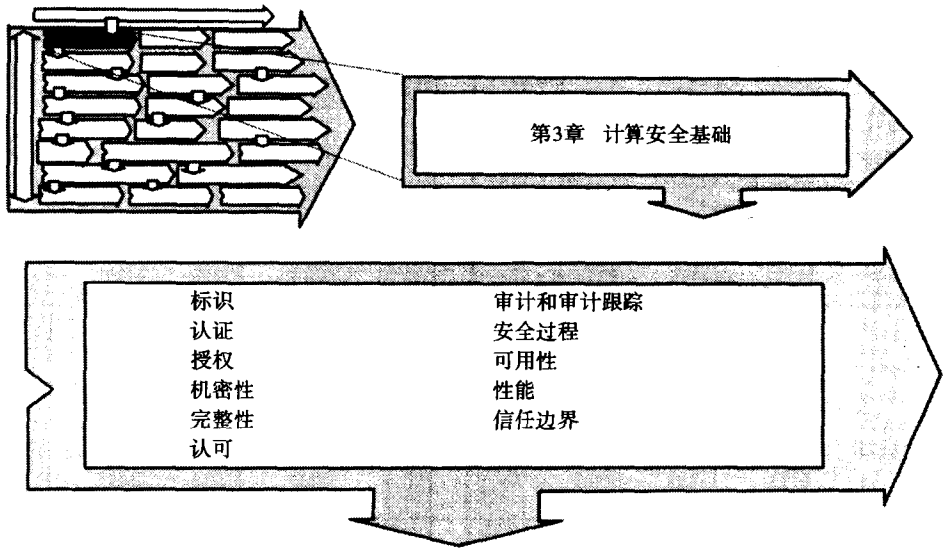
第二部分　基　　础



复杂问题的解决需要一个正确的基础。谁也不会在一个不牢固的地基上建造自己的家。否则，墙壁裂缝、地板塌陷或者其他问题是在所难免的。同样，如果没有正确的基础，那么计算安全问题的解决也就成了“水中月，镜中花”，对其时间和资源的分配也就根本无从谈起。

在第3章，将分析计算安全的基础，其意图是介绍一些贯穿本书的基础概念和术语。接下来的一章将检查计算安全的基础，它会讲述安全问题的几条基本原则，并介绍组织对安全问题所采取的不同方法。第4章则介绍计算策略和策略的主题。在第5章，将介绍一个用于分布式计算环境安全问题的体系结构，或者称为模型。有两个主要原因可以解释为什么组织会有一个安全体系结构。第一，这使得组织可以使用一个可视化的模型来说明多种安全组件的相互关系，如认证（例如，你是谁？）和授权（例如，你被允许进行什么操作？）之间的关系。第二，这有助于把安全体系结构同整个的计算机体系结构结合在一起。使用一个可视化的模型有助于确定体系结构需要处于的位置。本部分的最后一章处理计算安全策略的建立。安全策略提出了可接受的用户管理和管理职责。安全策略应该为计算安全问题的解决提供充分的指导和高级管理支持。

第3章 计算安全基础



如果对计算安全问题进行讨论，那么需要公共的术语和概念来描述安全系统所涉及的事物。计算安全所带来的基本挑战之一是简单地标识谁需要在计算系统中做什么。我们需要使用一个方法来证明系统的用户确实是他们所说的用户。当一个用户或者实体请求一个动作时，我们使用他的标识来判断他是否授权。在本章中，将介绍几个用于安全组件基础的定义和概念。信任是一个非常重要的概念，它是我们同邻居、朋友以及社会进行交往的方式。当应用到计算系统时，它也是一个非常重要的概念。本章将详细讨论信任概念对于计算来说具有什么意义以及为什么说它很重要。

3.1 安全的概念

任何安全系统的目的都是保持秘密的能力。这一点对于人和自动系统都是正确的。下面我们将看一下对保持秘密的需求以及帮助我们实现该目的的方法和技术。当存储信息或者在网络上传输信息的时候，保持信息的秘密是非常重要的。在大多数情况下，当前通过网络发送的信息同蜂窝电话具有同样的安全性。使用正确的设备并选择合适的时机，任何人都可以对你所说的内容进行监听。所以，你需要使用一个方法来保证你所说的内容只能被你的谈话对象所听到。

X/OPEN对信息技术安全所下的定义是："IT安全是IT系统的状态。在该状态下，通过采取适当的措施，IT系统上的应用风险降低到了一个可以接受的程度。IT安全的目的在于保护资产免遭威胁"[XOPEN1]。X/OPEN是一个由多家公司组成的非营利联盟，它专注于开放系统的发展。安全需要来保护资产（大多数情况下以数据的形式存在）免受威胁，从而保证数

据的正确性并帮助用户信任系统。安全是由完整性、认证、访问控制或者授权、机密性以及认可所组成的。所有这些主题都需要分析。

3.1.1 标识

你是谁？安全系统最基本的元素是标识当事人是谁的能力。在授予系统访问许可之前，我们需要知道你是朋友还是敌人。如果在酒店舞厅里有一个私人舞会，那么我们可能需要正式请帖或者客人清单来判断谁可以参加该舞会。在计算机系统中，需要某种方法来识别个体和检查客人，从而可以知道什么时候来了一个入侵者。在我们有能力判断他们是否是他们所说的之前，首先必须知道它是谁。一个更重要的问题可能是我们使用什么来进行标识。使用如用户ID之类的指派名、如指纹之类的生物特征，还是使用已定义的位置来进行标识？这种标识应该是唯一的。两个人同一时间内不能使用同一个标识。

3.1.2 认证

证明你是你所说的。一旦我们有了请求者的标识，下一个最重要的计算安全系统元素是证明一个人确实是他所说的人的能力。认证是标识的证据。这个证据可以建立在多种事物之上。对某个人进行认证的最常见方法是看看他是否知道只有他才应该知道的秘密。在计算机系统中，证明标识的最常见的形式是使用密码。在进行任何操作之前，安全系统需要证明你知道这个秘密。通常，这称做第一功能认证——你所知道的某样东西。

认证的另外一种形式是看一个人是否拥有只有他才可以拥有的某样东西，其形式可以是个惟一的卡、磁盘或者特殊的令牌。通常，这称做第二功能认证——你所拥有的某样东西。一种更安全的认证形式是建立在某个只有你才拥有的物理特征或者特性之上的。这可以是指纹或者拇指印、视网膜扫描、语音印迹或者其他的个人特征等。这称做生物认证或者第三功能认证——你是某样东西。

认证有许多不同的方式，并且有许多支持技术可以应用到指定的需求中。认证的强度需要定义什么类型的功能或者技术应该实现。对于办公室中的大多数人来说，一个简单的密码可能是非常有用的。当这些人进入到办公室的复杂环境中的时候，他们已经通过了某种形式的安全屏蔽，他是否由由其他员工所检查。第二功能认证，如一个安全令牌、一个点子密码生成器或者其他设备，可能需要来控制对办公室内个人区域的访问。使用一个安全令牌也是提供对网络进行远程拨号访问的流行方法。第三功能认证只在最安全的环境中实现。

另一种提供认证的有趣方式是使用位置做为请求的测量之一。全球定位卫星（global positioning satellite，GPS）系统能够为我们提供全世界范围内的一个精确位置，其误差在几米之内。我们可以使用来自GPS系统的位置信息做为参数来生成位置签名，然后把此位置签名同所预期的位置签名进行比较，从而确定它的真实性。例如一个位于加拿大境内的银行和一个位于巴西境内的银行，如果二者之间的一笔事务处理产生于欧洲，那么该事务处理就值得怀疑。

3.1.3 授权

你可以做什么？一旦我们知道并且能够证明某个人或者某个东西确实是他所说的（那个

人或者那个东西），那么我们需要确定这个人可以做什么。授权依赖于对用户或者请求的成功标识和认证。授权是把权限授予用户、程序或者进程。我们使用授权来定义用户可以看见什么、使用什么以及在哪里进行处理。系统中有很多资源，以及很多针对于用户或者进程（已认证）的服务，对这些资源和服务的访问要有选择地授予用户。取决于所需的授权级别，这些权限会按照许多不同级别来进行授予。

授权可以明确授予，也可以明确拒绝。如果我们明确拒绝访问，那么任何人都可以访问信息，但那些我们希望保持机密的信息除外；如果我们明确授予访问，那么除了我们授予访问许可的信息以外，其他任何信息都将不能访问，这种授权可以定义在系统范围内，或者取决于技术，也可以限定在具体的数据元素上。我们也能基于访问类型进行授权。访问类型一般就是读、写、创建、更新、删除以及执行。

3.1.4 机密性

你能保守秘密吗？保守秘密的能力也是一个非常重要的元素。机密性就是保护信息免遭未经授权的泄漏。一般有两种方法可以用来提供机密性。一个方法是限制对有限个体的访问来限制对信息的访问。我们修建一个高且厚的院墙来阻止任何人进入院内，甚至连偷看也不行。只有那些具有正确权限并且已严格认证的人，才可以进入院内。如果有一些可以绕过院墙溜进院内的途径，那么信息的机密性就会受到损害。

另一个方法是使用一个只有我们才知道的特殊代码来打乱信息。这就是加密。我们打乱或者加密信息来隐藏信息，当需要使用这些信息的时候，再对信息进行还原翻译或者解密。只有那些知道打乱代码或者密钥并且知道如何使用它们的人才能够访问这些信息。一般来说，这种方法要比修建一个大院墙更能提供机密性。即使有人费力爬进了院内，里面的信息也不会有危险。为了能够提供机密性，加密通常看做是任何安全系统中的必需组件。使用加密来保持机密的能力是建立在两个因素之上的：第一个因素是加密算法的长度；第二个更重要的因素是保持密码安全的能力。

3.1.5 完整性

你得到我发给你的消息了吗？完整性是保证消息或者数据的安全并且不会修改的性质。完整性检查可以帮助检测系统、应用程序、数据以及网络的未授权使用和修改。提供完整性检查的一个方法是在发送或存储之前先在数据或消息的结尾处附加上一个特殊的指示符或者消息摘要。该消息摘要是对消息内容进行散列过程所产生的一小段数据。对消息内容的任何更改都会产生一个不同的消息摘要值。在消息发送或存储之前，摘要本身也可以加密。

当提取数据时，或者在接收数据时，同样的过程还要执行一遍。如果所计算的消息摘要同追加在消息后面的摘要相匹配，那么消息肯定没有篡改。如果二者不一致，那么在消息发送或者存储以后，消息肯定更改过了。有许多方法都可以用来提供完整性。

3.1.6 认可

你确实得到了我的消息！认可是防止对消息的发送、接收或者进行处理的否认。它证明一条消息确已发送和接收。它保证消息的发送方不能否认消息的发送，消息的接收方不能否

认消息的接收。认可是建立在惟一签名或者标识的基础上的，该签名或标识可以证明谁创建了消息或者信息，以及该消息或信息发生了些什么事件。如果有认可的话，那么可以证明一个人同一个动作或事件有关联。

下面是一个真实的故事，它可以非常好地说明认可的必要性。在未经敌国的允许之下，某个国家在丛林深处进行了一个使用多种监听设备的隐秘军事行动。该行动意在监听该国境内的各种通信。来自司令部的一个紧急编码消息突然打破了平静。该消息迅速解密并认证，然后得到的结果是一条命令，即立即离开并破坏该行动。该消息包含了一个特殊的鉴别码，来证明该消息只能来自司令部中的一个已授权个体，这是由一个特殊的过程来验证的。该消息只能由那个已被授权那么做的人来发送。

作为该命令的结果，当“特殊的红色按钮被按下”时，破坏过程开始了，丛林里乱作一团。建筑物里的所有设备都上好了炸药。当人员撤退好以后，所有建筑物和所有设备都被炸得粉碎。接下来的那些天里，士兵们藏在丛林中等待援救。但是根本没有人来，所以肯定出了什么差错！

真相很简单，即消息送错地方了！该军事行动是不应该终止的，但他却收到了一份本应该是一个需要中止的行动收到的消息。该故事很好地说明了不仅仅消息源需要认可，而且消息传送也需要认可。在该故事中，消息源没有问题。如果司令部要求接收方对消息的传送进行确认，那么这次灾难就可以避免。

3.1.7 审计和审计跟踪

审计似乎是那些通常出现在安全系统中的必要的罪恶之一。审计是一个可信赖的、博学的以及独立的个体积累并评价事实的过程，其目的在于报告要评论的对象同已建立的标准之间的符合情况。计算机审计的目的在于根据已经建立的安全策略、原则或者工业标准来评审系统或者网络的性能。我们需要具有这样的能力，从而可以跟踪并查阅可能会看做是安全问题以及可能会导致安全问题的所有情况。

审计跟踪是组织列表，通常其由事件所发生的时间来维护。这些事件可以通过对信息的访问、系统资源的使用或者同安全机制的交互来触发。如果我们成了一个未经授权的入侵者的目标，那么我们需要理解该安全危害的影响，其方法就是查看什么信息被访问了以及系统是否不正确地使用了。查看审计跟踪能够帮助我们做到这一点。一个好的安全系统应该具有健壮的审计功能和审计跟踪。

3.1.8 安全过程

现在，可以使用上面这些安全组件了。图3-1说明了这些组件使用的地方。我们假设一台个人计算机，称为client（客户机），是系统的主要入口。该机器需要连接到另一台称为server（服务器）的计算机上以执行事务。客户机和服务器都位于同一个domain（域）中。

当客户机上的用户提交一个登录请求时，他需要提供用户ID和密码。在送往服务器进行认证之前，该用户ID和密码应该首先加密以保持机密性。服务器使用密码来对用户进行认证。在服务器上，密码应该总是以加密的形式保存。如果用户ID和密码是有效的，那么登录请求会生成认证响应。根据用户认证标识，如果用户正确授权，那么接着就会执行一系列的事务。如果需要，事务消息可以通过使用追加消息摘要和数字签名来包含完整性和认可组件。

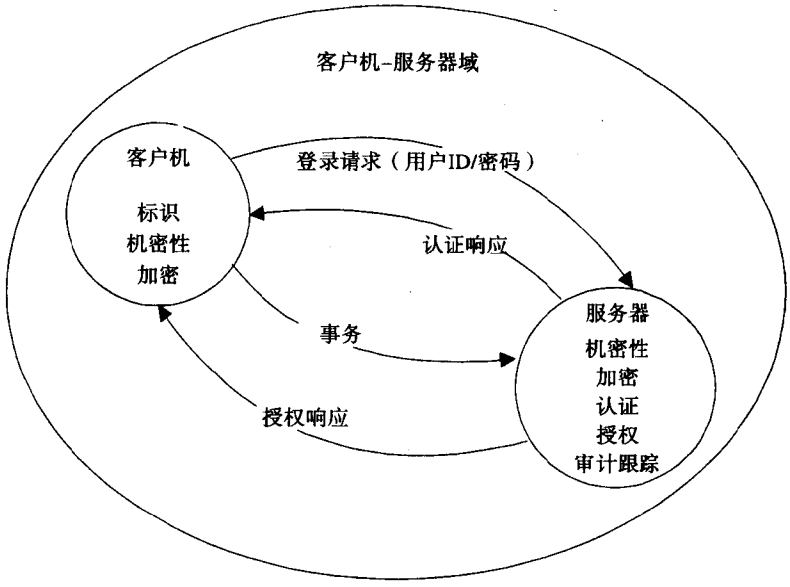


图3-1 登录过程

3.2 信任——信任的概念

假设你的业务处理系统分散在多个处理器上，而这些处理器并不是全在你的控制之下，那么为什么应该信任该系统上的业务处理？除非组成这个系统的所有组件都符合你所要求的信任级别，否则你不会使用这个系统。只有所要求的所有属性都包括在系统中时，这种信任级别才会建立。根据个人观点和具体需求的不同，满足所需级别的组件和属性可能会有很大的差别。对于同一系统组件来说，由于不同的原因，不同的人可能需要不同的属性。

房屋建筑者总是希望房屋的设计简单并且容易修建。他们愿意使用高质量的材料以及好的工具，从而提高其时间和技术的使用效率。最终完成的房子应该是没有瑕疵的，并且任何需要的变化都能够容易且廉价地实现。为了销售，地产商所希望的是房子能够具有大量的功能，并且出自著名的建筑师之手。买方希望房子应该是可靠的、安全并且价钱公道。所有这些需求是不同的，但是同一套房子必须同时满足它们。当每一方的特殊需求都被满足了的时候，信任就建立起来了。

信任可以定义为对完整、真实或者公正等方面的相信。信任所指的是应用程序进行完整性操作、保持机密信息的秘密以及持续工作的能力。负责计算系统设计、部署和管理的人必须认识到并且能够对如下问题做出回答：在系统可能是分布式的情况下以及系统和控制可能处于更大危险的情况下，如何维护系统的一个高信任级别？一旦丢失了信任，将需要进行大量的工作和说服以重新建立信任。

对于任何自动化系统，你可能会问：为了使我们相信它，系统必须提供什么？我们使用下面的模型来解释计算系统中的信任组件，它由三个主要部分组成：安全性、可用性以及性能。安全是由安全构建部件组成的，这包括认证、授权、完整性、机密性以及认可——这些都是我们刚刚讨论过的。可用性依靠那些用于保证系统总是可用的功能和属性。另外，需要

可接受的性能来防止失败和潜在的绕过控制行为。如果在信任可保证之前这三个因素都满足了，那么我们就有更大的信心来得到一个可信环境。图3-2说明了构成一个可信系统主干的组件和属性。

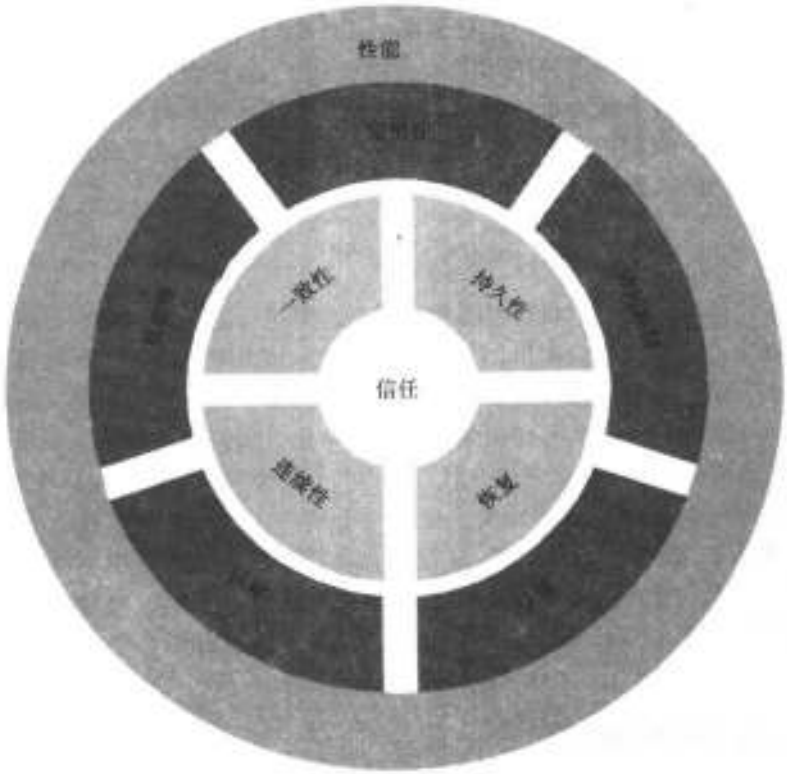


图3-2 信任模型

3.2.1 可用性

信任所要求的第二个特点是可用性。用户需要能够确信刚输入的事务会立即执行。有一个未记入文档的但却得到普遍认可的系统失败级别，它定义了实际响应时间同所期待时间之间的关系。如果两个时间非常不一致或者不稳定，那么该失败级别就会根据二者之间的差距进行指数增长。如果系统在需要使用的时候不可用，那么你的工作处理就会受到影响。即使系统并不正确进行业务需求所需要的工作，在需要的时候它也必须要处于可用状态，并且还要能够在合理的时间内完成任务。可用性包括：

- 持续性——在物理组件出故障的情况下，能够经受一次完全服务中断并随后进行恢复的能力。
- 持久性——在分布式系统的一些物理组件出故障的情况下，能够经受一次部分或者平缓服务降低的能力。
- 恢复——在没有任何人工干预的情况下，从一次完全服务中断中进行恢复的能力。
- 一致性——在给定相同数据和处理标准的情况下，能够产生同一结果的能力。分布式处理必须是一致的。

3.2.2 性能

可信环境的第三个组件是性能。为了接受和使用计算系统，它必须能够在所需的时间范围内尽快地完成请求。然而，如果使用计算系统会为你的工作带来负面影响或者手工处理任务的速度更快，那么你可能不会使用这种系统。这方面的例子有很多，如一个需要多次输入密码的认证过程，或者一次需要使用几个处理器——这些处理器是通过一个慢速网络进行连接的——查询。使用计算系统不应该降低工作效率。安全系统和相关安全策略的设计应该考虑到交付一个合适的、可信的系统所需要的实际权衡因素，这些系统才能工作正常并且能够同它需要保护的资产的价值相吻合。

3.2.3 信任边界

在任何分布式计算环境中，我们都需要对信任边界进行定义。如果我们居住在一个具有上锁入口或者安全走廊的大厦公寓里，会认为任何进入该大厦的人都是已授权这么做的。通过第一道测试——前门以后，我们赋予他们某个信任程度。我们也需要定义分布式计算环境中的信任边界。不管我们在什么时候把信息进行分布化，我们都希望信息分布到的地方总是安全的。授权的范围有多大？什么时候、什么地方我们需要对认证和授权再次进行检查？

上面这些问题同信任边界密切相关。做为一个用户，我们希望信任边界能够包括在业务处理期间可能与之交互的所有一切。这意味着我们不应该必须使用不同的密码登录到几个系统上。从安全角度上看，我们希望信任边界能够变得非常窄，只允许那些具有明确授权的人访问系统和信息。信任边界必须能够促进工作效率但却不会引入太多的风险，这正是它的范围。

3.3 信任——需要信任的原因

当只使用一台计算机时，可以相对很容易地检查出系统是否按照预期情况进行工作。如果系统工作不正常，那么问题的根源所在无非也就几个地方。获得信息或者更新一个文件之类的操作通常都是很简单的。然而，对于复杂的系统，其中可能有多个文件需要更新。在这种情况下，系统需要一个机制来处理更新一致的问题——如果一个文件更新失败，那么多个文件都要回滚到操作发生前的那一时刻。如果一个分布式计算系统跨越位于多个地点的多个计算机之上，那么你可以想象一下这种情况下计算机程序执行和文件更新的复杂程度。系统产生你所需要的结果吗？如何判断执行成功与否？如何把所有的处理都回滚到同一时刻，就好像错误根本没有发生一样？信任永远不必疑问系统或者数据的状态。

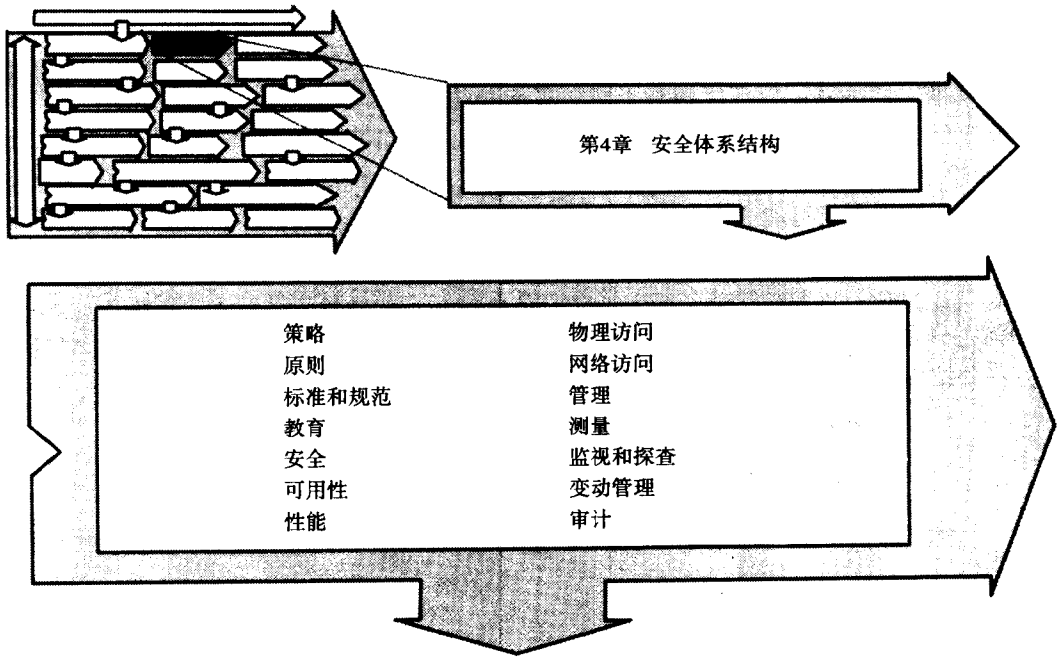
当前，有三个因素制约着分布式计算系统的发展：可用的网络带宽、健壮管理系统以及完全分布式安全系统。为了能够把一个计算过程的任何部分都放到位于使其具有最佳意义地点的计算机上，我们需要具有足够的网络通信能力以保证使用网络不会造成瓶颈。足量的网络带宽是一个成本因素而不是一个技术因素。如果调整了成本，所需的网络带宽是可用的。系统管理问题也是一个亟待解决的领域。标准系统管理组件并不适用于所需的所有平台，以及各种不同的分布式系统实现方式。安全组件标准在定义和实现方面进展缓慢，它们必须满足分布式系统的需求。我们有能力把数据和处理分布化，同时不需要提供健壮性来对它们进行管理和安全保证。

3.4 小结

信任应该一开始就设计到计算环境中，而不是后来再添加。如果有信任问题，那么一个系统可能很少有人会买，除非用户的需求就是要那样。一个可信的分布式计算系统需要很多安全机制，而不是仅仅够了就行。设计一个分布式计算系统有很多方法。保证一个可信系统所需的组件应该在系统设计阶段中进行选择。一个可信计算环境必须包含符合所需信任级别的安全、可用性和性能组件。

什么时候你知道可以完全信任一个系统？不幸的是，信任没有一个绝对的测量基准。即使有一个基准，同一系统信任级别对于两个实现来说也可能有所不同。信任永远不会太多，但多少是足够？对于一个银行或者空中交通控制系统来说，其性能和可用性需求可能要比一个库存系统的需求更加严格。银行系统对精确性和效率有很严格的需求。根据系统用户的数目，可用性和安全需求可能会有所不同。如果你家屋后的雪地上有脚印，你会关心吗？如果你的房子靠着一个运动场，你可能不会关心；但如果你的房子位于一条乡间小道上，你可能会非常在乎。现在，我们已经建立起了信任的基本概念，并且对安全有了一个定义，下面将讲述如何在分布式计算中使用这些组件。解决计算安全问题的一个挑战是围绕这些安全组件来开发一些结构并且开始构建一个安全体系结构。

第4章 安全体系结构



前面我们已经说了几个故事，并且提出了一些问题，现在你还想同分布式环境打交道吗？根据我们已经掌握的知识，现在首先来设计一个安全环境。我们已经定义了什么是分布式处理系统，并且涉及到了一些问题，现在要把这些知识综合在一起。我们需要掌握同分布式计算安全有关的一切知识。在集中式系统中，安全可以主要由技术应用来提供。当系统分布到多个平台以及多个地点上时，管理安全组件就变得非常困难。下面的内容将提供比较安全过程和分布式平台组件的参考框架。这些组件逻辑地组合在该参考框架中——我们把该框架称为安全体系结构。

> 坚固持久的建筑物都是建在一个牢固地基上的。对此地基，我们首先需要一个计划，从而使得现在和将来所需要的一切都能支持。你也可以在问题出现之后再给地基添加东西，但那将是一个浩大的工程！

将体系结构定义为实现以下功能的虚拟实体：

- 提供未定义环境的概念上的定义和结构。
- 在环境内可以设计独立组件。
- 说明独立组件应该如何集成进整体环境中。
- 保证完成后的环境符合最初建立的虚拟实体。

安全的目标是降低企业的财政风险和由各种情况带来的损失，如入侵、系统误操作、特

权滥用、数据篡改、欺骗以及服务中断等。外部威胁（如黑客攻击、外敌攻击以及犯罪行为等）和内部威胁（如不稳定的或者肆无忌惮的内部员工破坏等）都需要加以防备。另外，为了允许信息技术的电子化高效应用，安全基础结构不可以禁止先进信息服务的使用。安全体系结构的目的是产生一个模型，该模型可以提供所需的安全组件以及用于比较多种机制和方法的基础。

安全体系结构同各种元素协同工作以保护信息的机密性，并且保证所有对计算资源的访问都是授权和认证的。安全体系结构的具体目标是提高分布式应用和信息的完整性、一致性以及机密性。其总体目标是让分布式环境成为可信的。欲达到这些目标，需要一些机制来提供认证、访问控制或者授权、完整性、机密性和认可。在安全能够真正落实到位之前，所有这些属性都必须要得到保证。我们需要能够信任用户访问系统的所有地方，而不是仅仅信任信息和工具所在的地方。

安全体系结构重要吗？这是一个老生常谈的问题。没有一个安全体系结构不会导致系统不可用或者立即被破坏，但是它会使系统变得很脆弱，一个误操作可能就会把系统搞坏。例如一个由几个业务单元所组成大型组织，其中每个单元都有自己的安全机制和支持系统，如果每个单元不遵从一个公共的安全体系结构，那么从一个系统到另一个系统的安全访问会是很困难的，甚至是不可能的。每个系统可能有所需要的安全，但是各个安全系统之间可能互相不兼容，当需要进行业务单元间的访问时，这种不兼容性就会导致问题。如果一个业务单元的安全需求要比另一个单元的高，那么问题尤为突出。

一个良好定义的安全体系结构能够保证应用程序和系统的设计符合所需的安全目标。安全体系结构能够帮助指导系统和平台的选择，并且保证所有的系统都符合一个标准的最低安全级别。把一个安全体系结构应用到已经设计好或者建立的系统上是非常困难的。安全体系结构并不依赖于系统体系结构的存在。一个良好定义的安全体系结构并不涉及到技术。

对于一个体系结构，其中一个常见的组件是关键成功因素的定义。关键成功因素定义是为了完成业务目标而必须绝对正确的某些事物。安全体系结构中的主要关键成功因素不是对必须成功改变的事物的一个标识，而是在业务过程和结构不断变化的情况下如何维护业务过程的安全完整性。

> 把一个体系结构放在合适的地方有助于理解分布式环境。这将提供一个参考点，你可以使用它来挑出所有局部组件而从整体上把握问题；并且这会提供安全解决方案的某些一致性。

安全体系结构是由一些构建部件组成的，这些组件一起定义了综合解决方案的框架。图4-1表示了一个安全体系结构应该包含的主要组件。该模型是表示一个安全体系结构应该包含什么主要组件的方法之一，并且它说明了各个组件的相互位置关系。该模型有助于解释这些组件及其相互关系。其他模型或者表示可能是有效的，重要的是使用一个体系结构方法。

该模型分为三个主要组件：基础、信任和控制。基础是由组织定义的总体安全原则组成的。它也包括控制实现的安全策略以及安全机制的使用。如果选择了的话，它还可以包括具体的安全标准或规范。信任层定义了安全性、可用性以及性能三方面，这都是在分布式系统中建立信任所必需的。控制层概括了用来管理和控制所需安全组件的机制。

下面详细地逐一探讨这三个组件。

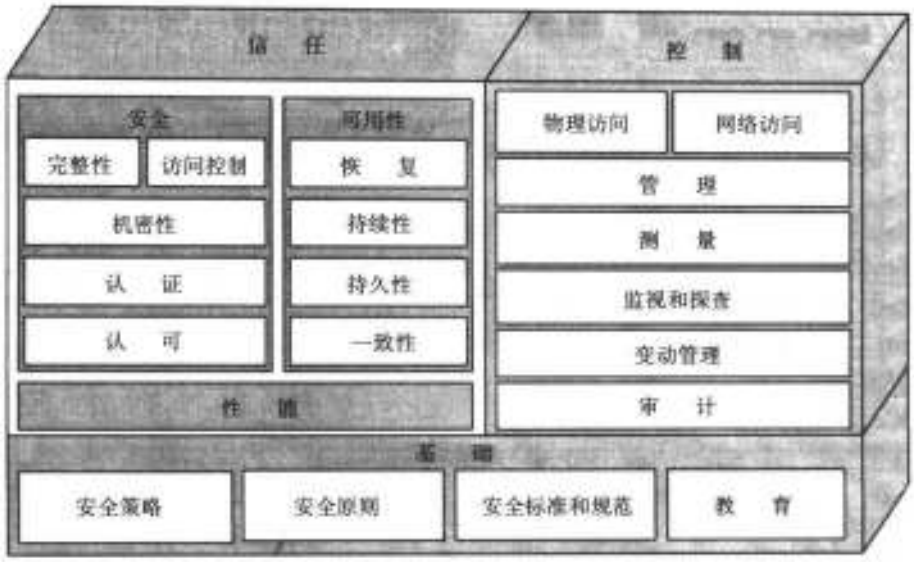


图4-1 安全体系结构模型

4.1 基础

安全体系结构的基本构建部件是基础。它由公司策略、原则的定义以及一套良好定义的安全标准和精选的规范（可选）组成。安全、分布式处理环境的建立必须要在清晰简洁的指导策略的控制下进行，并且要得到管理层的支持。这些安全策略提供了一个框架，这对在分布式环境中保证信息资产安全是非常重要的。基础的另一个构建部件是安全原则。安全原则反映了组织的思想和风格。安全标准用来指定一个建立良好的安全参考规范，该规范是可选的组件。指定安全标准也是可选的，它也作为体系结构的一个基础而必须存在。下一章将仔细介绍基础的各个组件。

4.1.1 策略

安全策略可以描述为高级管理部门用于明确组织目标和可接受过程的一系列声明。这些声明对每个独立组织都是不同的。在分布式计算环境中，安全策略是非常重要的，它用于保证所有的信息资产都会被一套可接受的控制行为所保护。策略可以设置方向、提供大范围的指导、表明管理层对安全环境的支持和承诺。组织的过程定义策略的实现过程。策略能够促进用户动作和可接受行为的一致性。

计算安全策略必须是完整的和易于理解的。安全策略的目标是高效的管理风险，定义员工对信息资产保护所承担的责任，为一个稳定的处理环境建立一个基础，保证适用法规和规章制度的一致性，并且在资产误用、损失或者未经授权之泄漏的时候维护管理选择权。

4.1.2 原则

原则定义了安全对于组织的意义及其实现方式。每个公司都有自己独特的原则。原则为指导同安全系统组件和技术相关的需求和选择提供了基础。如果公司原则说明用户是可信任

的，那么安全策略和机制应该实现为：明确拒绝对未授权资源的访问、隐含允许对其他人的访问。如果用户不是可信任的，那么安全策略和机制应该实现为：明确允许已经授权的访问、隐含拒绝其他人访问。原则定义了组织对安全问题的总体思想。原则把这种哲学思想翻译成可以应用和遵循的规则。

4.1.3 标准和规范

安全标准和规范是基础中的可选组件。一个安全标准是一个定义良好的参考规范，可以应用到安全组件和技术上。得到最广泛承认的参考规范之一是由美国国防部制定的可信计算机系统评估标准（Trusted Computer System Evaluation Criteria）。许多政府系统都采用此标准作为一个技术选择标准。

安全标准对于体系结构基础的建立也是非常重要的。具体安全标准的选择可能是对需要同组织外部进行交互的系统的需求。互连自动柜员机的使用需要严格应用和遵循安全标准。

4.1.4 教育

对于坚固基础来说，教育是一个附加组件。每个受到安全体系结构影响的人都应该通过一个认识和教育程序来学习安全体系结构。组织应该有一个认识程序来概括出组织对安全的重视和承诺，并且应该开发并维护一个正规的教育程序以辅助那些对安全机制和安全处理有责任的人。如果人们不知道安全策略的存在或者不知道他们自己对安全应承担的个人责任，那么安全策略就不会有效果或者就不会实施。对于一个安全体系结构来说，教育和认识程序要作为一个必需组件来考虑。

4.2 信任

在计算系统中信任定义为安全、可用性以及性能的综合体。在信任建立之前，这三个组件应该都是存在的，并且都应该在可接受的范围之内。信任指的是一个计算机系统三个方面的能力，进行处理保持完整性、保持机密信息秘密以及持续进行所需功能。在一个分布式网络上建立信任是很困难的，因为在这种情况下许多因素和实体都会影响到所需要的安全性、可用性以及性能的获得。如果对安全、可用性以及性能的需求满足，那么信任就是安全的。

4.2.1 安全

信任的第一个构建部件是安全。这一部分包含构建安全分布式环境所需的全部基本机制。我们在第3章中讲过，安全构建部件是由组件完整性、授权、机密性、认证以及认可几个部分组成。

1. 完整性

完整性保护机制能够保护数据免遭因为事故或者因为故意行为而导致的破坏（修改、损失、重放、重新排序或者替换）。数据可能会被故意地或者不小心地破坏，完整性保护机制能够保证任何破坏都会发现，然后进行纠正或者做出标记以示提醒。有很多方法都可用到安全性上。大多数方法都涉及到同数据进行比较的控制价值的使用。

这些方法的应用对象可以是包含在传输消息中的数据或者是存储在磁盘上的数据。消息发送方使用一个算法来处理数据，然后把结果作为一个后缀加在消息之后。消息接收方使用

同一算法来重新处理消息重新创建测试数据，然后同后缀相比较。如果二者相符，那么就表明消息没有改动。这种方法的一个更为安全的变种是在计算后缀之前在消息中先添加一些秘密数据，然后在发送消息之前删除这些秘密数据。接收方在计算后缀之前也先添加同一秘密数据。这样，后缀的创建就不会完全依赖于消息中的数据。

另一个需要安全性的地方是系统应用程序的操作。这需要一些机制来控制开发、修改和实现应用程序以保证系统在操作中的完整性。我们需要确信分布式应用程序本身不会进行未经授权的改动。应用程序的完整性应该受到专门处理该问题的安全策略的控制。

2. 访问控制

访问控制的使用为增强系统资源使用授权提供了一个方法。授权，也称做访问控制，通常是建立在标识机制和认证机制的基础上的。如果同一性得到认证，那么就会赋予对资源的访问权。在分布式计算中，确定授权的两个最常见的方法是根据用户ID（结合一个合法的密码）和已认证的系统ID。

不可能有一个授权解决方案可以适用于所有的分布式系统。授权是建立在需求目的的基础之上的。关系型数据库技术在数据记录、元素和视图级上提供了授权以及访问控制。访问控制列表（access control list，ACL）机制，比如分布式计算环境（Distributed Computing Environment，DCE）中的实现，为过程访问和数据资源提供了一个基于标准的授权机制。基于系统ID或者基于用户ID，操作系统本身也实现了多种类型的访问限制。

授权可以建立在时间依赖、数据分类、角色或者用户职能、系统地址、事务类型以及请求服务类型的基础之上。有时候在一个应用程序内，授权也在函数级上实现，其方法是使用一个用户/函数矩阵。在分布式环境中，授权机制的管理是一个主要的挑战。

3. 机密性

在计算机系统中，机密性指的是系统保持机密电子信息之秘密并使其免遭未经授权泄漏的能力。机密性的实现保证了信息不能被一个未经授权的用户或者进程所访问。保持信息之机密性的主要方法是通过加密来改变数据的形式。现在有很多可用的网络，它们本身都不允许我们对其操作予以高级的信任。数据的加密，至少是用于认证的所有数据，应该是必须进行的，从而防止未经授权的探测和破坏。如果所有的流量数据都进行加密，那么加密将是很慢的并且成本不菲。安全需求、网络和处理成本以及网络性能三者之间应该达成一个平衡。

4. 认证

认证是唯一标识用户、机器或者应用程序并校验此标识的方法。它是授权的基础，并且也是访问控制和审计的一个基本需求。认证可以建立在你所知道的某样东西（如密码）、你所拥有的某样东西（如一个电子身份卡ID）或者这些方法的综合的基础上。考虑认证应该包括为每个系统/应用选择一个独特的认证机制或者在多个系统上使用一个公共的认证机制（单一登录）、在客户机-服务器环境中对客户机和服务器都进行认证以及认证过程的完整性（例如明文密码）。

5. 认可

认可是对资源的拥有者/创建者标识的完全信任。认可指的是防止对消息的发送、接收、动作处理进行否认的能力。当有方法证明消息或者动作只能是由发送者产生的时候，我们就可以提供认可。认可有两个应用地方。源否认是对数据产生者的不信任，交付否认是对数据是否真正收到的争执。在电子介质中，认可是通过使用数字签名来实现的。

4.2.2 可用性

可用性总是需要的，但是它通常没有得到同安全性一样的重视。当考虑一个安全体系结构时，可用性应该是信任的一个重要组件。如果一个系统在需要的时候总是保持可用状态，那么你将会信任这个系统。决定可用性的因素是：

- 持续性——在物理组件出故障的情况下，能够经受一次完全服务中断并随后进行恢复的能力。
- 持久性——在分布式系统的一些物理组件出故障的情况下，能够经受部分或者平缓服务降低的能力。
- 恢复——在没有任何人工干预的情况下，从一次完全服务中断进行恢复的能力。
- 一致性——在给定相同数据和相同处理标准的情况下，能够产生同一结果的能力。

实现一个分布式系统应该考虑到可用性。对系统的未经授权的攻击可能会使可用性受到影响，并导致系统资源被毫无意义的处理消耗掉，从而影响合法用户对系统的使用效率。只有系统中使用了正确的机制来支持可用性需求，系统才能看做是可信的。

4.2.3 性能

性能是信任的第三个组件，它对于分布式处理环境来说是非常重要的。一个跨多平台的系统必须能够在用户所需的时间内提供分布式处理结果。如果系统的响应时间超过了可忍受的范围，那么用户不会依赖甚至不会使用这样的系统。一个能够完成某种工作的系统，如果存在性能问题，那么它不会使用。安全技术的选择以及安全机制的使用程度都会影响到整体性能。例如，数据事务的加密和解密会延长处理时间。如果一个系统看起来毫无响应，那么你不会信任这样的系统。你会迫切地清除屏幕然后再次尝试。

4.3 控制

为一个安全体系结构和信任机制制定了策略和策略以后，我们需要考虑一下如何对它们进行管理。构成安全体系结构的第三个部分包含了用来控制安全机制的功能。这些功能提供了监视系统之安全操作的管理和测量能力。这一部分中控制或者指导标准最少。分布式安全元素的管理和控制带来了一个管理和安全挑战。分布式安全控制和管理过程必须作为整体安全体系结构的一个必需组件来加以实现。

所需要的控制机制包括：

- 物理访问——对实际计算和网络设备访问的控制。
- 网络访问——对网络访问的控制。
- 管理——对安全机制的控制。
- 测量——安全机制的影响以及例外事件的潜在探测。
- 监视和探查——检查破坏行为何时进行的能力。
- 变动管理——安全机制变动的管理。
- 审计——用于跟踪安全事件可用信息的记录。

4.3.1 物理访问

如果没有附加的严格控制，对于那些希望对系统搞破坏的人，访问一个计算或者网络设备通常不是很困难。一般来说，随机器而来的访问控制是不够的，或者没有实现为支持设备的安全使用。附加的访问安全需求可以使用附加设备（密钥、智能卡、令牌）来实现，这样从而在物理上保证系统或者设备访问的安全。这些东西可以用来局部证明某个个体应该具有访问权，或者在某些情况下一个装置（可能要花点时间）或者智能卡可用来对用户进行系统认证。更高的安全需求可能会限制对某个特殊个体的访问，通过使用一些设备，这些需求也可以得到满足，这些设备包括指纹扫描器、视网膜扫描器、语音模式、击键模式以及签名校验设备等。

4.3.2 网络访问

网络访问控制提供了把网络访问限定给那些具有访问授权的用户或者进程的能力。当整个网络处于公司的管理和控制之下的时候，网络访问控制是容易实现的。然而，这种可信网络并不是完全安全的。对网络的访问权可以通过分接通信线路和监视流量来获得。一个可信网络并不排斥其他安全机制，如加密。当公司网络连接到公共网上或者同其他网络互相连接的时候，它必须要使用额外的控制。

4.3.3 管理

分布式系统管理是一个复杂的问题，它包括多种资源的管理、任务管理以及解决方案结构。对于系统管理员来说，一个关键挑战在于每个分布式环境的惟一性。分布式系统在许多层上都需要高效率的管理，其中包括操作系统、网络、访问控制机制、密钥管理、数据库管理、中间件以及网络操作系统。管理工具可以跨越两个或三个领域，但是很难找到一个跨越所有这些领域工具。许多领域的管理功能可能也是集中式的，但是不可能集成到一个管理单元中。

4.3.4 测量

测量工具用来分析和报告各个组件的性能状态，它也常常用来说明组件的使用情况。测量工具的使用通常并不位于安全元素的控制之下，但是它们对于探查预期行为规范之外的未经授权行为是非常有用的。我们可以使用一些技术来分析计算使用情况，以及对任何非预期的行为进行标识以指出问题区域。

4.3.5 监视和探查

实现分布式安全系统带来的另一个烦人的问题是对监视安全系统的操作和探查实际的或者潜在的安全问题的需求。该问题有两个解决办法。实现被动监视，使用一个机制来查看所有可用的信息并对所发现的任何脆弱点都进行报告。使用一个动态监视系统，主动训练并探查安全系统以寻找脆弱点。如果使用自动控制来实现，那么动态系统是最好的选择。

4.3.6 变动管理

分布式处理环境的另一个挑战是安全组件的管理和维护，这些组件包括如用户和密码列

表、访问控制列表以及加密密钥。在安全环境中，如要提供安全机制的完整性，那么上述组件需要进行变动。实现和协调跨分布式环境的变动需要一些过程和机制，这也是一个重要的需求。对于分布式安全环境来说，管理功能的安全性也必须要考虑到，因为恶意破坏者可以操纵这些功能来获得对系统的非法访问权。

4.3.7 审计

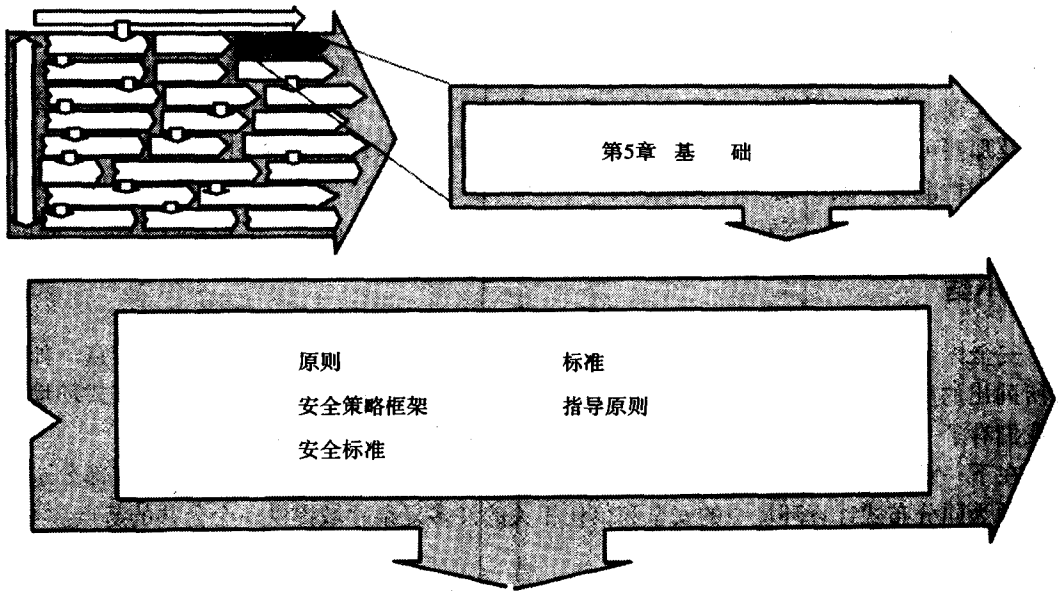
审计可以分为两个领域，其中每个都有一个不同的着重点和目的。每个安全体系结构都应该包括一个独立的和有见识的控制机制评论。这样会根据规范和已建立的标准来测量安全系统的实现。第二个领域是使用审计跟踪来跟踪用户行为、事件、那些可用于报告和控制目的以及那些可用来对一个事件进行重新构造或者调查研究的对象。审计功能的安排和操作可能会受到性能和（或）成本问题的影响。一个鉴定过程会对安全机制进行查阅以保证它们确实到位并且工作正常。在安全机制更新或者变动之后，它应该进行一次鉴定以校验其操作。

4.4 小结

安全体系结构为比较一个分布式系统的需求和多种实现机制提供了一个参考点。如果本章所列出的所有组件都以合适的方式得到实现，那么你就构建了一个安全的分布式环境。现在我们有了一个好的计划，可以放下分布式环境中的信任所需的基础。

在下一章中，将讲述安全和信任的关键构建部件。将对一个牢固基础的组成部分进行探讨，并为同分布式计算环境中的安全和信任相关的许多复杂领域提供一个公共的起点。

第5章 基 础



前面我们探讨了安全体系结构，现在看一下体系结构基础的构成（图5-1）。体系结构决定了房屋的外观和质量。基础将决定房屋的特征并为其提供支持所需结构的能力。基础也应该能够支持将来的改造和修补。在安全体系结构中，基础是由一些声明和决策组成的，它们描述了组织需要什么类型的安全环境以及该环境必须具有的属性。

在本章中，将首先探讨原则——描述组织属性和思想的声明。接着将分析控制分布式环境所需的安全策略。你会看到很多的“C2”或者“B1”之类的字眼，它们都是描述系统安全级别的方法。这些级别用来比较系统的安全质量和参考标准。我们看一下这些级别的定义以及它们作为安全性基础的含义。关于计算机安全有一些标准，特别是在术语定义和标准方面尤为如此。当牵扯到安全体系结构时，我们会介绍其中的一些标准。基础的组成还需要认识和教育部分，同安全环境维护有关的人员需要通过这一部分来学习安全体系结构。建立了一个健壮的基础之后，安全体系结构的其他部分就可以安全地在基础之上进行构建了。

在本章中，将探讨基础的组件以及基础如何支持安全体系结构的其他部分。前面已经说过，安全体系结构的基础是建立在一套组织已经制定或者应该制定的决策或声明之上的。组织参考这些决策和声明来开发体系结构的细节、安全策略集以及安全标准标识。这些决策和声明应该回答如下问题（并非全部）：

- 组织需要保护什么？
- 组织的安全思想是什么？
- 应该遵循什么标准？

- 员工有权访问所有的信息吗?
- 谁负责安全?

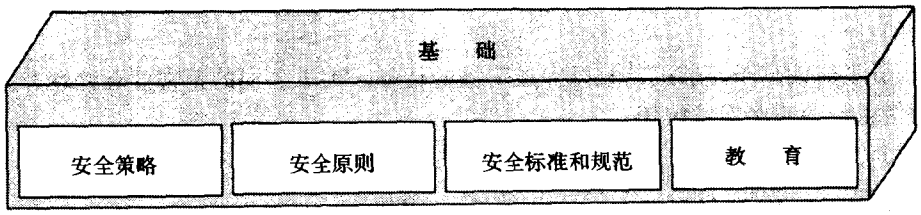


图5-1 安全体系结构基础

5.1 原则

安全原则是对公司价值、业务或者信念的声明，它依次定义或者影响安全体系结构的所有元素。原则集合反映了组织对安全的哲学思想，安全体系结构和安全策略都在其基础上进行开发。安全策略和过程应该依次在反映公司思想的业务原则、声明的基础上进行构建。原则应该是完整的，并且应该是易于理解和执行的。原则是驱动着安全框架的价值、操作或信念的声明。原则可以看做是安全策略和过程必须遵循的规则。下面是一些声明类型示例，几乎所有的组织都会把它们列为原则：

“数据是一项公司资产，它必须在公司范围内进行管理。”

“组织应提供足够的安全和业务控制以保护数据，同时管理部门应进行风险评估。”

“数据是离散信息的表示，信息是决策制定数据的一个汇集。”

“公司应提供足够的安全和业务控制以保护数据，同时管理部门应进行风险评估。这些控制应定义一个可用于所有系统的最小标准，并提供一个系统测量标准。”

“信息的拥有者、提供者、用户以及同信息安全相关的其他人员所承担的责任和义务必须是明确的。”

“与信息安全相关的拥有者、提供者、用户以及其他人员必须能够迅速获得同安全策略、实践以及过程的存在和扩展相关的知识。”

“安全测量、实践以及过程不应该明显的降低授权用户对公司资源的使用效率。这些测量、实践和过程应该协调并统一起来以创建一个连贯的安全系统。”

“对具体信息系统来说，根据其价值和重要性，安全需求会有所不同。安全实践和过程应该同信息系统的价值和企业对其依赖程度相一致。这些策略应该直接和间接地同潜在危害的严重性、可能性以及危害程度相称。”

“信息的拥有者、提供者、用户以及公司的管理部门必须要及时配合，预防、检查并响应安全漏洞。”

“信息的安全性必须要定期进行重新评估，同时要考虑技术和业务需求的变化。”

“安全策略、标准以及过程应该建立起来以用做信息安全活动的管理计划、控制和评估的一个基础。”

有一些原则依赖于组织的具体思想和文化。某些组织可能默认信任所有员工或者在一个严格的需要知道的基础上授予权限。例如，有的组织可能把原则定义为对敏感信息专门赋予访问权限，默认拒绝所有其他访问；而有的组织则恰恰相反，明确拒绝对敏感信息的访问，

默认允许所有其他访问。原则会影响到安全机制的实现。下面是几个声明类型示例，对每个组织来说，它们可能不尽一致：

“信息安全途径应该使用小且简单的保护机制以提供所需的安全级别，而不应该使用大的或者复杂的。”

“公司员工应该赋予足够的特权以完成所派的任务。为了公司的正常业务处理，公司应该对员工的信息资源使用保持信任。”

“公司的安全策略应该放在分布式计算环境的中心位置。”

5.2 安全策略框架

安全分布式处理环境的建立必须要在管理部门所支持的清晰简洁的声明基础上进行。安全策略提供了一个对于分布式处理环境非常重要的框架，其目的在于保证公司的信息资产能够在一个指导策略的控制之下保持安全。安全框架定义了安全策略，它能够为安全测量的实现和维护提供指导，从而保护组织的信息资产。安全框架应该提供指导和概括责任，而不是定义过程或者处理。安全策略框架的目标是高效率的管理风险，明确员工的信息资产保护责任，为一个稳定的处理环境建立一个基础，保证同可实施的法规和制度的相容性，以及在出现资产误用、损失或者未经授权泄漏时维护管理选择权。

安全框架必须是开放式的以进行连续的开发和更新。有许多原因可以解释实现一个安全框架以及用于框架管理的处理是非常必要的。同其他组织的采集、合并、合作或者交互都可能会给安全策略或者标准带来问题。组织所选择的策略和标准不可能照顾到环境的所有方面，因为环境会变化，所以这些策略和标准会不断地分析和更新。

安全策略框架的成功可以通过关键成功因素来进行测量。这些因素定义成一些标准，安全策略框架应该绝对成功实现这些标准。关键成功因素包括高级管理部门对安全策略和标准的承诺和支持，用户群对策略和标准的认识，保持策略和安全标准的技术独立性，策略的可读性和明确性，以及策略和安全标准在安全体系结构中的集成。

信息资产指的是信息以及用来访问、处理和传递信息的资源。安全框架应该应用到支持组织业务活动的所有信息资产和服务上。信息资产包括数据、图像、文本、语音、视频、任何信息处理设备（如大型机、PC、打印机）、介质（如磁盘、磁带）、网络、支持工具以及信息处理服务。

安全策略提供了安全基本结构的基础。如果没有正规的策略，那么让用户为其行为负责是非常困难的。策略提供重要的指导以帮助不断增加的组织间互连和合并。策略和过程是独立于技术的。组织的策略和过程必须是可实施的以及易于理解的。

策略通常存在于具有数据分类、标识、授权、认证和数据级别的地方。策略也为组织内和组织间的关系和责任提供指导。我们将详细探讨需要考虑的主题和策略开发过程。我们假定安全框架策略不会涉及到员工的行为，那是其他行为标准或者其他业务策略所处理的问题。

5.3 安全标准

安全标准的定义（和原则）是安全体系结构基础的第二个主要部件。安全标准指的是对一个安全环境所需属性的确定和定义。这些属性可以分成几类，它们为安全组件和机制的产生和过程选择提供了基础。安全标准为安全组件、技术功能、业务处理的评估提供了一个参

考规范。安全标准定义了所需要的安全功能集，它是独立于平台或厂商的。不同的业务领域或者不同的计算系统需要不同的安全类型。在互联的分布式平台上提供系统和服务需要仔细选择技术和产品，而安全标准则直接影响到所做的选择。

当前有几个看做是工业标准的标准安全规范，但是其中大多数都是适用于军事系统的。美国国防部和国家计算机安全中心已经编著了很多有关计算机安全方面的书籍，这些书分别针对于计算机安全的几个不同方面，其中每一本都由一个不同的颜色来标识。可信计算机系统评估标准（Trusted Computer System Evaluation Criteria，TCSEC）桔皮书提供了一个广泛认可的框架，系统厂商都以此为基础来构建其产品的安全组件。可信计算机系统评估标准中的可信网络说明（Trusted Network Interpretation，TNI）红皮书对桔皮书做了扩展，它提供了桔皮书应该如何在网络环境中解释的详细指导。红皮书也描述了适用于网络环境的附加安全服务。欧洲计算机制造商协会（European Computer Manufacturers Association，ECMA）定义了一个更为商业化的面向安全的标准。开放式系统组织X/OPEN已经着手定义操作系统安全功能的最小集合，该功能集要包括到带有X/OPEN商标的系统中。

1. 可信计算机系统评估标准（桔皮书）

美国国家计算机安全中心（U.S. National Computer Security Center）已经在一个称做桔皮书的文档中定义了对安全分类的一个描述和需求。该文档可以用于评估商业产品，其评定和分类结果公布到一个受评估产品列表中。各种部门和机构都采用该列表来比较不同产品的安全性能和组件。对组织具体安全分类的判断为需求的定义和安全产品及机制的评估提供了一个基础标准。

可信计算机系统评估标准中定义的桔皮书分类主要着眼于军事和政府部门对分类信息机密性的需求，但是它也能为一般的安全技术途径提供指导。市场上的许多操作系统都是“C2”认证或者“C2”兼容的。二者区别是，被认证的系统已经通过了一个官方认证过程；而兼容系统虽具有该分类定义的所有功能但却没有进行过认证。一般来说，“C2”应该是企业中每个系统所应达到的最低分类标准。表5-1概括了桔皮书中定义的各个安全分类。

表5-1 ITSEC桔皮书安全标准规范

类别	说 明
无	任何不满足A、B、C分类要求的系统
C1	**任意安全保护**：系统具有某种形式的基于个体的有限访问控制功能。用户决定实施何种保护，如UNIX文件许可
C2	**受控访问保护**：本类系统比C1系统具有更严格的访问控制，通过登录程序、事件审计以及资源隔离等措施从而使得用户可以独立地为其行动负责
B1	**标记的安全保护**：除了C2级别所需的全部功能以外，本级别还需要一个安全策略模型的非正式声明、数据标记以及对命名主体和对象的强制访问控制。系统实施某种形式的保护，该保护不受你控制
B2	**结构化的保护**：B1中的全部控制都扩展到了系统中的所有主题和对象之上。另外，本级别还涉及到转化通道
B3	**安全域**：必须满足参考监视需求。该需求转交主题对对象的所有访问。它可以防止篡改，并且足够小以进行测试和分析
A1	**已校验的设计**：等价于B2级别，但用户必须证明安全模型及（或）其实现是安全的
A+(2)	**已校验的实现**：大量的校验方法、可信设计环境以及高级测试过程

“B”级安全类别主要用于军事系统，但是在某些情况下它也适用于商业系统。“B1”类

别是需要安全保护的敏感数据及应用的目标。有些特殊的应用和数据需要B2级别实现对主机或者服务器上的所有组件进行高级保护。安全分类B1级需要强制的访问控制并对输出的信息进行标记。C2级为受控访问保护建立了一个基础，这种受控访问保护为分布式平台上的选择提供了指导。一个已标识的安全标准分类为那些可以在分布式平台上评估、选择以及实现的安全机制建立了一个公共基础。

2. 面向商业功能分类

欧洲计算机制造商协会在其发布标准205——面向商业功能分类（Commercially Oriented Functionality Class，COFC）中文档化了另一个安全实施功能和定义规范。该标准主要面向商业市场和多用户独立IT系统。有关网络和分布式处理的规范正在开发中。

COFC标准为商业市场定义一个可广泛接受的基本安全功能分类。该标准有别于桔皮书，但是它同TCSEC趋向融合。COFC标准的描述方式同TCSEC一致。COFC也可用做安全术语定义基础。

3. X/Open基线安全服务

X/Open是一个在1984年建立的组织，其目的是为开放式系统产品协调需求和标准。遵循该组织定义标准的产品可以使用X/Open商标。X/Open组织已经定义了一个分布式安全框架，它可用来指导开放式系统的安全技术开发。X/Open基线安全服务（X/Open BASELINE Security Services，XBSS）规范定义了开放式系统必须提供的安全相关功能的最小集合以及开放式系统安全相关参数的默认设置。XBSS规范是X/Open商标过程的一部分，它可以对支持安全的系统进行商标化。厂商提供的许多操作系统没有打开安全功能。在这些功能打开之前，系统将会无安全运行。XBSS为厂商获得其产品中所包括的安全功能认可提供了一个方法。

5.3.1 标准

安全组件和安全方法的标准选择也是安全体系结构基础的一部分。我们需要为安全体系结构的定义和通用组件选择已经建立的标准。现在还不是确定技术标准的时候。前面提到过，安全体系结构不应该依赖于技术。当到了定义基于安全体系结构的安全解决方案的时候，策略、原则以及安全标准的建立将会包括技术标准。下面是一些可用的标准选择。在确定安全体系结构的时候，它们可能会有用。

1. 原则

在会计学里，有一种称做通常公认的会计实践（Generally Accepted Accounting Practices）的声明。这些声明为两个组织间会计实践的比较提供了一个基础。与此类似，信息系统安全协会（Information Systems Security Association）及其他相关组织定义了一个通常公认的系统安全原则（Generally Accepted System Security Principles，GSSP）。1994年出版的一份报告中提供了定义和推荐框架，并且概括了信息安全专业人员和信息处理产品应该遵循的原则。GSSP文档概括了17条原则，它们可以对一个组织的安全原则开发工作提供指导。

2. 策略

现在没有同策略或者过程的开发相关的标准。策略题目及其内容的选择因组织而异。美国国防部和国家计算机安全中心提出的TCSEC中的红皮书为策略标准提供了一些指导。

- 桔皮书：可信计算机系统评估标准（Trusted Computer System Evaluation Criteria）。
- 红皮书：可信网络说明（Trusted Network Interpretation）。

- 绿皮书：密码管理指导（Password Management Guide）。
- 蓝皮书：个人计算机安全考虑（Personal Computer Security Considerations）。
- 褐皮书：可信数据库管理系统说明（Trusted DBMS Interpretation）。
- 黄皮书：可信环境说明（Trusted Environment Interpretation）。

3. 定义

安全体系结构标准ISO/IEC 7498-2，信息技术——开放式系统互联——基本参考模型——第2部分：安全体系结构（也称做ITU-T Recommendation X.800），为用于标准定义的术语提供了正式的定义。当定义和说明安全组件时，应该使用这套定义来建立一个公共框架或者参考。

4. 物理安全

当前，这一领域只有一个标准：美国国家通信安全指令5100A（National Communication Security Instruction 5100A，NACSIM 5100A）中制定的风暴标准（Tempest standard）。该标准适用于可以通过使用计算机设备进行电子侦听来获取信息的环境。风暴参考包含或者抑制电子设备的信号发射技术，并且指定了允许的电子设备发射限制。根据风暴标准制造的产品如果要出口到美国之外，那它必须要有美国国务院的军需控制办公室授予的相关输出许可。风暴标准文档（NACSIM 5100A和NACSI 5004）是自身分类的，它只在一个需要知道的基础上才可用。

5. 安全管理

SNMP标准的第二版中包括了RFC 1351：管理模型（Administration Model）；RFC 1352：安全协议（Security Protocols）；以及RFC 1353：用于SNMP用户管理的受管理对象的定义。这些标准保证了网络管理通信的认证和保密。认证保证了消息的正确源，保密则保护消息免遭泄漏。

6. 监视和探查

OSI标准，ISO/IEC 10164-7（X.736），安全警报报告功能，列举了14个安全警报类型及其可能的原因。例如，如果有人尝试在调度时间段之外执行一个操作，那么这就会产生一个时间域违背。

7. 审计

审计过程和机制方面几乎没有什么标准。DCE RFC 29.0为DCE审计子系统的设计概括了实现规范。OSI标准ISO/IEC 10164-8（X.740）——安全审计跟踪功能（Security Audit Trial Function）定义了审计跟踪记录和生成该记录类型的事件类别。

8. 认证

认证标准是ISO在ISO/IEC 10181-2（X.811）——认证框架（Authentication Framework）和ISO/IEC 9594-8（X.509）——目录认证框架（Directory Authentication Framework）中提出的。X.811标准解释了描述认证原则和体系结构的术语，并解释了一个用于不同认证交换机制的高级分类方案。X.509标准使用密码或者公钥加密的X.500目录认证。

9. 访问控制

ISO/ICE 10181-3（X.812）——访问控制（Access Control）为在分布式环境中提供访问控制提供了术语和体系机构模型。

10. 机密性

ISO/IEC 10181-5（X.814）——机密性框架（Confidentiality Framework）描述了提供机

密性所需的术语和机制。该标准是ISO开放式系统互联——基本参考模型——第2部分：安全体系结构（ISO Open System Interconnection——Basic Reference Model——Part 2: Security Architecture）的一个组成部分。ISO/IEC 10181-6（X.815）——完整性框架（Integrity Framework）为提供完整性服务所需的定义和规范进行了文档化。

11. 认可

ISO/IEC 10181-4（X.813）——认可框架（Nonrepudiation Framework）为在分布式环境中提供认可提供了术语和体系结构模型。

12. 鉴定

计算机安全认证和鉴定指导（the Guidelines for Computer Security Certification and Accreditation）是一个来自联邦信息处理标准发布（Federal Information Processing Standards Publications，FIPS PUBS #102）的文档。该文档包含了为安全系统的认证和鉴定建立一个程序所需要的信息。

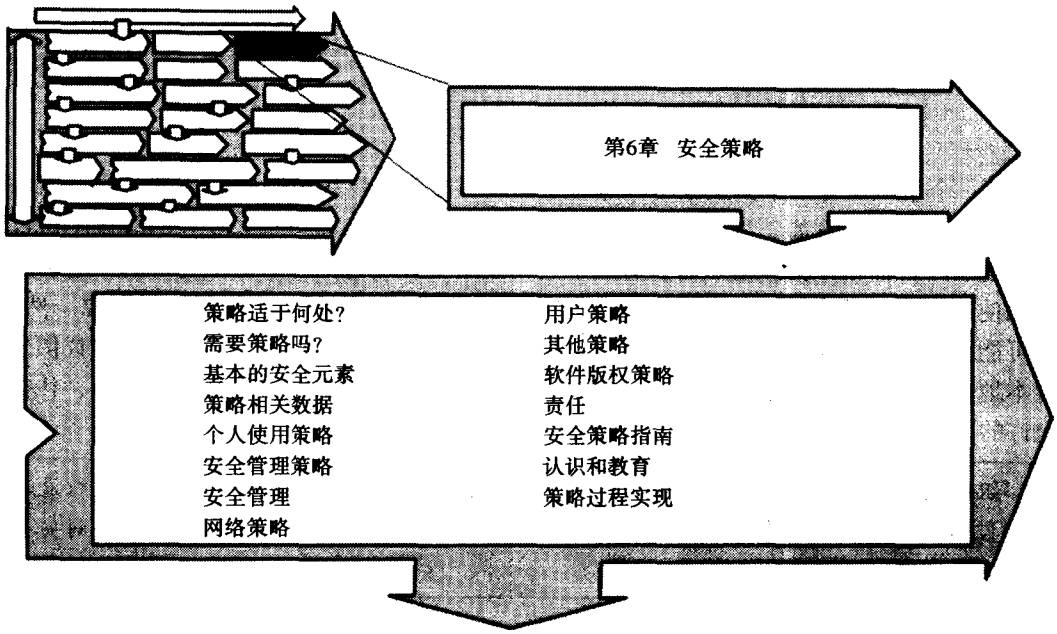
5.3.2 指导原则

对一个综合的分布式安全环境，指导原则是个重要的组件，但是它不考虑为安全体系结构的一部分。指导原则是应用到原则、策略和特殊领域标准之上的文档化的实现过程和程序。原则、策略和标准在体系结构定义的范围内保持一致。指导原则为这些元素如何应用到具体平台或技术上提供了详细的指导。例如，一个策略可能为分布式环境中的授权定义目的、范围以及一致性需求。具体的指导原则将来定义说明授权功能如何在具体平台上实现。指导原则是策略和原则所定义的安全法规和规则的理论定义同实际应用机制之间的转换。每个使用不同技术基础的平台可能有不同的指导原则的实现。

5.4 小结

就如同一个坚固耐劳的房子一样，一个构建良好的分布式安全系统也应该建立在一个牢固的基础上。基础的一些组件会因组织而不同。一个深思熟虑的基础在一个分布式环境中要比在一个集中式环境中更为重要。基础是体现一个组织决定其安全思想的地方。这些决定不应该在信息技术系统有了大变化的时候就必须改变。应分析为一个集中式环境建立的任何原则和策略对于一个分布式环境的适用性。在下一章中，将讨论一下组织如何通过一个正规的计算策略来向员工提供指导从而建立正确的环境。

第6章 安全策略



前一章中已经提到，安全策略是用来构建安全体系结构基础的最重要的组件。安全策略通常是一个受到关注的领域，特别是当读者有自己的看法时。但是一个实现拙劣的策略通常不能覆盖到分布式环境所涉及到的所有方面，并且不能实现预期的目标。组织需要使用一个方法来建立一个正确的结构，该结构应该涉及到综合策略中的可接受的行为、权利和义务。该结构应该辅助同安全相关的行为，它可以设置方向、为公司提供指导、说明高级管理支持。

在任何计算环境中，安全策略都是非常重要的，这可以保证企业的所有信息化资产都是安全的。在分布式环境中，安全策略可以说是至关重要。在这种情况下，企业的信息资产可能分布在多个平台和多个地域中。策略不仅保护着信息的机密性，它还保护着信息的可用性、信息的使用、所提供信息的真实性以及信息的完整性。一个安全策略实际上是处理具体领域或者具体方面的独立策略的集合。这些可以看做是在第5章中提到过的安全框架。

安全策略适于的位置

在第5章中，概述了对一个牢固基础的需要。基础包括公司原则和安全策略。图6-1说明了决策和方向流如何导致安全策略和指导原则的定义。公司原则导致了公司业务守则的定义。原则反映了公司的思想，而业务守则把这些原则转化为员工自我管理的预期方式。如果原则表明公司以一种合乎道德的方式来进行业务，那么业务守则就会概括出员工如何实现此目标。安全策略使用原则和业务守则为基础来构建处理安全问题的具体公司策略。这些策略会依次提供对指导原则定义的输入，指导原则定义了安全策略的实现方式。

需要安全策略吗

实事求是地说，没有安全策略不会使一个公司无法运营。如果不出现安全事故或者安全危机，那么策略对于公司的价值可能是不明显的。在许多情况下，策略因为一个具体问题或者作为一个定期审计的结果而被摆上台面。对每个公司来说，其需要策略的地方会不同，这与公司的业务类型有关系。组织对业务处理的信息系统越依赖，它对健壮策略的需要就越大。如果公司的客户或者业务伙伴也依靠你的信息系统，那么这一点尤为正确。

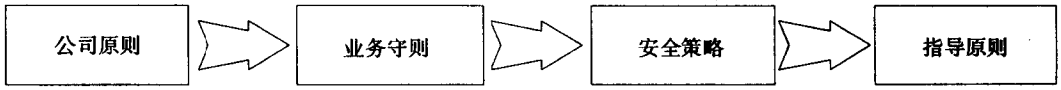


图6-1 策略适用的地方

如果一个组织没有一个安全策略，那么在出现明显违背组织原则的行为的时候，该组织可能对其无能为力。例如，当一个员工利用组织的计算资源来处理私人业务的时候。这可以认为同组织的目标相冲突。组织需要有一个公司策略来表明这类行为是不允许的，组织的任何资源使用都建立在该策略之上。否则，员工可能会不守纪律而非法使用组织资源。然而，组织可能允许一个个体使用其计算资源以支持一个非营利或者慈善组织，如一个体育俱乐部或者本地食品储备等。在这种情况下，组织应该使用一个策略来标明什么是允许的、什么是不允许的以及这些行为是如何标识和授权的。

> 组织可能需要一些策略来合法地保护公司不会受到员工行为的影响。如果组织没有一个控制策略来定义什么是可接受的或者不可接受的，那么组织就不可能对一个同安全相关的事件进行法律追究。

因特网是一个热点。大量的组织都使用因特网进行了互联。它也是推动组织使用特殊安全策略的一个重要原因。如果组织连上了因特网，那么员工在因特网世界上可以代表组织。这可以通过电子邮件地址或者信息张贴反映出来。组织需要员工使用因特网以完成合法业务目的，并且能够是组织的一个负责代表。如果员工使用因特网来访问面向审计的信息，或者发表了一些同组织原则相悖的看法，那么这些行为将看做是不可接受的。一个可接受的因特网使用策略应该规定用于业务目的的因特网使用范围以及可接受的行为范围，从而使得员工可以成为组织的一个电子代表。

6.1 安全策略框架

安全策略的构建是非常依赖于具体组织的，每个组织可能各不相同。几个组织已经使用了一个安全策略框架。然而，同一框架可能并不适用于所有的组织。但是它为安全策略的构建提供了一个途径。在安全框架中的策略通常是由一些标准部分组成的，其中每个部分概括了策略以及实现该策略的原因。策略应该是简短扼要的。它们会分布到所有相关的业务实体上。它们不应该具体到任何硬件平台上，并且应该是独立于技术的。每个策略应该包括：

- 策略声明——公司策略的一个正规声明。组织需要什么？不需要什么？策略声明应该是简洁扼要的。
- 目的——为什么需要该策略？它解决什么问题？策略有什么组件？该策略涉及到的领域有什么定义吗？

- 范围——策略应用的边界范围是什么？策略如何扩展？
- 策略兼容性——该部分包含策略的具体组件和应用。支持该策略具体需要些什么？有允许背离策略的情况吗？如果有，需要什么处理来授权这种背离行为？
- 处罚/结果——不支持策略有什么后果？或者进行什么处罚？如果实际情况中出现了背离策略的行为，那么这一部分可以规定正式制裁和（或）具体后果。

安全框架所包括的内容因组织而不同，它取决于具体的业务环境、操作模式以及管理原则。下面给出了几个可供考虑的专题。这个列表并不详尽，但是它标识了许多可能需要一个具体策略的领域。这些策略专题组成几个组，其中第一个涉及的是几乎所有组织都应该考虑的策略。

6.1.1 基本的安全元素

在任何安全框架中都可以找到一些基本的安全元素。组织应该通过一个或者一组策略来提供这些基本的组件，并规定它们应该采取的形式和使用方式。本安全策略组标识了安全机制基础。这些策略，或者这些策略之后的安全机制，包括在大多数安全系统中。

1. 标识

数据处理系统的每个独立用户和数据必须要进行惟一标识，从而提供个体责任。一个惟一的用户标识码，称做用户ID，通常用来代表一个用户的系统标识。一个有效的用户ID（根据密码进行校验）提供了对用户标识的认证。标识策略应该详细说明对标识的需求，并说明应该如何实现标识。例如，策略应该规定，惟一的用户ID会指派给个体，并用于该个体的标识。用户ID会用做惟一的标识符，并且用于授权、认证和审计目的。

2. 认证

任何安全系统的基础都能知道谁或者什么在请求数据处理操作。对请求标识的校验是允许对数据资产和系统进行访问的一项基本需求。如果一个标识具有某种形式的独特数据或者独特性质，那么就可以认证该标识。策略应该规定，何时需要认证、认证根据什么，以及使用何种认证机制。认证机制可以是带有密码的惟一的用户标识，或者用户具有的某样东西，如一个智能卡或者一个拇指印等。

3. 授权

对公司计算资源的访问只应该授予那些管理部门许可的个体或者连接主机。这种许可是授权机制所授予的，这种授权机制可用来允许、拒绝或者限制个体或者连接主机对资源的访问。授权通常建立在一个认证的标识上。策略应该定义用于授权的元素，如已认证的用户ID。策略也应该指出，谁建立授权标准，谁授予资源访问授权，以及谁管理授权过程。

4. 密码

用户或系统惟一标识和伴随密码通常就是提供保护系统资源的基本安全级别的全部。可能也会需要另外的安全保护，如一个智能卡，这要取决于对象数据的类别。密码策略应该涉及到密码的处理、格式、密码交换标准以及保持密码安全的需求。密码策略的关键组件之一应该是保持个体密码安全和问题出现后果的需求。

5. 信息完整性

信息完整性同组成目标信息的集合数据的可靠性、准确性以及管理密切相关。对于一个可靠的业务计算系统，系统的正确操作和信息完整性的维持是必需的。由于信息是一个数据

的集合，所以完整性的实现有助于保证所有的数据都是可靠的和准确的。对标识信息生成者的需求，或者完整性检查工具的定期执行，都是完整性方面的例子。关于信息完整性的策略应该概括出对实现完整性机制的需要，以及它们是什么和如何使用它们。数据完整性同独立数据元素的可用性和准确性密切相关。下一节将介绍关于数据完整性的策略。

6.1.2 同数据相关的策略

安全框架中的其他元素都围绕着数据保护这一问题。组织的秘密是安全预防的首要推动原因。组织需要明确各种数据组件的机密性需求，并对其给定一个定义。这会提供一些有关保护数据所需的安全测量的类型和深度方面的答案。例如，组织可能定义一个同公司文档的出版相关的策略，并规定谁可以访问这些文档。

1. 数据分类

综合安全框架主要的组成部分是数据的分类。组织内生成和维护的各种信息有着不同的敏感程度，这取决于数据的特性和用途。所有的数据应该根据其特性和目的进行分类，从而定义同信息的完整性、可用性以及机密性相关的正确测量。组织应该有一个策略来标识数据的类别，以及每个类别所要求的访问、控制以及管理操作。例如，一个策略可能会指出高度机密的数据必须要以加密的形式进行存储和传输。

2. 数据监护人（所有权）

组织中的数据是一项有价值的资产，因此必须要得到保护。资产保护的指派和管理是数据监护人的职责。术语“数据监护人”（data guardian）同数据所有权这个词区分开。在大多数组织中，数据的所有权不会给任何个人，而是由组织所保留。数据监护人有责任保护数据资产，但却不需要指派所有权。在某些类型的公司里，这会称做智力所有权的保护。关于数据保护的策略应该规定数据监护人的职责以及同数据所有者或数据保管员的关系。

3. 数据保管员（管理）

数据不仅仅必须要受到保护，还必须要管理它。该种管理责任由一个称做数据保管员的人承担。数据保管员有责任维护数据资产性能的可靠性以符合数据监护人的规范。数据保管员为数据监护人的数据管理起到了委托人的功能。数据保管员已经授权了资产的拥有，并有责任遵循数据监护人的授权或者指定控制。数据保管员有责任管理数据的可用性，保证适当的控制并使用，担任数据的公司代表角色，对未经授权的泄漏和数据破坏进行风险分析，为数据的访问和使用建立适当的数据分类和认证标准。

4. 数据完整性

数据完整性同独立数据元素的准确性和可用性密切相关。当需要使用的时候，用于提供公司信息的数据必须是准确的和可用的。这些数据必须要被保护以免遭到破坏或者滥用，并且它们要包括测量以保证其准确性和可用性。关于数据完整性的策略应该标识出对安全数据存储和数据存储备份机制的需求，对保护数据准确性和测试用来操作数据的程序和过程准确性的过程和程序的需求，数据的复制，以及数据的私密性和完整性。关于信息完整性的策略应该涉及到对管理完整性机制的需求。

6.1.3 个人使用策略（通用）

计算机的使用为我们的办公任务带来了一场革命性的变化。你应该考虑为员工或者其他

公司相关人员制定一些有关计算机使用的策略。

1. 个人使用

计算资源是服务于公司业务需求的。公司可能不允许员工把这些资源用于个人行为而不是进行工作，除非管理部门允许。组织应该有一个策略来指定个人使用公司计算资源的可接受范围，以及获得管理部门许可的标准和过程。

2. 信息系统的使用

现在，公司内部和外部对电子邮件和消息技术的使用正日益增长。这为创建并分布化信息提供了一个方便的的方法。信息技术系统的用户必须以一个合法、负责的方式来使用这些系统。系统和网络不应该用来生成或者分布非法的、不道德的以及同公司原则相悖的信息。

3. 保密

公司保持个人和其他数据以管理和操纵业务。个人信息机密性的保护是非常重要的。所有的员工必须要采取积极的步骤来维护信息的机密性。公司可能也受机密规定或者条例的控制。如果它们控制着组织的行为，公司应该有一个策略来概括出对条例或法规的需求。

6.1.4 安全管理策略

安全机制的管理也是策略应该涉及到的一个地方。策略应该指出其涉及的行为和责任。安全机制只有在符合定义良好的综合管理策略的情况下才有效。

1. 管理职责

安全机制、过程和程序的成功是一个管理职责。该职责通常给公司内所有具有管理责任的人。监视和实施安全结构是管理部门的职责，所有的个体和系统组件都在他们的直接控制之下。公司应该有一个关于管理职责的策略来概括出高效管理所需的这些职责和过程。

2. 记录管理

对机密信息的访问以及对重要公司数据的修改都应该记录下来。这些记录对提供一个综合安全框架是非常重要的。这些记录用来提供作用到数据和系统上活动的审计跟踪。策略也可以指定信息的持久需求，以及机密信息需要删除或者破坏的时间。组织可能需要使用记录来满足正式的和合法的需求。

3. 安全管理

安全管理是企业内信息安全机制的焦点所在。安全管理意在保证所有的安全测量和机制都是正确实现、管理和监视的，并且能够根据业务条件而正确地变化。它也有责任协调安全信息和安全事故调查。安全管理不负责管理策略——这是管理部门责任策略应该负责的地方。

4. 责任分离

在处理安全问题时责任分离是一个非常重要的原则。把同任务相关的安全和管理的组件指派给不同的个体是更可取的。如果没有个体具有对安全组件的总控制，那么就没有人能破坏安全系统。一般情况下，责任分离是在下面几个职责中实现的：系统开发、系统管理、安全管理以及系统操作。公司应该使用一个策略来增强责任分离原则，并说明原则应该如何实现。在大金额的交易中，银行一般需要两个员工签名，一个出纳员和一个经理，这就是一个责任分离的例子。

5. 风险评估

风险评估是一个过程，它标识一项资产所面临的风险，并且在资产的价值和损失所带来的影响的基础上指定保护该资产所需的控制。一个积极的安全计划的基本需求是对需要保护的所有资产进行标识和分类。一个正式的风险评估程序有助于资产保护措施的计划和选择。并不是所有的风险都是在可接受的成本范围之内，对于管理人员来说，接受某种级别的风险既是合理的也是谨慎的。安全和控制并不自我完成，它们应该总是由业务需要证明，从而限制资产对未授权用户的真正暴露。

6. 兼容性监视

安全是操作和使用数据处理系统的一个重要元素。策略、工具、过程和程序应该实现以具有所需的安全级别。有一些方法可以用来监视安全措施和策略的兼容性。同安全措施的不兼容性可能会导致信息的非法暴露，并且可能对业务处理产生影响。

7. 策略背离

策略是建立在所有平台和应用之上的业务控制标准。框架中提供的策略和标准将用于指导和控制。背离框架或者同框架不兼容应该指出并在策略的指导下进行处理。

6.1.5 系统策略

现在组织的计算机系统是其业务处理中的一个重要组件。如果系统在很长一段时间不可用，那么有些业务就会停止或者造成大量损失。对于系统不可用的时间，有些情况下它可以以小时为单位进行测量；而对于一些非常依赖于时间的事务来说，如股票市场交易所，它需要以分钟为单位来计算。

1. 系统管理

系统管理员有责任为信息处理系统维护一个一致的、高效的操作环境。稳定和一致的系统操作会提供一个高工作效率的环境，而一个不稳定的环境则容易受到破坏。

2. 系统分类

根据支持业务系统性质和用途的不同，组织内操作和维护的系统可能有着不同程度的重要性。为了应用适当的安全和保护机制，以及在灾难恢复和业务持续计划中包括关键系统，所有系统都应该根据其机密性和可用性需求来进行分析和分类。

3. 业务持续性

信息处理系统的可靠性是业务的一个重要方面。数据、组件或者系统的不可用性会对业务操作产生一个严重的影响。对于业务操作来说关键的系统应该具有一个策略，该策略应该表明系统对一个业务持续计划的需求。

6.1.6 网络策略

网络提供了对计算机系统的访问。如果不受到一些安全策略的严格控制，网络可能非常脆弱。这些策略应该指出谁可授予访问权以及在什么条件下可授予这种网络访问权。

1. 外部网络访问

互连网络来提高业务利润已经成了业务操作的标准的和可接受的模式。为了信息和业务目的而连接到因特网上正日益成为许多组织的业务需求。对于一个组织来说，其网络到一个外部网络的任何连接都应该受到严密控制，并具有严格的安全措施进行保护。如果没有严格

的安全措施来保护网络连接，那么组织内的所有计算机资产都可能遭到破坏。

2. 远程网络访问

员工可能请求或者需要从位于公司网络之外的网络位置来访问公司系统。如果有合适的理由，并且所需的安全措施都到位的话，这种远程访问是可以允许的。在这种情况下，系统还可能需要第二个功能认证机制。这些请求绝对不能破坏系统、网络以及数据的安全性。

为了合作，或者为了进行一些特殊的项目（如年中审计等），第三方也可以请求访问公司的计算系统。这些请求应该具有适当的认可，并应该在严格的安全机制控制下，如具体访问地点、时间限制或者其他限制等。这些请求绝对不能破坏系统、网络以及数据的安全性。

3. 通信秘密

电子通信已经成为业务处理的一个集成部分。对公司智力资产的保护需要公司对电子邮件和电话交谈进行监视。公司应该有一个通信策略来说明在电话交谈和电子邮件机密性上公司的位置。公司可以对电子通信进行监听或分析，这取决于公司的原则、文化以及其对保护敏感信息或智力财产的关注程度。通信策略应该指出是否存在着监听的可能性。

6.1.7 用户策略

系统用户必须要认识到、掌握、并且能够处理策略。如果他们对策略一无所知，那么安全策略也就没什么价值。

1. 用户认识

安全机制和组件应用中的一个主要组件是它们的存在以及要实现它们的原因。组织必须实现并维护一个用户意识程序，从而保证策略是众人皆知的并得到用户的理解。

2. 用户责任

一个用户是一个授权使用信息资产及服务的个体。数据用户是具有数据监护人所授予的数据访问和使用许可的个体。所有的用户都必须意识到安全特性和策略，并了解实现它们的原因。用户有责任承担同组织内信息资产和计算服务的保护及使用相关的义务。

6.1.8 软件策略

公司的软件是公司内计算机系统的管理引擎。保证公司使用的所有软件都得到合法权限是公司的责任。保护软件免遭病毒侵害也是公司的责任。

1. 软件版权

版权法保护了软件生产商创建和发布其软件的权利。软件的使用限制包括进了随软件包发送的生产商许可协议中。大多数生产商允许用户可以为了备份或者工作拷贝而拷贝软件。然而，许可协议一般都指出软件一次只能安装到一台机器上。如果软件的拷贝需要安装到其他机器上，那么用户必须要获得另外的许可，或者必须同生产商签订一份站点许可协议。如果组织需要开发计算机应用程序，那么得到适当的专利、商标以及版权对于公司利益的保护是非常重要的。

2. 病毒防护

在IT系统的操作中，组织需要维护一个病毒防护和检测过程。引入到公司系统中的所有外部数据和软件都必须要首先检查以防止带有任何已知的病毒。公司应该定期进行病毒检测扫描。如果发现一个病毒，就应该标识责任和处理。解决病毒问题的另一个方法是禁止任何

用户使用个人软件。

6.1.9 其他策略

还有其他几个领域也需要考虑安全策略。

1. 应用程序开发

系统开发生命周期中的所有应用设计和开发阶段都应该考虑到安全性。应用系统的设计者和开发者应该知道根据系统和数据分类的所有安全需求。在实现系统之前，一个策略可能需要正式的应用系统安全性评估。

2. 外部处理

用于处理系统的或者员工进行的备用处理的外部服务组织应该受到同内部系统一样的安全环境和控制的保护。你可能没有对环境直接控制，这会引入安全问题。任何外部处理只应该在同内部需求一样的或者等价的安全过程控制下进行管理。应该有一个认证或者分析过程来建立环境的安全性。这应该是用于建立关系的需求之一。

3. 物理安全性

访问包含数据处理工具的领域应该限定给那些具有一个明确业务访问需要的人。一个安全的、受保护的环境对于高效的系统操作是非常重要的。用于这方面的策略需要考虑对计算机资产的访问和保护需求，以及公司数据的存储和保护机制。还可以考虑包括员工所拥有的计算机资产。

4. 标准的使用

采纳基于法律的或者基于事实的标准是一个重要的原则，这些标准会为信息技术系统提供指导。基于法律的标准通常是由政府或者工业标准体定义和管理的标准，基于事实的标准是多个厂商产品广泛实现但却没有一个用于管理的代表标准体的标准。前者的一个例子是由国际标准化组织（International Standard Organization，ISO）管理的开放式系统互联（Open System Interconnection，OSI）标准，后者的一个例子是几乎所有的UNIX操作系统厂商所采纳的网络文件系统（Network File System，NFS）。标准的采纳为技术的选择、开发以及实现提供了指导。标准有助于提高应用程序在分布式计算环境中的互操作性和可移植性。

5. 传真和语音邮件

通常情况下，语音邮件是组织内的一个安全脆弱点，但它是非常重要的。策略应该涉及到语音邮件系统访问密码的保护，并且这要同计算机系统的访问条件相同。关于传真使用的策略应该考虑到传输中潜在涉及到的数据的类别。如果要保护，这种类别可能会阻止传真传输或者其他传真特殊安全机制的使用。组织应该要求，语音邮件和传真的保密性应该得到尊重并受到保护，并且这些形式的业务通信应该以一个安全的方式使用。

6.2 策略的例子

你应该考虑为前面提到过的每个领域定义策略。前面的列表并不是详尽的，但是它列举了需要考虑安全策略时的主要组件。下面是一个软件版权的示例策略，它说明了一个策略应该涵盖到的主题。

软件版权策略

> 同伴随于独立软件产品的许可协议中的声明一样，公司要求软件的所有用户都要严格遵循有关版权的术语和条件。

1. 目的

版权法保护了软件生产商创建和分布其软件的权利。软件的使用限制包括在随软件包发送的生产商许可协议中。大多数生产商允许用户可以为了备份或者工作拷贝而拷贝软件。然而，许可协议一般都指出软件一次只能安装到一台机器上。如果软件的拷贝需要安装到其他机器上，那么用户必须要获得另外的许可，或者必须同生产商签订一份站点许可协议。

许可看起来无罪的行为实际上是在违背版权法。它们包括：

- 在购买前拷贝软件以测试购买。
- 创建一份临时拷贝并一直使用到购买了一份为止。
- 拷贝公司软件到别处去用。
- 在试用期过后再次获得软件的评估版或者演示版。

2. 范围

对于所有软件的使用，具备现在的软件版权协议的知识是必备条件。所有的员工都需要真正了解受其控制软件的许可协议。

3. 策略的兼容性

独立许可协议的疏忽不是对不兼容性的合法解释。为了保证同有关版权之术语和条件的兼容，用户必须：

- 只依照在软件许可协议中提出的术语和条件来制做拷贝。
- 遵循有关版本和发布升级的软件许可协议。
- 在删除家用软件之前先得到管理部门的批准。
- 公司内使用的所有专利软件，一定要有在任何时候都可用的文档化的购买凭证。
- 在演示、评估或者诊断测试完成之后，删除专利软件。专利软件是某个人而不是公司具有版权的软件。
- 在公司的计算机上安装或者使用个人所有的软件前一定要先得到管理部门的批准。

4. 处罚/后果

把软件拷贝用于许可协议声明之外的目的是违法行为，应该严厉禁止。有意或者无意地违反版权法或者许可规定的用户可能会负个人法律责任。

6.3 建立策略的过程

前面已经定义了策略的框架，现在需要添加内容，从而保证这些策略是众人皆知的、文档化的、批准的、更新的和管理的。公司安全策略框架必须要由一个组织良好的管理过程控制。该过程将定义建立该过程的授权、策略需求和建立的实现、批准策略的过程、实施策略的机制以及用于策略维护和管理的程序和责任。高效安全策略管理的关键任务是建立一个认识和教育程序以保证该安全策略影响到的每个人都能知道安全策略的存在及其内容。安全框架应该包括框架中包含的策略的定义和管理的方法。

6.3.1 责任

安全策略的建立、实现、实施以及管理可能会牵扯到几个人或者几个小组。这些责任应该定义并文档化成安全策略过程的一部分。安全策略的构建、实现和实施应该得到公司的高级管理部门的支持。该管理部门应该最终为应用策略的业务的成功操作负责。其责任不仅仅包括在出现安全事件的时候进行补救的能力，还包括指示安全策略应该改变的权力。

需要建立一个正式的用于策略评审的过程。评审的目标是检查策略的适合程度并保证策略同公司原则、公司业务以及其他已经定义的策略是一致的。策略的批准应该来自于高级经理或者其他指派的高级管理人员。策略批准过程中所牵扯到的管理人员越高级，策略的影响就越大。策略评审和批准过程应该定期进行计划和安排。

一般情况下，安全策略的实施是具有已定义管理责任的人的职责所在，但是高级管理人员要最终为策略的实施负责。这不应该是负责安全机制管理小组的责任。违背安全策略的行为应该由通常的管理过程确定。

6.3.2 安全策略指南

安全策略过程的主要焦点是安全策略指南。该指南应该包含安全策略框架的目的或者需求、策略所基于的原则、用于开发策略的方法、如何管理策略、框架中包含的具体策略。任何受策略控制的人都可以参考该指南。该策略指南应该定期评审，并能同策略的采纳或更新保持同步。策略评审和批准过程后应该对策略指南进行更新。当策略正式批准的时候，应该对策略指南进行更新，并且要把所有的更新都发送给指南的持有者们。并且，这些更新也应该包括在策略所影响到的所有员工和用户之间的定期通信中。

6.3.3 认识和教育

建立一个全面的认识和教育程序是非常重要的。对于那些应该受到策略影响的人来说，如果他们对策略的存在以及策略的内容一无所知，那么策略就根本没有任何影响。对于公司内的员工来说，如果他们对策略和其他组织规则根本不知道或者置若罔闻，那么安全程序就根本起不到任何作用。每个人都应该清楚地知道他们对什么资产负有责任以及那些资产应该受到什么样的保护。

公司应该开发一个正规的教育程序，并使其为全体员工和用户所用，特别是新员工和新用户更要通过此来接受安全教育。该教育程序应该为每一个需要知道和掌握策略的人提供一个对有关安全、重要性以及责任的需求的总揽。该程序应该覆盖到主要的策略，并说明哪里可以得到全部策略的详细信息。公司应该为员工或者用户颁发签名证书以证明其已经阅读并掌握了安全策略，强烈向你推荐这么做！教育材料必须要保持最新。

6.3.4 策略过程实现

启动策略过程的方法之一是让公司的一个专题研究组来专门处理安全策略。例如，公司可以选出一个小组来开会解决以下目标，所需的安全需要和级别、安全角色和责任以及实现策略的过程。参加会议或者工作组的成员就是负责提供安全并且在出现安全问题的时候承担责任的决策制定者。内部审计员也应该参与这类工作。工作组应该提供对安全问题的一个公

共理解，并且提供一个论坛来定义解决问题所需要的计划。

得到组织最高管理部门对策略问题的同意、支持以及批准是非常重要的。安全性必须要看做是一个非常重要的公司需求。同质量承诺一样，如果管理部门对策略问题有一个非常高的同意、支持和批准，那么策略就可以看成是重要的，并且会遵循。组织的管理部门必须具有实施策略的责任和权力。

组织内有很多部门不一定同信息技术领域相关，如内部和人力资源部门。但这些部门也应该涉及安全策略过程。这些部门能够提供具体策略的需求和内容输入。人力资源部可以参与商量对违背策略行为的处罚措施和后果。许多人力资源部门通过面向他们自己员工的程序来强化计算安全意识。策略应该包括在任何新员工培训材料中。内部审计可以评审策略，并在需要策略或者修改策略的地方指出问题。

6.4 小结

安全策略的框架是安全分布式计算环境中的一个非常重要的构建部件。它为员工遵循公司标准提供了指导，并且指出了如果他们不遵循所造成的后果。如果策略同已经接受的行为标准保持一致，那么策略可以增加组织的整体安全性。但是光建立策略是不够的。若要成功地实现安全框架，用户必须要知道策略并掌握其内容，从而支持安全框架的成功实现。

第三部分　技　术

最近几年来，采纳分布式计算可谓掀起了一股浪潮。几年前，很多组织还很少提到过分布式系统以及客户机-服务器技术，但是现在他们已经成功部署了基于这种技术的标准解决方案。然而，在得到客户机-服务器技术带来的高效率之前，安全和信任问题必须要首先解决。我们相信，知识是解决大多数问题的基础。为了解释其原因，我们将引入一个称做“雪地里的脚印”的概念。

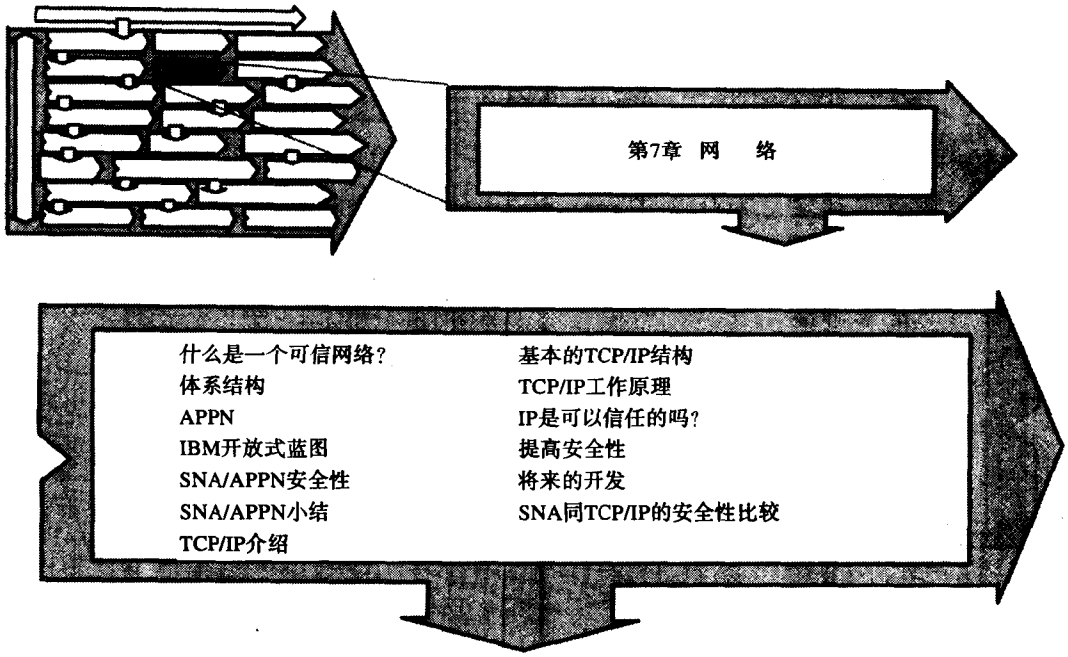
我们大多数住在郊区，那里并非安全之地。犯罪的事偶有发生。我们锁上了窗户和门，一条大狗为我们看门，我们相信它能吓住坏人。然而，如果你注意到了卧室窗户后面雪地里有脚印，安全意识可能会大大提高。你会采取积极的行动来检查入侵者。雪地里的脚印、地毯上的鞋泥以及门锁上的刮痕都是你应该关注的线索。

滥用计算机系统的人也会在雪地上留下脚印。问题在于，普通的网络或者系统管理员不会查看后院的雪地，并且即使他们看了，可能也不知道那些通往卧室窗户的脚印意味着什么。

在对分布式计算技术的分析中，将探讨和解释电子化的“雪地里的脚印”。将分析网络、中间件以及开放式系统环境的脆弱性和易受攻击点，并且要探讨一下加密技术。

还要讨论一下Windows NT的安全性和因特网的使用，后者是许多组织对安全性的关注所在。在本部分的最后一章中，将讨论一下开放式系统基础（Open Systems Foundation，OSF）中间件，这将为解决分布式计算问题提供帮助。在本章中，还要讨论事务处理的安全性使用。在每一章中，都将讨论技术为计算安全性问题带来的挑战以及它所提供的解决方案。

第7章 网 络



网络已经成为大多数计算环境的命根子。安全性过去一直集中于操作系统和应用程序两个方面，但现在必须要包括网络。因为网络在今天的计算环境中起着重要的作用，所以对网络安全性的关注是非常必要的。必须要有一些机制来保证网络的可信操作，这包括认证、完整性和授权控制。操作系统、应用程序以及网络的安全性不能互相脱离。它们都对计算环境中的整体信任水平起作用。一个领域的不足必须从另一个领域中补偿。例如，如果监听网络流量，那么应用程序可以通过加密流量来补偿这种缺陷。

为了理解为什么需要连网控制机制，来看一下什么使得网络可信，什么使得其不可信。

7.1 两个网络的故事

SNA和TCP/IP是两种不同的网络。现在看一下它们的安全性能。系统网络体系结构（System Network Architecture，SNA）是由IBM在20世纪70年代开发和引入的。它是现在仍保留使用的最常见的网络体系结构之一。现在，SNA已经演变成了高级对等网络（Advanced Peer-to-Peer Networking，APPN），它实现了分布式的、端级的连网。

TCP/IP协议的存在时间同SAN一样长，甚至更长。但是对于公司网络布置来说，它是个相对较新的后来者。它是使用最广泛的网络协议，用来互连遍布全世界的几百万台计算机。在本章中，将对SAN和TCP/IP环境的体系结构、安全控制以及问题进行比较和对比。但是首先要讨论一下可信的和不可信的网络。

在连网和安全员工之间的一个不调和的领域是开放式通道。在连网的开放式通道之下，所有连网的用户都可以对网络上的任何资源进行完全的和完整的访问。网络不应该对网络节点间的通信施以任何限制。然而，这个概念同许多业务客户的安全性利害有冲突。它会造成一个有趣的思想冲突。带来争论的解决方案通常要求平衡对安全通信的需要。如果开放式通道概念占上风，那么加强对关键服务器和应用程序的控制就是势在必行的。

可信网络的概念

一个可信的或者安全的网络是一个具有足够的安全控制以防止误用其操作的网络。它具有内部控制以保证网络的完整性。内部控制提供了保护，防止对网络的未授权访问与对网络流量的违法监听。对敏感网络流量进行加密是计数器监听的技术。这类网络一般是由政府、军事组织和国防工业所使用的。

不可信的网络是在其中，对网络的访问是未经授权的网络。对攻击者来说，攻击网络或者监听和解析流量是可能的。攻击者也可以插入他们自己的流量，并伪装它使其看起来像来自于一个合法源地址。或者他们可以简单地截断网络上的流量。

有很多种攻击网络的方法。攻击者可以带着一台具有适当接口的笔记本电脑，找到一个网络连接（或者在备用办公室里，或者拔下一个合法的工作站），然后连接到LAN上。连接上之后，他们可以命令笔记本电脑把网络流量截获到自己的本地磁盘驱动器上。因为一个不可信的网络上的用户之间通信一般是不加密的，所以用这种方式来截取和解析收集到的流量是很容易的。第二个原因是以太网之类的广播LAN技术的广泛使用。有很多公开可用的网络攻击软件，如有的软件能够守候单词“login”并有选择地拷贝下一百个字符。收集到的数据肯定包含尝试登录用户的账户名和密码。接着，攻击者可以登录一台合法机器，并开始访问那个用户所拥有的资源。通过得到对一台合法工作站的访问权，也可以按照此方式进行监听。

对于这种类型的网络来说，攻击者也可以在流量中插入自己的流量，并使其看起来像来自一个合法源地址。网络上的每台计算机都至少有一个标识这台计算机的地址。一个攻击者可以很容易的生成网络流量，并欺骗另一台计算机使其认为该流量来自于一个合法源地址，其方法就是使用那台机器的地址。

网络流量的恶意破坏可能会拒绝合法用户的服务。能够访问不可信网络的人都有能力生成大量的垃圾流量，从而使得网络不能工作。图7-1说明了使用不可信网络的时候所面临的安全问题。

下面，来看一下两种截然不同的网络——SNA和TCP/IP——是如何解决这些问题的。

7.2 系统网络体系结构

系统网络体系结构（Systems Network Architecture，SNA）是IBM在1974年发明的，它是对大规模计算机连网的回答。SNA是第一个进行如下工作的：用综合的体系结构来标识网络的所有组件以及这些组件的属性、性质以及行为的规范。1985年，该体系结构中添加了一些功能，它允许进行更低级的对等通信。1992年，IBM提出了开放式蓝图，它描述了分布式系统环境的结构。现在，该体系结构已经从主机中心网络转成了分布式端中心网络。

该体系结构使用了会话的概念。当两个网络节点互相联系，并且一致同意他们进行通信的原则，那么我们就说他们建立了一个会话。两个网络节点之间可以有许多活动会话。该体系结构也是建立在分区（围绕其来定义网络）概念上的。一个SNA网络中的每个大型计算机和通信控制器都有一个分区号。

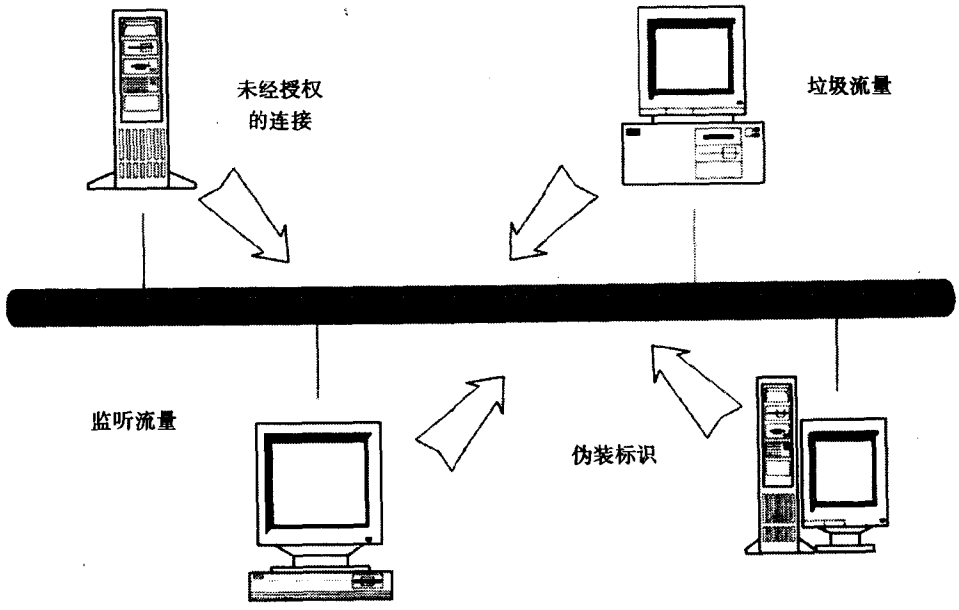


图7-1 不可信网络的问题

该体系结构最初建立成一个分层网络，并由虚拟远程通信接入方法（Virtual Telecommunications Access Method，VTAM）进行管理。VTAM是控制IBM大型计算机上网络操作的应用程序。如果VTAM不知道或者不担心，那么网络上什么也不会发生。VTAM知道一切，看到一切。子系统控制点（Subsystem Control Point，SSCP）是SNA网络的控制主。SSCP控制着从初始化网络到帮助建立会话的一切网络活动。

7.2.1 体系结构

SNA是建立在一个同OSI模型类似的七层模型基础上的。开始是物理层，它定义了网络节点是如何连接的（线路规格等）。数据链路层包含了网络节点间的传输协议，本层把数据发送到物理层以进行传输。网络或者路径控制层提供了路由机制和管理员预定义的网络路由器。传输或者传送控制层调整一个网络节点向另一个网络节点传输消息的速率。如果启用了相关的选项，就会以此模式进行加密和解密。会话或者数据流控制层调整消息到最终用户的传递。表示层管理两个网络节点之间的数据流，它可能会进行压缩或者翻译。应用层管理网络上应用程序的接口。

SNA下的大量通信都使用一个面向连接的传输，同步数据链路控制（Synchronous Data Link Control，SDLC）协议。SDLC是一个面向比特的协议，它把信息组织成定义良好的单元（称做帧）。一个面向比特的协议把数据作为一个比特流而不是字节流来进行传输。这种类型的协议可以使用特殊的比特流作为控制码，从而代替面向字节协议中使用的保留字节。网络

路由遵循固定的路径，这些路径是在网络初始化或者具体路由组件激活的时候设置的。流量控制是由具体服务的类别管理的。用于多个网络会话的数据流量可以共享一个网络路由。传输优先级可以在服务类别的基础上给定。因为SDLC是一个基于帧的协议，所以打印流和在线事务可以共享同一个流，这不会对在线服务产生影响。SDLC协议提供了传输完整性，从而保证所有的消息帧都会传输并且按照正确的顺序汇合。

体系结构中定义了节点概念，节点分为两种：分区节点和外围节点。有很多种节点可用来描述不同的网络特性及其拥有的质量。用于所有网络活动的根是5类节点，它驻留在一个主机计算机内。链上的下一个节点是4类节点，或称为通信控制器。这是把所有的网络通信线路都联在一起的硬件单元。通信控制器是通过一个程序加载的，该程序称作网络控制程序（Network Control Program，NCP），它定义了连接网络的配置并管理通信线路。大量的低级网络活动都从大型机转到了通信控制器上。5类节点和4类节点都是分区节点。图7-2说明了体系结构层次图中各个节点的关系。

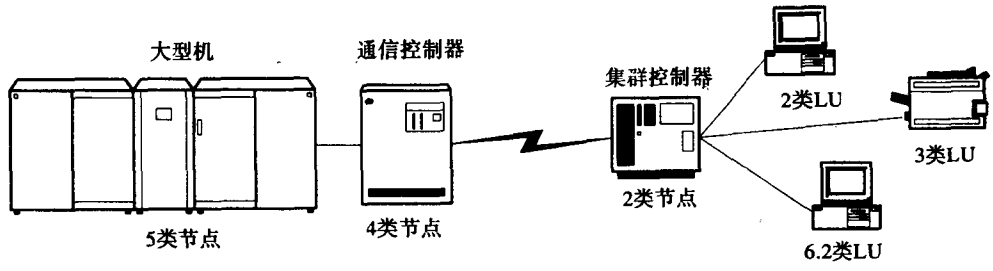


图7-2 SNA网络

位于通信线路另一端的是2类节点或者集群控制器。这是位于布线室内的盒子，也可以位于引出所有同轴电缆的地面。连接到集群控制器的是网络访问终端，这可以是终端、PC和打印机。那么3类节点在哪里呢？3类节点曾经看起来很好，但是在体系结构发布之前其规范就做废了。

网络可访问单元（Network Accessible Unit，NAU）是体系结构的另一个基础组件。NAU实现了体系结构的上四层、事务服务、表示服务、数据流控制以及传输控制。NAU可以定义为通信的实体，而体系结构的下三层定义了NAU是如何通信的。有三种不同类型的NAU，物理单元（PU）——激活和管理链路；逻辑单元（LU）——定义了支持的SNA功能的类型或者特性；控制点（CP）——管理域内的网络资源。我们已经讨论了SSCP，它控制来自一个5类节点的网络资源。

体系结构也包含了对逻辑单元（LU）的定义。1类LU是用于打印机和远程作业入口站的；2类LU是用于3270类型终端的；3类LU是用于一个3270类打印机的；4类LU是用于智能型打印机的；作为对体系结构的扩展，6.2类LU出现较晚，它指定了为通信或者APPC编程的应用程序。在SNA术语中，可以把PU看成控制设备的硬件，把LU看成PU所控制的设备。

体系结构的另一个基础是会话的概念。会话是网络中两个NAU之间的逻辑连接。两个NAU之间的物理连接定义成一个用于会话的路由。会话在概念上同登录到应用程序上类似。一旦登录到了一个应用程序之上，就已经建立了如何同该应用程序通信的规则。一旦两个NAU之间建立了一个会话，那么它们之间通信的规则就建立了。

应用程序到应用程序的通信

作为体系结构的一个附加，规范提供了网络计算机上应用程序到应用程序的通信（Application Program-to-Program Communication，APPC）。APPC是开发用来支持两个智能端点间的分布式处理的。它定义了对应用的一个全方位服务，其中包括同步选项、进程激活、确定选项以及错误管理。APPC是建立在LU 6.2规范之上的，该规范已经包括进了很多OEM和第三方产品中，从而允许同基于IBM的网络应用程序进行基于应用的通信。APPC是用于CICS OLTP产品中分布式事务处理的通信工具，它是支持SNA进程间通信的主要工具。

用于通信的公共编程接口（Common Programming Interface for Communication，CPI-C）是一个进步，它使得开发人员可以使用一个对底层LU 6.2结构的公共接口来方便地开发APPC应用程序。该公共接口也使得应用程序只需要最少的改变就可以从一种平台移到另一种平台上。当一对事务程序都使用APPC的时候，两个6.2类逻辑单元间的会话就建立了，这称是对话。对话表示了两个程序间的逻辑连接。APPC是一个机制，它在两个网络应用程序间实现事务级上的真正分布式处理。

7.2.2 高级对等网络

1986年，SNA引入了另一项扩展，高级对等网络（Advanced Peer-to-Peer Networking，APPN）。它最初是为IBM的微型计算机市场引入的，其目的是在不需要大型机的情况下提供计算机间的互连。APPN引入了一个新类型的物理单元——2.1类PU，它使得对等网络成为可能，而代替了对分层方法的需要。设计APPN来处理主机应用程序和分布式平台间的同批类型或者会话式通信中的事务处理。操作一个网络不再需要5类节点。这是从早期层次结构规范到对等网络的一个重大革新。

随着高性能路由（High Performance Routing，HPR）的引入，APPN在1993年迎来了一个更大的发展。HPR带来的基本改变是在一个更低的级别而不是前面的APPN级别上提供中间节点间的高速路由。HPR由两个主要组件构成，快速传输协议（Rapid Transport Protocol，RTP）和自动网络路由（Automatic Network Routing，ANR）。快速传输协议控制数据的高速传输。自动网络路由是一个高速源路由协议，它在数据包的网络头携带整个路由信息。HPR定位于利用交换网络技术，如帧中继（Frame Relay）或者异步传输模式（Asynchronous Transfer Mode，ATM）。

APPN有两种已定义的设备，APPN终端节点和APPN网络节点。前者是具有操作系统、应用程序以及外围设备的计算机，后者是在终端节点间移动流量的中间路由节点。许多平台都可以操作为终端节点或者网络节点。APPN支持很多网络协议和拓扑结构。APPN网络地址是字母数字混排的，在一个APPN子节点中可以包括多达10 000个节点。通过自动注册功能，网络节点能够自动学习端口、链路以及资源的细节。APPN是能够取代SNA的新型网络结构。有些人认为它已经取代了。

7.2.3 IBM开放式蓝图

1994年，IBM定义了开放式蓝图，这是该公司下的大赌注。开放式蓝图是对几个资源管理服务的描述，建立在这几个服务上的分布式应用程序在分布式、异构的环境中操作。该蓝图的基本方法是允许网络操作系统可以像一个连网操作系统那样操作。同其他蓝图一样，

IBM开放式蓝图把各种资源管理器组织到了服务层中。

基础层代表物理网络。网络服务层包含了各种网络协议，如SNA/APPN、TCP/IP、OSI、NETBIOS、IPX，以及公共传输语义（Common Transport Semantics）。分布式系统服务层包含了通信服务、对象管理服务以及分布服务。设计这些服务来提供分布式应用程序和资源管理器间的机制。应用程序和应用程序启用服务提供了一套资源管理器以处理许多不同专题，如表示服务、数据访问服务、系统管理服务、本地操作系统服务以及开发工具等。

开放式蓝图并不是产品或者标准的一个有限规范，它是一个框架，对说明分布式计算环境中关键功能和标准间的关系是非常有用的。IBM承认，在分布式世界中有很多不同的技术和标准都占有一席之地，并且还可能有更多。开放式蓝图是把这些技术和标准组织到框架中去的方法，并且也是一个解释它们之间关系的方法。

7.2.4 SNA/APPN安全性

用SNA实现的层次化网络中，安全性是容易处理的。外部设备不可能添加到网络中。通过指定安全包，如IBM公司的资源访问控制工具（Resource Access Control Facility，RACF）和CA（Computer Associates，计算机协会）的访问控制工具2（Access Control Facility，ACF2），主机e提供了严格的安全控制。当有一个中心点控制网络的时候，对等网络（如TCP/IP）所包含的许多脆弱性都避免了。如SDLC和RTP之类的协议提供了网络传输完整性。协议的实现包括了传输错误检查和校正工具。分区节点间的动态路由和恢复能力提供了网络流量的高可用性。

数据安全性可以在逻辑单元级上提供给会话级的加密、逻辑单元到逻辑单元的校验以及终端用户校验。会话级的加密可以是强制的，可以对所有的数据都进行加密，也可以只对所选择的数据字段进行加密。会话级加密和LU到LU校验都使用数据加密标准（Data Encryption Standard，DES）加密协议。逻辑单元到逻辑单元的校验是建立在会话激活过程中LU提供的密码的基础上的。终端用户校验用来对终端用户进行认证，其方法是使用密码。在对话分配过程中，一个6.2类对话将校验该密码。

APPN和APPC可以在会话和对话级上提供安全服务。在两个终端节点之间建立一个会话之前，APPN先要检查用户提供的密码。为了同用户建立对话，应用程序本身也必须要提供一个正确的密码。APPN和APPC安全性可以在网络级上提供。大多数其他安全性机制都要依靠应用程序级的安全性。

7.2.5 SNA/APPN小结

SNA的主要缺点是：它是IBM的一个专利体系结构。它同IBM硬件、软件以及其他与IBM网络交互的实体工作得非常好。SNA并不是用来在一个异构环境中进行互操作的。该体系结构历经了几次转换。对于APPN提供的对等网络模型和SNA的层次模型，人们存在着争议。APPN的改进让IBM重新回到了主流的分布式环境中。已经部署SNA的组织可能想采用其他分布式系统，对于他们来说，APPN是非常有吸引力的。

开放式蓝图表明了IBM对开放分布式环境的承诺。该蓝图出现稍晚，但是它非常全面，并且可能有点雄心勃勃。开放式蓝图代表了IBM的新动向，控制开放分布式市场中的每一个人。该蓝图的成功将依赖于IBM对采用该蓝图产品和服务的承诺。该蓝图会受到其他标准演

变的影响，特别是TCP/IP。

7.3 TCP/IP介绍

TCP/IP是美国国防部开发的，它是DARPA——国防部高级研究计划局（Defense Advanced Research Project Agency，ARPA）的一个项目。其设计宗旨是通过使用包交换技术在网络中提供冗余。其目的是创建一个即使在一段网络出故障的情况下也能继续保持功能的网络协议。该协议允许确定多个路由器并用于一个目标地址。如果一个路由器出现故障，该协议可以支持流量的自动重路由。TCP/IP的第一个大规模部署是ARPANET，这是当前因特网的前身。

IP、TCP以及UDP

TCP/IP是一个广泛使用的术语，它描述了因特网e许多不同的协议，其中有IP、TCP、UDP、ARP以及ICMP。因特网协议（Internet Protocol，IP）是前三个协议中的低级协议。它为上面的TCP和UDP信息块（通称为包）的传输提供了基础。一个IP包中包含了一个源地址和一个目标地址，以及其他包完整性信息和数据。IP级对包流没有任何顺序控制（如后发的包可能先到）。这是上层协议或者应用程序的工作，它们需要正确的收集并汇总数据包。

传输控制协议（Transport Control Protocol，TCP）位于IP协议之上，它提供了顺序以及完整性控制。TCP使用序列号来保证包按照正确的顺序重组，并且重发丢失的包。因为这些完整性控制，TCP当然会有一些开销。

用户数据报协议（User Datagram Protocol，UDP）是另一种传输替代方案。UDP具有有限的完整性检查和顺序控制。包在传输过程中有可能丢失，或者到达顺序有误。应用程序可自由处理包的定序和重传问题。有些应用程序需要那么做，而有的则不需，所以UDP是一个好的起点。UDP通常用于消息和挑战/响应应用程序，这类应用不要求完整性检查或者顺序控制。大多数UNIX系统用来共享磁盘资源的网络文件系统（Network File System，NFS）就是基于UDP的（但是它也可以运行在TCP上），很多人对此十分诧异。NFS知道UDP的限制，它添加了额外的控制以保证数据传输的完整性。

地址解析协议（Address Resolution Protocol，ARP）是用来转换两个不同类型地址的方法。例如，ARP可以把高级的IP地址转化成以太网（或者链路级）地址。其他类型的LAN，如FDDI，使用了其他机制，如SNAP。在某种程度上，ARP更像是以太网而不是IP的一部分。以太网地址是硬件设备（如LAN卡和网桥）使用的低级地址。ARP提供了一个IP地址到硬件地址的映射。一些替代协议，如RARP和BOOTP，提供了一个硬件地址到IP地址的逆向映射。

> TCP与UDP、信使与邮政服务之间可以进行一个对比。UDP与使用一项邮政服务来交换明信片类似，而TCP则类似于使用一个信使服务。当你使用邮政服务或者一个信使来发送大量包的时候，包裹是否会到达其目的地是没有保证的。可能有一个，也可能全部都丢失，这同所选择的方法无关。但是一般来说，100次中有99次会到达！如果一个邮寄包裹没有到达，那么接收方可能根本不知道它丢失了（除非发送方告诉了他会有3个包到达，而他只发现了2个）。即使你知道丢失了一个包裹，你也没有什么简单办法去跟踪那个包裹。
>
> TCP同使用一项信使服务类似。在这种情况下，每个包裹都附了一封信，指派了一个控制号，并且记录了该包裹中物品的个数。传递过程事先通知给了接收方，并且他知道会到达多少个包裹。毋庸置疑，这项服务肯定要比使用邮寄服务贵。

简单网络管理协议（Simple Network Management Protocol，SNMP）是一个更高级的协议。它同时使用TCP/IP和UDP/IP来在SNMP代理和一个SNMP管理者之间进行通信。SNMP被用来提供很多种网络管理服务。一个本地系统或者设备会包含一个管理信息基础（Management Information Base，MIB），这包含了本地SNMP代理所维护的信息。使用SNMP协议，SNMP管理者会查询，或者能够更新这些变化的信息。

因特网控制消息协议（Internet Control Message Protocol，ICMP）是被一个IP网络用来监视和控制网络上之流量的。例如，ping服务就使用了该协议，该项服务可以确定一个远程目标地址是否为可达到的。该协议也被用来提供路由信息，或者作为一个消息传输。

7.3.1 基本的TCP/IP结构

使用TCP/IP通信要跨越五个独立的层次，其中每个都有其自己的责任。它们是应用程序、传输、网络、链路和物理层。与客户交互的是应用层，它可以是一个应用程序、一个操作系统工具或者一项TCP/IP网络服务。后者包括终端会话以及文件传输和打印服务。传输层用来保证传输数据的完整性，并为流量的移动做准备。数据包的顺序和重组在该层进行。网络层的责任是通过网络为包选择路由。它实际上提供了一个空中交通控制功能，保证了包能够找到其目的。然而，网络层没有一个联邦范围内管辖的空中交通控制系统所具有的任何权力。链路层的任务是保证数据在网络节点间的正确传输。应该指出，TCP/IP可以运行在多种通信链路上，这包括以太网、令牌环网以及串行线路。物理层是由支持TCP/IP（但不是TCP/IP的一部分）的一些硬件设备组成的。它包括线缆和网络设备，如集线器、网桥以及路由器。

TCP/IP的基本通信单元是包。包具有多种格式和大小，这取决于包括所使用的协议在内的很多因素。图7-3说明了包含在IP包和TCP、UDP包中的信息类型。

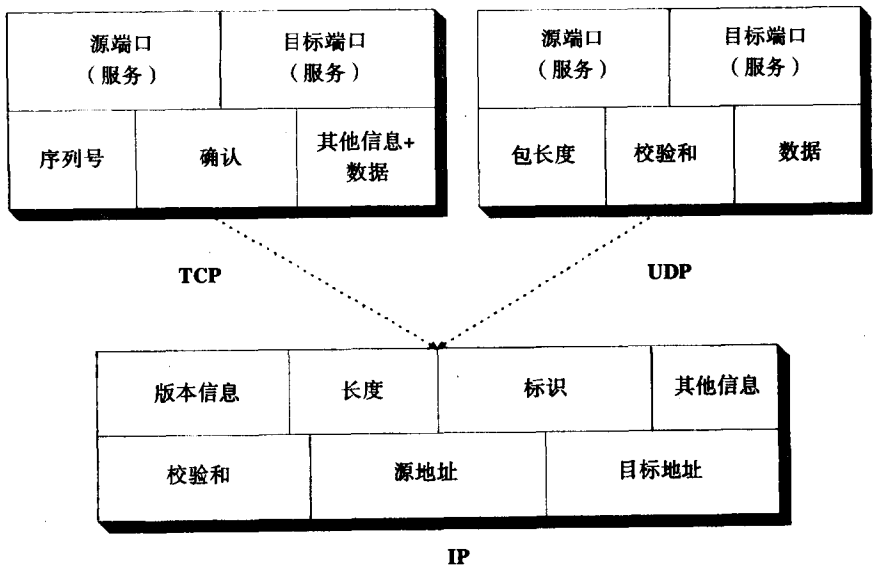


图7-3 IP、TCP和UDP的概念视图

7.3.2 TCP/IP工作原理

为了理解TCP/IP的工作，我们来看一下一个简单的telnet会话后所隐藏的东西。Telnet是一个TCP/IP服务，它允许用户同一台服务器进行远程终端会话。如果一个用户想从他的机器my.own.box.com启动一个到服务器where.am.i.com的telnet会话，那么它可以输入

```
telnet where.am.i.com
```

服务器的名字是where，它位于suborg.i.com域中。其中的com表明一个公司（其他如gov代表政府，edu表示教育组织），i是公司指示（如hp代表Hewlett-Packard），suborg是次项（如研究）。如果发现的话，服务器会响应，要求提供一个有效的用户ID及其密码。这两项信息都输入了以后，终端会话就开始了。这看起来很简单，但是后面要进行很多工作。图7-4说明了在终端会话可以开始之前必须要进行的许多操作。

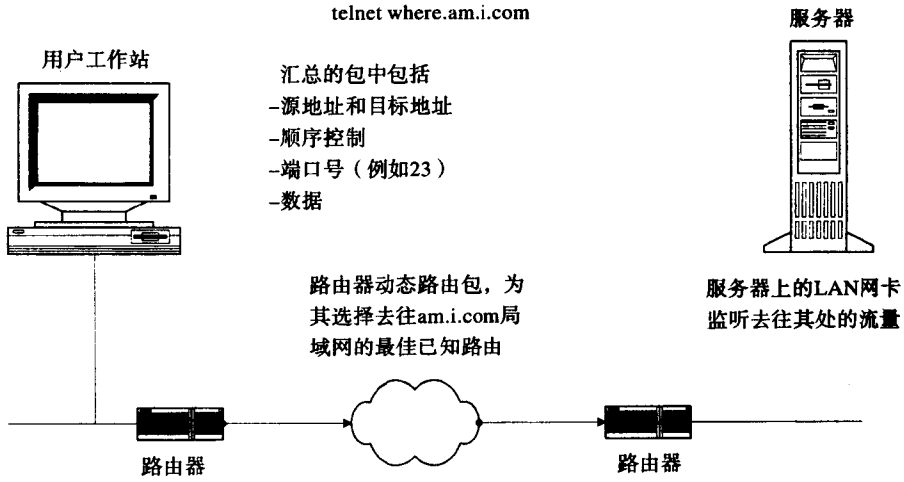


图7-4 TCP/IP工作原理的一个例子

现在我们看一下这个连接是如何建立的。首先，我们不想为telnet服务发送一个六字符的请求。如果我们用一个两字节的数字来引用telnet，那么在时间和空间上效率都将大大提高，这是因为ASCII字符串比较要比数字慢。在TCP/IP的术语中，这个数字称做端口号。用于telnet的端口号是23，并且由于23一般是用于telnet的，所以它称为一个已知端口。TCP/IP实际上会同时使用目标和源端口号，它允许客户机和服务器间的对话使用未用的端口。根据TCP/IP服务，服务器用来回答客户机的端口可能只在连接时间才确定下来。如果我们希望为基于某个端口的服务中断流量，那么这会带来复杂性，其原因在于不能精确地确定服务器可能会使用哪个端口来响应。

端口号不是惟一需要转化的东西。服务器的名字where.am.i.com必须要转化成一个32位的IP目标地址。机器名my.own.box..com也要转化成一个源IP地址。IP地址是由系统管理员指派的，但是并不保证它在组织内的惟一性，更别说在世界范围内了。这些地址对于每台机器来说应该是惟一的，但是TCP/IP没有任何办法来保证这种惟一性。然而，有中心管理局（NIC和InterNIC）来注册因特网地址和在一个组织外可见的域名。在一个独立组织内部，一般有一个中心管理部门（例如网络组）来处理和控制IP地址的使用。但是需要指出的是，并没有关

于地址的强制规定，这完全在本地系统管理员的控制之下。

用于已知机器的IP地址可以保存在本地系统上。但是更常见的做法是使用一个IP查询服务把机器名转化为一个IP地址，其方法有域名系统（Domain Name System，DNS）或者网络信息服务（Network Information Service，NIS）。

此外，由于IP地址不用在链路层上，所以还需要进行另一项转化工作。链路层使用一个独立的地址，它称做链路层地址。链路层地址是由LAN卡的生产商提供的。尽管这些地址对于每个卡来说应该是惟一的，但是它们也可以改变。UNIX和DOS系统都允许系统管理员修改链路层地址。用一个专门的工具来把IP地址转化成链路层地址。对于以太网来说，用地址解析协议来执行此操作。

包正确地格式化后，它就通过LAN卡在LAN连接上进行传输。去往服务器的实际路由可能会变化，其原因在于TCP/IP网络不断的发现并使用去往远程网络的新路由器。如果本地路由器不知道去往am.i.com子网的直接路由，那么它会把包转发给一个默认路由器。路由器永远也不会猜测下一个目的地。它们或者有一个去往目的网络的明确路由，或者有一个用于“其他一切”的默认路由。该默认路由器可能依次把包传递到另一个路由器上。如果在一个可接受的时间内没有发现任何路由，那么就会返回给用户“网络不可到达”的错误。目标服务器在其LAN卡上不断地监听，当以它为目的地的telnet包路过时，LAN卡就会接收该包。然后，服务器检查该包，并响应服务请求，发送一个包来初始化登录顺序。一个返回包可能会也可能不会沿着初始请求的路由。两台机器要交换包含登录请求、用户ID、密码提示以及真正密码的包。所有交换的数据，包括用户ID和密码在内，都是以人类可读的形式保存在数据字段中的。

7.3.3 网际协议是可信任的吗

绕过使用TCP/IP和UDP/IP协议的操作系统和应用程序所具有的安全性是很容易的，有许多方法可供选择。如果没有补偿控制（如包加密），这些协议对于下面的操作来说是非常脆弱的：

- 包监听。
- 泄漏。
- 伪造地址（欺骗）。
- 顺序攻击。
- 路由攻击。
- 拒绝服务。
- 诊断地址。

1. 包监听

TCP/IP协议不会阻止LAN上的任何用户，所以谁都可以跟踪包。有大量公开软件可用于此目的，另外厂商也提供了很多诊断软件，它们都可以用来截获和分析网络流量。数据包的内容也可以检查并修改。

使用这些工具来截获大多数密码是很容易的，这是因为密码大都以明文的形式进行传输。有些操作系统和中间件服务，如Windows NT和Kerberos，使用其自己的加密服务来保护用户密码。但是即使密码加密了，攻击者也有可能认出一个包是属于一次登录过程的。攻击者可以拷贝密码包，然后修改源地址，接着把该包转发到重放攻击的服务器上。

2. 泄漏

当一个网络设备错误地提供了信息，如网络地址或者服务时，就发生了泄漏，这些信息对一个不可信网络应该是保持隐秘的。导致泄漏的最常见原因包括网络设备的不正确设置以及在重新配置网络设备上的访问列表时不注意暴露了网络。大多数路由器上的访问控制单都是应该严格进行配置的。很多网络管理员都曾经因为一个简单的配置错误而把一个受保护的网络暴露给攻击者，当发现以后，他们无比震惊！

3. 伪造地址

许多TCP/IP服务，包括伯克利“r”命令、NFS以及X Window在内，都要在一个网络标识的基础上对客户进行授权。这种类型的授权通常都要牵扯到访问控制列表的使用。访问控制列表在客户机器名或者IP地址的基础上对其授权。问题在于源地址以及包内的其他所有字段都是可以改变的。在利用这种脆弱性的地方，大量的最近攻击都记录下来了。

现在问题来了：“如果源地址伪造了，那么攻击者如何从攻击中接收任何信息？”答案是，在许多情况下，攻击者会利用一些不需要响应的技术。例如，攻击者可以把密码文件e-mail给他的攻击，这种攻击不需要一个有效的源地址或者任何应答包的接收方，它只需要一个有效的电子邮件地址。

技术并不限定于软件服务，它还可以用来对付网络设备，如路由器。作为防火墙配置上的过滤器，路由器可能会被欺骗，从而允许来自外部的非法包进入内部网络。假设路由器遇到了一个其源地址来自于内部网络的包，但该包可能是来自内部网络之外的，路由器就被欺骗了，从而允许该分组进入内部网络。如果一个路由器上有三个或更多个LAN卡，并且其中之一连接到一个外部网络，那么问题更加明显。该路由器应该放行内部流量而过滤外部流量，但是它可能会被欺骗从而放行伪造的流量。

> 我们可以把多LAN卡路由器的问题和一个不对来自国内的和来自国外的乘客进行隔离的国际机场进行一下比较。确定哪些乘客来自国外并把这些乘客同国内旅客区分开来可能是非常困难的。能够过滤传入包的路由器可以极大地克服这些问题。

4. 顺序攻击

一些TCP/IP服务使用随机数字来确定它们在通信和认证过程（有些情况下）中的位置。连接建立以后，客户机和服务器共享一个公共的TCP/IP序列号。这里使用了两个数字，一个通知客户机应期待服务器上的哪个帧；另一个通知服务器应期待客户机上的哪个帧。当客户机向服务器发送了一个顺序请求的时候，它将提供共享的序列号作为通信顺序的证明。问题在于在有些实现中，序列号不是随机的，因此可以猜测或者计算。这就使得攻击者可以绕过正常的认证过程，其方法是提交一些包来愚弄服务器以使其认为认证已经进行了。

例如，攻击者可以构造一个请求并发送到一个支持伯克利主机等效性服务的UNIX系统上（该服务会使服务器相信，即使没有进行认证客户机也是可信的）。UNIX X Window和网络文件系统服务，以及Novell Netware LOGIN认证，都曾受到过这种攻击。尽管这不是一个TCP/IP网络问题，并且它属于使用这种模式的应用程序服务，但是它仍然是一个同TCP/IP的使用相关的问题。

5. 路由攻击

TCP/IP网络不断地发现去往他们希望与之通信的网络的新路由。路由器传递外部网络的

信息，并且只要在合适的时候它们就会建议去往那些网络的新路由。从一个不可信的源地址接受的路由信息应该怀疑。对于那些用于内部网络的路由信息来说，这更要注意。

> 这里可以做一个对比。假设有一个农村的孩子要从镇上回家，他向人打听回家的最佳路径。如果有人向他推荐了一条路，但是这条路必须要通过一条荒废的市区小径，那么这个孩子可能会起疑。如果这个孩子还带着钱包的话，他可能会更加怀疑！

6. 拒绝服务攻击

拒绝服务攻击就是怀有恶意地试图阻止其他人获得对服务的使用。攻击者可以使用很多手段，如把大量的邮件消息发送到某个邮件地址从而导致磁盘空间问题。或者，生成大量的垃圾LAN流量从而导致网络不可用。

7. 诊断攻击

大多数TCP/IP机器都关联着一个指定地址。该地址用于提供对LAN卡的本地诊断支持。当独立运行时，许多网络服务也使用本地回环地址。例如，一个DNS解析器可以使用回环地址来同运行在同一主机上的名字服务器通信。通常该地址是127.0.0.1，并称做回环地址。因为攻击者知道该地址存在于许多机器上，所以他们过去曾操纵过它。实际上，任何去往回环地址的网络通信都应该看做是无效的，路由器和网桥应该丢弃这些包。

7.3.4 提高IP网络的安全性

解决不可信网络的安全性问题可以采用对敏感网络流量进行加密的方案。不幸的是，对网络流量进行加密需要很高的成本。加密设备成本不菲，并且加密也会带来性能影响。

用来增强IP网络安全性的最常用方法是把网络化分成多个子网。路由器和网桥可用来把网络分割成多个部分（子网）。在互连的网络之间流量可以过滤，而大多数流量保持在本地工作组中。路由器和网桥把目标不是外部网络的流量限定在本地工作组中。子网控制的实现阻止了对网络流量的监听，其原因在于限制了可以从一个节点上监听的流量数量。

图7-5说明了如何在TCP/IP网络上实现附加的安全性。

集线器可以配置成只允许对预定义的硬件地址进行访问，从而防止了对LAN的未授权访问。如果发现了新设备，集线器也可以把这一变化情况转发给网络操作软件。网络中也可以使用能够进行加密和保护的安全路由器。这些路由器可用来保证一个暴露连接（如两个建筑物之间或者同一个远程子网的连接）上流量的完整性和机密性。还有其他一些办法，如建立一个询问-响应认证，特别是针对于通过一个因特网防火墙的IP欺骗。我们将在第14章中介绍防火墙和其他因特网安全技术的使用。

重要的是，网络设计者应该具有安全性意识，从而在网络的可信和不可信部分建立边界。路由器和网桥提供了对来自网络不可信部分的未经授权的流量进行过滤的能力。不仅仅来自不可信网络的流量可以过滤，更常见的做法是在服务类型（如端口号）的基础上进行过滤。例如，如果产生于本地工作组之外，那么所有的X Window流量都可以阻止。

子网化还有其他一些好处，如建立匿名管理域的能力。一个子网地址范围内的IP地址可以分派给一个本地工作组。例如，可以为一个软件开发组分配某个范围的地址。由于软件开发组需要高级的特权和环境的不稳定性，所以通常他们没有太严格的安全控制。可以采用一个可信子网方法来允许开发工作组进行任意行为，但要把其同内部网隔离开。这防止了松散

安全性的滥用——许多开发环境都有这个问题。

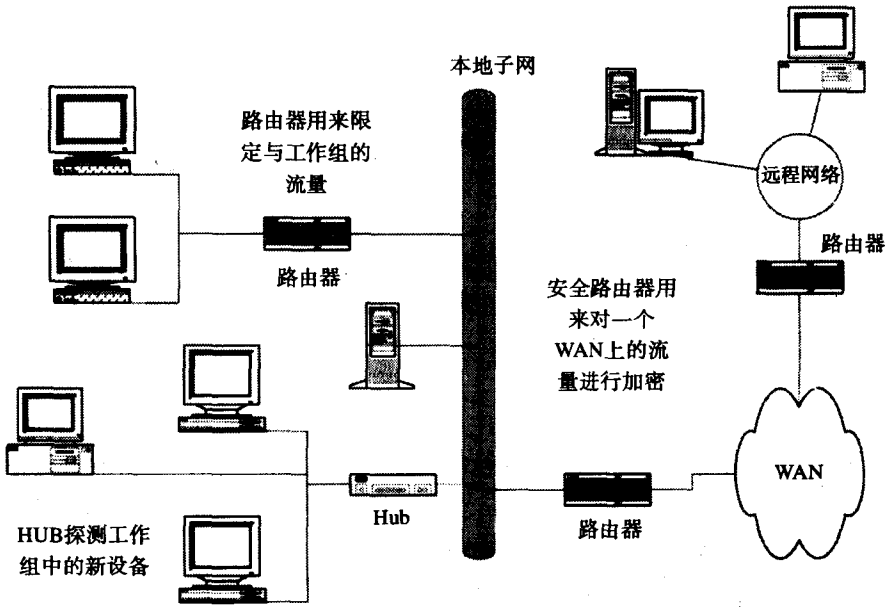


图7-5 提高一个TCP/IP网络的安全性

> 提高网络性能和安全性不是两个互相排斥的目标，这是个好消息。用于实现一个目标的方案和技术也可以用于另一个目标。子网的使用勾画出了网络的前景。大多数网络流量都符合老电话公司Pareto原则或者“80/20”原则——80%的通信是本地的，20%是通往外部的。如果这一规则可用于网络流量，那么尽量把流量限定在本地子网内部就是非常可取的。这会提高网络其余部分的性能。

构建可信子网的趋势是路由器必须允许一些流量通过——如果什么都不通过的话，那么连网是为了什么呢？但是过滤器应用到了可能伪造的IP地址。过滤模式有点难以实现。这样网络就有漏洞可用。可信子网的管理也需要很多工作，并且可能会用到一些非常复杂的网络管理工具来提供支持。还有，使用一个可信子网方法并不会阻止来自子网边界内部的攻击。从可信子网内部到不可信互连网络的流量仍旧是暴露的。

尽管存在着这些局限性，许多组织仍旧成功地把他们的网络分成了多个可信子网。这是否是一个合适的方案要取决于当前连网计算的使用以及企业的文化。

7.3.5 将来的开发

现在TCP/IP世界面对着一个重要问题。由于该协议在因特网上的巨大流行，现在可用的IP地址快用光了。为了给该问题提供一个解决方案，一个称做IPng（IP Next Generation，IP下一代）的工作组已经建立了。其解决方案称作IP第6版（IPv6）。在该方案中，因特网地址从当前的32位扩展到了128位，并且该方案还有一些其他的改进。这种地址扩展会极大地增加可用地址的数量。然而，存在一个问题，要求IPng解决许多其他的问题。这就增加了IPv6的复杂性，并推迟了IPv6的发布日期。IPv6中也涉及到网络安全性问题，它使用了一些方法来

认证一个包的源地址并为包内容提供加密。

因特网协议安全组（Internet Protocol Security Group，IPSEC）正在评审一些安全增强机制，其中包括认证控制、加密、校验和以及访问控制功能。该体系结构将对公钥加密技术起到了重要作用，它由一个密钥管理和安全封装协议组成。密钥管理协议会生成加密密钥，并对传输的发送方进行认证。安全封装会保护流量，并提供各种级别的机密。此技术标准已经在Internet RFCs 1825~1829中进行了文档化。

7.4 SNA同TCP/IP的安全性比较

这是两种根本不同的网络。SNA对所有的网络连接都进行授权，而TCP/IP连接是不授权的。SNA采用层次化设计，而TCP/IP是对等式的。SNA使用固定路径，而TCP/IP使用动态路由。SNA的性能是高度可预测的，并且能够管理。TCP/IP的分布式体系结构导致网络响应时间的变化很大，这取决于网络流量。然而，TCP/IP可用于很多种平台上，并且容易进行设置，也容易同其他计算机进行通信。此外，世界上最大的网络——因特网——是基于TCP/IP的。尽管层次化的SNA会对SNA网络的所有连接都进行授权，但这并不意味着一个SNA网络的安全性就不能被破坏。使用一个SNA协议分析器，就可以截获和分析SNA流量。同TCP/IP一样，SNA网络中的密码也以可读的形式进行传输。SNA不提供数据加密或者保护服务，它们不是传输的一部分。

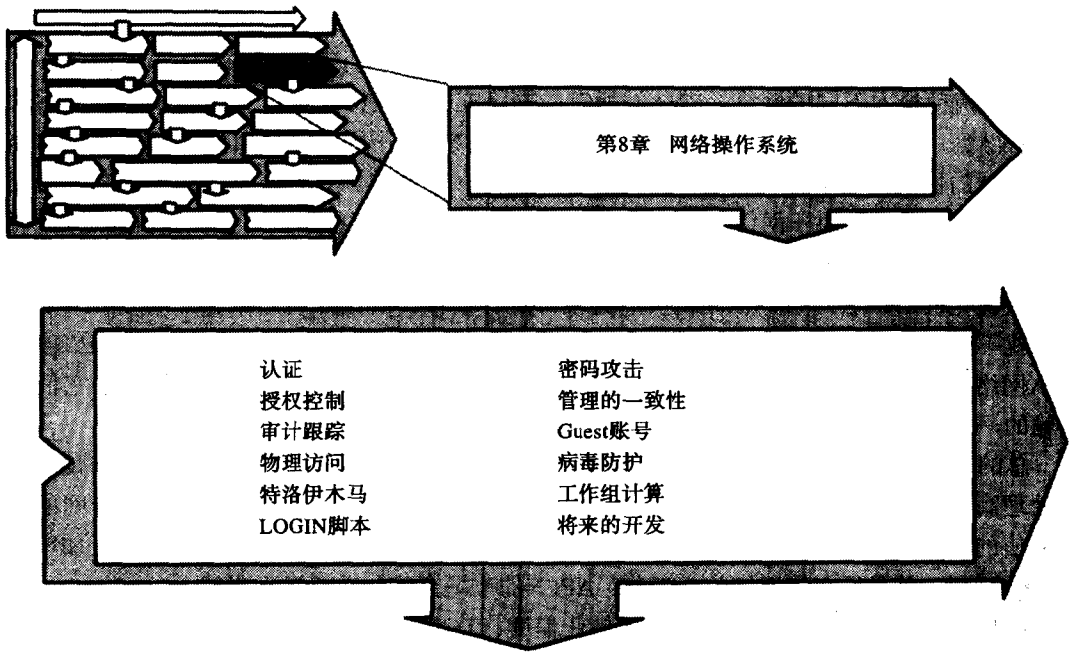
总的来说，SNA/APPN体系结构的安全性不像TCP/IP中的那样容易破坏。APPN是建立用来处理高端的、关键任务的、分布式的应用的，TCP/IP是发明用来处理匿名管理域间的大范围通信的。尽管SNA有一个明显的已建立的基础，但TCP/IP在计算世界中确立了更加牢固的“市场地位”，并且仍在快速增长。当前，APPN的市场突破和增长尚不明朗。

如果要达到SNA网络环境所具有的信任程度，TCP/IP环境需要提供另外的安全功能。直到现在，所需安全控制是建立于协议中，还是设计进使用该协议的服务和应用程序中，还是不明朗的。并且也有可能采取其他方法。

7.5 结论

在机制可用于提供对监听和其他入侵行为的保护之前，我们有两个选择。可以接受风险，并等待现有协议的改进；或者可以使用补偿控制来降低风险。补偿控制有很多方法，如把网络划分成多个可信的子网等。另外也可以把加密网络设备（如安全路由器）使用到运行在一个不可信的载体上的网络连接段中。并且，也可考虑安全中间件的使用，如OSF/DCE，它假定网络是不可信的，并提供很多针对入侵者的额外控制。将在第16和17章中探讨OSF/DCE及其安全机制。最后，也可以考虑使用商业访问控制方案来替代当前操作系统认证和访问控制机制。我们将在后续章节中讨论这些方案。

第8章　网络操作系统



网络操作系统，如Novell的Netware和Banyan的Vines，已经成为很多组织计算环境中的一个共同元素。为网络操作系统下一个实际的定义是非常困难的。毕竟，每种操作系统都有一定的网络交互能力。我们要定义一个网络操作系统（NOS），并不与今天它们的位置有关，而与它们的产生地有关。为了给局域网上的用户提供资源共享和主要的磁盘和打印服务，网络操作系统应运而生。这种计算的着眼点是本地工作组。老的定义已经不再适合当前的情况。现在，焦点已经从工作组扩展到了企业范围，并且许多高级功能也已经添加到了传统的NOS产品中。NOS和其他具有连网能力（例如UNIX和Windows NT）的传统操作系统之间的区别正在迅速消失。

本章将讨论网络操作系统的使用、优点以及同它们的使用相关的安全问题。我们不是要探讨具体的实现，而是要分析一下所有NOS的一些共性。将在以下四个主流网络操作系统的基础上进行分析：

- Banyan的Vines。
- IBM的LAN Server。
- Microsoft的LAN Manager。
- Novell的Netware。

是在学习它们的工作原理之前，首先在总体上看一下网络操作系统。

8.1 网络操作系统的功能

NOS环境能够提供许多高级功能，但是它的基本用途是：

- 允许多个用户间的文件和打印服务共享。
- 允许DOS、Windows以及OS/2应用程序和数据的共享。
- 作为“前端”允许对信息的大型数据存储进行快速和有效的访问。
- 方便工作组内的信息共享。

我们将看一下NOS一般是如何进行认证、授权以及审计这些安全功能的。还要简单地看一下控制监视器和集中式安全管理方案的使用，这些产品增强了环境的安全性和完整性。最后，要看一下NOS环境所面对的主要安全问题，并探讨一下将来的开发。

8.1.1 认证

尽管网络操作系统可以使用很多种不同的方法，但认证用户一般都是使用登录ID和相关密码来进行。用户通过运行一个程序而登录到LAN上，该程序允许用户提供一个账户名（例如登录ID）和一个密码。用户ID和密码要由服务器进行校验。大多数NOS实现都把用户密码以加密的形式保存在服务器上。但很多NOS系统在客户机和服务器之间的LAN上以明文形式传递密码。但有些系统已经实现了防备密码被发现的认证机制。

例如，Novell的Netware使用一个单向认证模式来认证用户。LOGIN程序要求服务器生成一个特殊的键（称做登录键）来代表登录的用户。登录键只能使用一次，使用完后它就变成无效的。用户输入他的密码，然后该密码和登录键一起用来创建第二个键。第二个键发送到服务器上。Novell服务器重复用户刚刚进行的这一过程，并把它生成的键同用户LOGIN程序所发送的键进行比较。如果二者匹配，那么用户就通过了认证！图8-1说明了这种Netware认证方法。

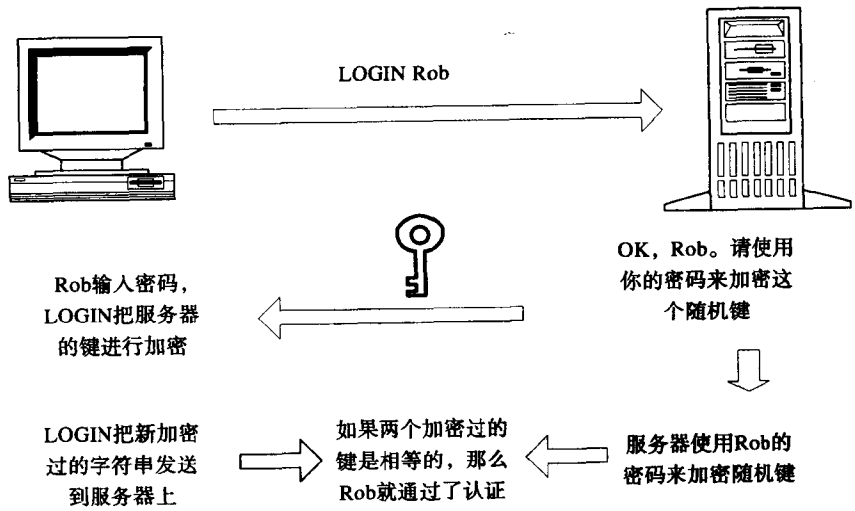


图8-1 Netware认证

这个方法的先进之处在于密码在LAN上不可能探查到。并且它也防止了重放攻击，因为第一个登录键是随机的，并且永远不会重用。并且，密码不是被保存在客户机上，也不是以

明文的形式进行传输。网络操作系统也为密码策略的实施做好了准备。一般控制包括所有的用户都要有一个密码。NOS中也可以实现密码时效性——用户需要定期改变他们的密码。有些NOS实现，如LAN Manager，允许管理员为密码设置一个最小天数，在该期间内密码必须是激活的，并且用户不能重新设置他们的密码。在大多数NOS中还有其他的一些同密码相关的控制措施，如限制密码的最小长度等。密码的重用可能也是不允许的，其实现方法是在NOS中保持一个以前使用过的密码所组成的列表。用户不能把他们的密码改变成该列表中的任何密码。该列表通常有大小限制，一般设置为保存最后5个密码。

8.1.2 授权控制

网络操作系统使用三种不同的方法来提供控制和限制，使用特权、通过文件访问权限以及通过事件访问控制。特权是指派给独立用户的特殊级别的权力。例如，OPERATOR权力允许用户管理打印服务和执行备份。尽管它们的名字对于具体实现来说各有不同，但是可用的基本管理和支持功能对于每种NOS来说都是通用的。表8-1列出了这些基本功能。

表8-1 典型的NOS特权

特 权	说 明
SUPERVISOR	能够进行用户管理、服务器的重新配置以及访问控制的管理
OPERATOR	管理打印服务和备份/恢复操作
DELEGATED PRIVILEGES	大多数NOS实现允许SUPERVISOR的权力完全或者部分委托给其他用户（通常被委托给一个独立的用户）
USER	没有任何特殊权限的普通用户
GUEST	一个普通用户ID，由那些没有自己的系统账号的个体所使用

网络操作系统通过使用许可来授权用户对文件和目录的访问。许可在用户ID或者组成员的基础上限制访问的用户和类型。表8-2列出了典型的访问许可。

表8-2 典型的NOS访问许可

许 可	说 明
READ	能够打开和读文件，列出目录的内容
WRITE	能够改变文件或者目录的内容
CREATE	能够创建新文件或者新目录
DELETE	能够删除文件或者目录
EXECUTE	文件可以作为程序来运行
CHANGE	用户可以改变对象的许可
ARCHIVE	用户能够创建对表的特殊引用，这称为外键

网络操作系统使用了基于许多事件的附加访问控制。这些控制包括：

- 在工作时间之后禁止访问的时间控制。
- 一段时间（几个小时）后的用户自动注销。
- 如果连续一段时间不活跃，那么注销用户。
- 如果用户数次（通常三次）尝试登录失败，那么禁用该用户账户。
- 限制并发会话（通常为一次）。

有些厂商提供了相对惟一的访问控制。例如，Novell Netware能够要求登录必须来自于一

个预定义的LAN地址。Banyan Vines可以强制登录来自一个特定的操作系统，如OS/2或者DOS/Windows。

Novell也鼓励第三方使用Netware可加载模块（Netware Loadable Modules，NLM）来为Netware环境创建安全方案。NLM使得特殊权限可以扩展到普通用户上，并且能够用于包括安全在内的很多目的。例如，LAN支持组（LAN Support Group）的NetSqueeze+Encryption NLM允许用户自动地对敏感文件进行加密/解密。NLM可以通过密码来保护。

8.1.3 审计跟踪

同NOS环境中的大多数事情一样，审计跟踪也有一个类似的目的但每个厂商的实现各有不同。记录在审计跟踪中的基本活动的类型对于大多数厂商来说都是共同的，它们包括：

- 对用户的登录/注销活动的记录。
- 对有关资源的许可的改变。
- 对用户特权和状态的改变。
- 访问资源请求（包括登录请求）的成功或者失败。

大多数审计跟踪都不是默认启用的。包括LAN Manager在内的一些NOS环境允许审计跟踪限制所创建的审计数据量。访问选择的资源或者选择的用户可能被特殊记录下来。并且，失败的访问尝试（通常是登录）可以进行审计以检查猜测尝试密码。

审计跟踪的当前实现中存在很多问题。审计跟踪保存在本地系统上，因此一个具有足够特权的用户能够破坏它们。一个获得了SUPERVISOR特权的用户能够删除他们的审计记录，甚至破坏整个审计跟踪。并且，审计跟踪的格式和结构对于每个厂商来说是不同的。这使得统一集中式的报告变得很困难。如果可用的话，会将警告发送到系统控制台上，但网络管理员可能不会定期的访问控制台，所以他们可能会过一段时间才看到警告——如果该警告还没有被后续的消息所覆盖。

8.1.4 NOS安全方案

很多厂商提供了可用的第三方方案以提高NOS环境的整体安全性。这些方案提供了控制监视和集中式安全管理功能。控制监视器是一种特殊的安全工具，它能够检查NOS安全结构中的脆弱点。一般的检查包括脆弱的或者不存在的密码、不活跃的账户以及过多的用户访问权限。

例如，Frye Utilities是一个Netware的工具，它能够在登录时进行很多检查，其中包括不存在的密码和不符合标准的密码等。如果一个账户在预定义的天数之内没有被访问，或者当用户赋予SUPERVISORY特权的时候，该工具就会进行报告。

一些厂商也开发了很多提供增强的访问控制和审计功能的方案。例如，Centel Federal System公司的Net/Assure产品提供了系统访问控制、加密、数据访问以及系统集成服务以补充固有的Novell安全性控制。

8.2 有关NOS实现的问题

下面看一下有关使用网络操作系统的一些问题。根据可能发生的攻击类型，这些问题大致可以分为几类。首先，讨论一下对一个NOS服务器进行限制访问的问题。

8.2.1 物理访问

大多数计算机，包括NOS服务器在内，都可以使用对其系统控制台的物理访问来破坏。即使使用了密码来保护系统控制台，系统也可以使用一个特殊格式化的软盘来重启。一旦系统重启了，那么使用一个普通的诊断工具（如Norton Utilities）就可以很容易地泄漏服务器的安全信息。因此，使用上锁的工具来保护服务器是非常重要的。通过实现DOS或者OS/2访问控制方案（在软件中或者通过硬件），系统可以获得附加的安全性。有些管理员把服务器上的螺丝卸掉以防止服务器被破坏。最终这会使得硬件维护变得非常有趣。

8.2.2 特洛伊木马

在PC客户机上进行攻击是NOS环境中最常见的攻击形式之一。通过插入一个伪造的登录程序或者使用一个终结并常驻（Terminate and Stay Resident，TSR）程序来捕获击键，就可以捕获密码并在以后重用。如果PC客户机上没有实现健壮的访问控制，那么这些方法是很难发现并阻止的。现在有很多个人计算机访问控制解决方案，如PC Dynamics公司的Menu Works产品和Mergent International公司的PC-DACS。这两种产品都提供了用户ID和密码控制功能，并且能够限制对文件和目录的访问。

8.2.3 LOGIN脚本

有些命令或程序能够自动地把用户连接到网络上，它们也带来了潜在的脆弱性。通过访问这些脚本，黑客能够捕获密码，甚至有时候能够完全绕过安全机制。一本流行的计算机杂志曾经介绍过这么一个程序，通过该程序，黑客能够绕过Novell LOGIN程序。该程序的实现方法是糊弄服务器以使其认为当前用户已经通过了认证。

8.2.4 密码攻击

通过猜测密码来进行攻击可以得到对NOS服务器的未经授权的访问。有一种程序能够尝试使用一个字典中的每个单词作为账号密码进行登录，黑客可以使用这种程序来进行攻击。这种攻击类型的一个变种是尝试获得对没有密码的账号的访问权。防止这类攻击的最普通的方法是在一个审计跟踪中捕获失败的登录并向系统管理员发送警报。

8.2.5 管理的一致性

NOS的管理最容易产生问题。尽管有些网络管理员知识渊博并且忠于职守，但是许多组织内的NOS管理仍然缺乏一致性。很多LAN管理员都未经培训，或者经验尚浅，这使得系统的安全控制变得很脆弱。控制的不一致应用常常导致抱怨，例如默认Supervisor密码的使用，或者在用户离开组织以后其账号仍然可用。

一些最新的“企业级”NOS发布版本为此问题提供了解决方案。例如Netware 4.1有许多用于企业计算的高级功能，其中包括Netware目录服务（Netware Directory Service，NDS）、时间同步服务（Time Synchronization Service，TSS）以及中心管理工具（NWADMIN）。Banyan Vines早已提供了服务复制以及集中式目录服务。

可用目录服务来实行命名和目录标准，而TSS则给出了时间的一个公共定义。集中式管理

功能使得标准和策略具有更大的一致性，这是因为只有少数管理员需要负责他们的实现。策略和标准可以实施在整个域内的网络服务器上，而不是每台服务器都有自己的一套。

8.2.6 GUEST账号

GUEST账号通常用来为临时员工或者临时用户授予NOS资源的访问权。GUEST账号允许那些在服务器上没有一个独立账号的用户访问服务器上有限数量的资源。它允许用户对服务器进行临时的访问而不必惊动服务器管理员。实现GUEST账号有很多不同的方法。例如Netware支持GUEST账号的使用。LAN Manager使用一个PUBLIC共享来实现，允许任何人可以对一个指定的目录和子目录进行访问。

许多问题都同GUEST账号的使用有关。GUEST账号可以说是非法或者不正确内容（例如盗版软件或者色情内容）的包容所。许多GUEST账号没有与其相关的密码，这使得它们很容易滥用。

大多数用户并不知道，把一个对象放在GUEST账号中会使得该对象能够被所有的网络用户所访问，而不仅仅是他们所预期的接收用户。许多组织允许合约人员和顾问访问组织的内部LAN。放在一个PUBLIC目录下的信息经常会不经意地被非雇员看见或利用，这是一个经常听到的抱怨。

8.2.7 病毒防护

服务器上的病毒防护工作应该比个人计算机上的更加重要。不幸的是，我们发现很多组织为个人计算机上的软件投以巨资，但却没有考虑到服务器。服务器上的病毒应该严肃对待，这是因为访问服务器的用户要比访问一台个人计算机的用户多得多。这使得服务器染上病毒的可能性也比个人计算机大得多，原因就是服务器要被大量的人访问。服务器是病毒扩散的中心点，通过服务器，病毒能够感染许多用户。服务器感染病毒所带来的工作损失和中断要比一台工作站大得多。连接服务器的所有用户都应该检查自己工作站上的病毒情况。

对于个人计算机的所有用户来说，病毒防护软件的使用是一个必要的预防措施。对于服务器来说，这同样正确。在服务器上使用病毒防护软件有一些需要考虑的地方。并不是所有的病毒防护软件生厂商都为他们的产品提供服务器版本。病毒防护软件应该能够检查到潜伏病毒的存在。例如，压缩文件也应该进行病毒检查。病毒软件应该能够向网络管理员发送警报。当从软驱上启动服务器的时候，病毒防护软件应该激活。

所有的病毒都要执行用于传染的代码。有个简单办法能够降低病毒在服务器上发作的可能性，禁止用户在所有那些包含可执行程序的目录上拥有写特权。当然，通过一个登录脚本来在客户机上运行病毒防护软件也是个好方法。另外，对其他用户目录下的文件的执行许可也应该受到限制。最后，不要忘记定期更新病毒防护软件！

8.2.8 工作组计算

使用个人计算机的一大优势是能够提高开发文档和共享工作处理的效率。然而，如果你是团队的一部分，并且必须要同团队协调进度，那么这种效率会消失。“谁有那份报告的最新拷贝？”、“哪个是当前的电子表格？”——如果组织里使用了大量的软盘，那么前面的问题可能会不绝于耳。已经提出了针对该问题的软件方案，它能够让这种混乱走向有序，并使得

一组个体能够一起工作。这类软件称做群件。例如，有些人认为电子邮件是群件的一种实现。

群件指的是允许一组人——通常具有一个共同的业务关系，并且可能分布在多个地点——能够使用共享资源像一个团队一样工作的一类计算机软件。这些解决方案能够囊括很多业务行为，其中包括电子邮件、时间管理和计划、数据管理以及文档开发和管理。群件服务的集成度越高、所能涵盖的范围越广，群件的工作效率就越高。

群件实现也提供了相当健壮的安全控制，这包括公开密钥加密的使用。在后面，将更加详细地探讨有关群件的知识，并对Lotus Notes进行特别研究。

8.2.9 将来的开发

许多NOS厂商使用了，或者有计划结合了TCP/IP网络协议。尽管使用TCP/IP协议能够带来同外部世界更方便的通信能力，但是它也会增加NOS控制的暴露程度和测试工作，并且提高NOS的不安全性。

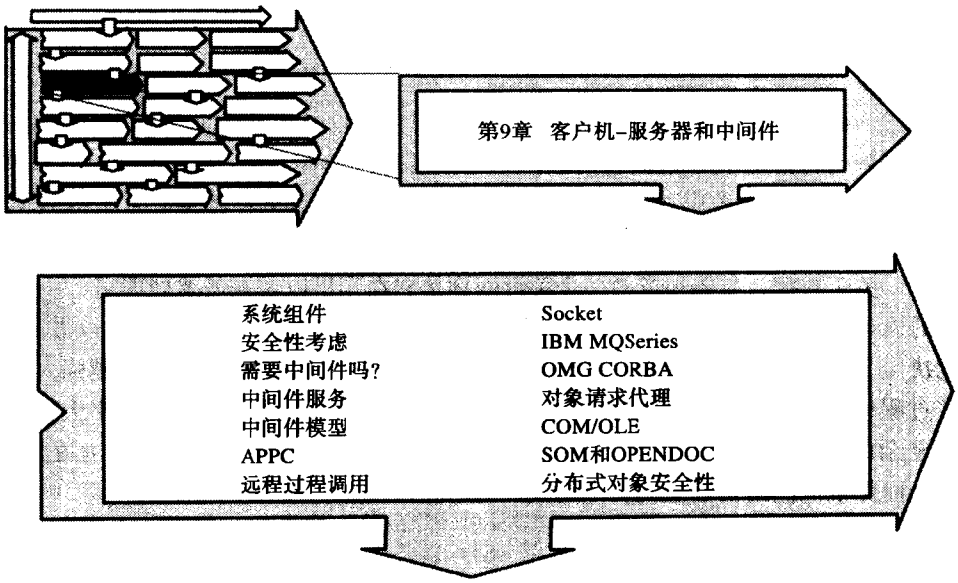
有些厂商正在把开放软件基金会（Open Software Foundation）的分布式计算环境（Distributed Computing Environment，DCE）结合进他们的产品中，这是一项积极的开发工作。例如，Novell已宣布支持OSF/DCE，这使得包括OS/2和UNIX在内的其他操作环境能够同Netware环境共享认证服务。IBM也宣布为OS/2 LAN Server支持DCE。

8.3 结论

网络操作系统和开放式系统操作系统将来应该趋向统一。随着这种统一，新的问题会产生，但是这些问题应该能够通过更健全的管理、访问控制以及审计功能解决。认证和访问控制解决方案已经开发出来了，它允许对两个环境进行共同的安全管理。可以预见的是，这种工作还会继续，在未来几年内还会出现更多可用的集成解决方案。

在下一章中，我们要继续探讨网络。要进入客户机—服务器的世界，并探讨一下把应用程序和服务结合在一起的东西——中间件。

第9章 客户机-服务器和中间件



在第8章中，探讨了网络操作系统以及它们为分布式环境带来的信任元素。在本章中，要分析另一种分布式形式——客户机-服务器计算。客户机-服务器系统并不是一个神秘的概念，但是它的实现可能非常神秘。我们将尝试概括一下客户机-服务器的组件，并对一些有关信任的考虑事项进行分析。有很多不同的方法和技术都可用于提供客户机-服务器计算。尽管本章不能涵盖所有方法，但掌握其概念并知道哪里有优势哪里是脆弱性所在很重要。

为了能够应用适当的安全机制，高级客户机-服务器系统涉及到了很多概念和构建部件。掌握这些概念和部件是非常重要的。在第1章中，我们讨论了几种不同的分布式处理模型。本章将着眼于客户机-服务器及其协作处理模型。记住，这些方法包括了在客户机和服务器上某种形式的应用程序的逻辑执行。我们将引入逻辑工作单元（Logical Unit of Work，LUW）作为一种描述如何定义和划分需要进行的工作的方式。我们还要介绍一下用于保持数据库更新完整性的两阶段提交过程。当数据库能够被多个用户更新的时候，一个两阶段提交是非常重要的。中间件是提供服务以使得分布和管理系统工作能够更加容易的软件。我们将介绍一些中间件方法和技术。最后，探讨一些有助于分布工作的可用技术。

现在，客户机-服务器计算被认为是把系统结合在一起的标准方法。随着个人计算机的使用、LAN的实现以及网络操作系统的完善，这种方法开始得到大量关注。客户机-服务器计算的增长似乎是由计算领域的两个目标演变而来。一方面的推动力是PC日益增长的处理能力和以一种受控方式来共享PC生成数据的需要；另一方面的推动力来自于对利用PC相对廉价的处理能力而同时限制昂贵的集中式处理的成本的需要。在这两种情况下，我们都需要管理数据并利用方便的计算能力。

当一些使用大型机计算的组织开始使用PC的时候，下面这一点迅速成为很明显的事实，一个计算系统的大多数输出成为另一个系统的输入。大量的数据重新输入到一些电子表格程序中，这些程序降低了由一个基于大型机的系统所生成的文档报告的数量。通过把文件从大型机上拷贝到PC上，这种麻烦得到了改进。然而，虽然这种方式避免了数据的重新输入，但是它仍旧需要PC上有一些有用的操作。更好的做法是让PC程序同大型机系统直接对接，并能够在需要的时候得到当前数据。然而这种集成常常需要进行编程以得到并转换数据。

系统的组件

当我们仔细研究一个计算机系统的时候，会发现它可以分成几个简单的组件。第一个重要部分是数据或者信息。关于数据和信息的定义，人们一直有争论。根据这本书的目的，我们把数据看做在计算机系统中操作和存储的具体内容，而把信息看做是有意义的数据表示。一般来说，当使用一个计算机系统的时候，不是查看信息，就是把数据变换成你能查看的新形式的信息。关于数据，需要知道的一个关键问题是它在哪里以及谁需要处理它。

可用的以计算机可读格式存在的数据越多，生成的信息就越多，这样就可以做出更好且更及时的决定。用户方便访问的处理能力越强大，访问和分析数据的能力也就越强大。然而，如果这些过程包括多个位置数据的修改，这种访问也会为维护数据的完整性带来大量的困难。我们需要复杂的软件来帮助管理访问并维护数据的完整性。也需要能够跟踪谁正在访问和改变数据，从而保证他们确实是他们所说的人并且已经授权那么做了。

9.1 客户机-服务器

客户机-服务器的定义多种多样，这要取决于所涉及到的具体问题或者具体软件厂商所提供的产品。它描述为跨越多个平台的分布式智能的一个紧密的纲领性的结合。当你使用PC机同网络上的应用进行通信的时候，就会牵扯到客户机-服务器这个概念。客户机-服务器计算这个术语通常用来描述以下情况，PC机支持图形用户界面，该界面用来访问一个连网服务器上的应用程序。根据本书的目的，我们把客户机-服务器的定义尽可能地保持简单。

> 客户机是向服务器请求服务的进程。服务器是服务于客户机请求的进程。客户机-服务器描述了把计算系统分割成客户机部分和服务器部分的方法。

1. 逻辑工作单元

分布式计算系统中的一个重要概念是逻辑工作单元（Logical Unit of Work）或者LUW。逻辑工作单元是必须协调和执行以完成的进程集合。如果在协调进程的任何部分中出现了一个故障，那么所有的进程都必须要回滚到该逻辑工作单元初始化之前。LUW必须是整个全部完成，或者如果进程不成功的话，那么LUW必须全部回滚就好像根本没有进行似的。一个LUW的完成通常标记做一个同步点或者一次提交，来表明所有的数据都处于一个一致的状态。

图9-1描述了一个包含两个独立逻辑工作单元的进程。读文件A和B然后更新文件A，这一动作看做是一个逻辑工作单元，而读取并更新文件C看做另一个逻辑工作单元。每个逻辑工作单元代表一个任务的一次完成和一致状态数据的同步化。我们能够发现不知道的已经定义和执行的LUW。所有的LUW可以不总是包括对提交或者同步点的需求。即使是一个因特网浏览

器（如Netscape）也包含可以看成是逻辑工作单元的程序逻辑。

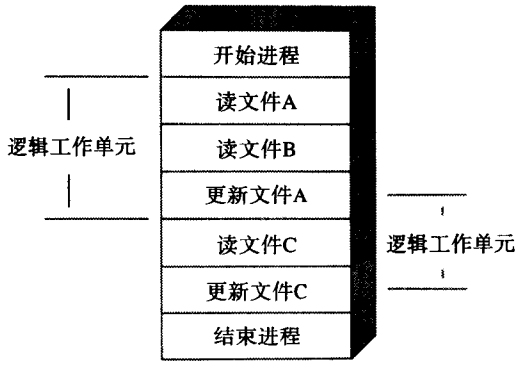


图9-1 逻辑工作单元示例

2. 两阶段提交

在分布式系统中，两阶段提交协议是非常重要的数据库更新（无论什么时候）。当把工作分布到多个进程上的时候，一个需要解决的重要问题是保证所有的工作都成功地完成，或者所涉及到的数据都回滚到一个时刻就好像工作根本没有进行似的。管理这种需求的方法就是两阶段提交。如果逻辑工作单元分布在两个独立的进程上，那么两阶段提交就更加重要。

图9-2说明了两阶段提交过程。当进程A请求数据库更新的时候，会发送通知到进程B（数据的管理者），接着B准备更新。如果进程A收到进程B的许可确认，那么A就会发送提交请求。一旦进程B使用提交的消息来确认，那么逻辑工作单元就完成了。如果进程A或者进程B不能完成它的逻辑工作单元，那么更新过程就放弃了。当一个逻辑工作单元的成功分到几个进程上时，两阶段提交协议是非常重要的。

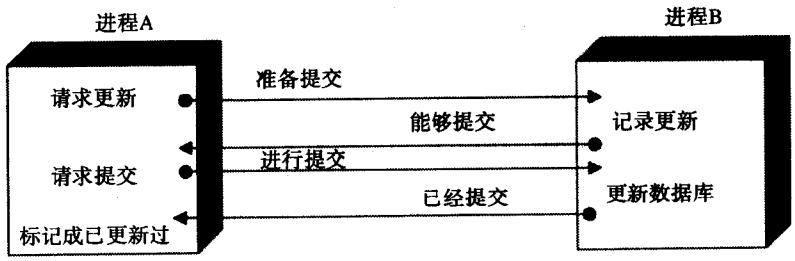


图9-2 两阶段提交

3. 协作处理

在客户机-服务器计算流行之前，术语“协作处理”用来描述一个应用程序功能分散到两个或多个进程之上但执行起来仍然好像是一个应用程序的能力。在客户机-服务器计算中，协作处理用来描述一种能力，即把一个逻辑工作单元的处理分布到两个或多个进程上，然后以一个及时的和协作的方式完成该工作单元。这些进程通常在几个处理器上执行，但这并不是必需的。协作处理的一个早期实现来自于IBM对其应用程序间通信（Application Program-to-Program Communicatioin，APPC）工具的实现。

协作处理的定义很窄，它取决于一个逻辑工作单元到多个独立进程的分布，而不是一个更简单的客户机-服务器关系。要成为一个协作进程，必须满足如下条件：

- 执行一个逻辑工作单元必须要牵扯到多个应用功能（程序）。
- 协作进程必须具有可编程的信息。
- 在逻辑工作单元的执行中，参与处理器必须使用该可编程的信息。

协作组件之间的处理可以是异步的也可以是同步的。在逻辑工作单元完成之前，协作程序之间的控制方向可以是主从式的也可以是对等的。在LUW的一个提交点，工作必须永久地应用于数据，或者所有涉及到的系统都回滚。协作处理是一般客户机-服务器计算的一种特殊形式，它需要应用程序执行的紧密集成。

客户机-服务器的安全性考虑

在客户机-服务器关系中，最重要的元素之一是客户机到服务器和服务器到客户机的相互认证。客户机必须向服务器提供标识证明，服务器必须证明客户机提供的标识是正确的。客户机和服务器之间的通信需要加密以保证机密性，这种能力是任何标识和认证详情都需要的，并且其他机密数据可能也需要。客户机-服务器关系可以使用第三方的安全性，如由Kerberos提供的。这种机制能够提供客户机和服务间的相互认证和网络机密性。Kerberos的详细介绍将在第17章中提供。在客户机-服务器关系的双方，机密数据（如加密密钥）的安全存储也是需要的。在客户机-服务器关系中实现健壮的审计功能也是一项重要的需求。

9.2 中间件

对于有些电脑术语，你可能整天听到但却不能准确地说出它的意思。中间件就是其中之一。在一个单词后面加上“ware”似乎就能说明一种新类型的产品，这是一个少有人知的语言规则。当要执行使用两个或多个计算机的程序——但你却不想了解这些机器的时候，中间件就可以大显神威。有时候，中间件指的是能够把分布在不同地点的应用程序和数据结合在一起的神奇粘合剂。实际上，中间件是一些软件服务的集合，这些软件服务能够提供了一个容易的方式来实现客户机-服务器应用程序，从而使得用户不用亲自对所需的全部分布式服务进行编码。当使用了多种不同的平台时，中间件能够提供一个公共的接口和服务范围。高级中间件能够提供服务以防止应用程序知道其他应用程序或数据具体位于何处。后面将详细讨论这些内容。

> 中间件是一个软件层，它提供了应用程序、数据和操作系统之间的一个公共接口和转化。

9.2.1 需要中间件吗

对这个问题的简单答案是如果没有特殊要求的话，不需要。然而，如果不使用某种形式的中间件，你可能需要为智能进程间的协调、通信以及恢复所需要的全部服务亲自编写代码。在出现故障的时候进行恢复是健壮的中间件的最大价值所在。在分布式系统中，可能出现故障的地方有很多。合适的中间件有助于在系统的某个部分出现故障的情况下管理恢复，同时不给应用程序添加额外的负担。关于故障发生时应用程序对恢复过程的知道程度，以及在恢

复过程中所涉及到的程度，人们在需求方面有着争论。关于，中间件的变化范围很大，从保护应用程序完全不知道它的分布情况到提供能够用于管理分布的特殊应用接口和过程。应用程序可能需要知道何时故障出现。为此，中间件应该提供应用程序可以调用的程序接口。有时候，应用程序需要知道它是如何分布的。这种需求主要是对使用动态决策（使用针对情况的程序逻辑）来控制程序执行的一个功能。

9.2.2 中间件服务

中间件提供了几种服务以使得能够成功地把处理分散到分布式环境上。下面的列表给出了一些服务。该列表并不详尽，但是它说明了中间件一般会提供的服务类型。有些中间件实现包含了更丰富的服务，而有些则只提供基本的服务。

- **目录**——跟踪网络资源（如文件、服务器和应用程序）的位置和特性。目录是基于独立于物理位置的名字之上的，这样使得物理位置可以改变。
- **安全**——为注册的系统和用户提供授权、认证和审计服务，从而保护网络资源免遭未经授权的使用。
- **管理**——包括分布式环境上的问题、操作、配置、改变和性能管理服务。
- **应用编程接口**（Application Programming Interface，API）——提供一个一致的接口，通过该接口，应用程序能够调用中间件的服务。API可以在许多不同的平台上可用，从而提供异构分布式环境上的一致性。
- 时钟——用于网络资源的时间校对，从而可以调度服务和协调网络分布式行为。

9.2.3 中间件模型

有许多中间件方法和技术可用于客户机同服务器的交互。中间件机制一般符合图9-3中概括的几种模型。符合对话模型的中间件能够为客户机和服务器间的简单通信提供方便性。客户机要求服务器进行某种操作，然后等待答复。这种交互类型同电话交谈类似，它可以称做请求/响应。在一些网络协议定义中，这也可以称做是面向连接的，这意味着为了进行交谈客户机和服务器之间需要一个网络连接。在这种情况下，应用程序可以检查到任何故障，并估计恢复所带来的负担。

远程过程调用模型提供了一个能够定向到网络中任意服务器上的程序回调机制。图9-3说明了RPC同对话模型的类似之处。这种回调模型同用来执行应用程序的子例程的模型类似。同样，应用程序能够知道RPC请求的任何故障，并且需要采取动作。这种模型是同步的，依赖于客户机和服务器的协调执行。

消息模型使用消息队列来同客户机和服务器之间的请求任务进行通信。客户机和服务器会查看这些消息队列中是否有消息。如果有的话，它们还要根据那些消息进行动作。消息模型使得实现消息驱动的处理成为可能，从而替代了过去经常使用的事务驱动的处理。消息驱动模型对于实现工作流处理或者任何不需要同其他进程有持久关系的协调处理来说是非常有用的。该模型是异步的，客户机和服务器的执行互相并不依赖。在有些网络协议中，这可以称做无连接模型，其意思是客户机和服务器之间不需要一个具体的连接。

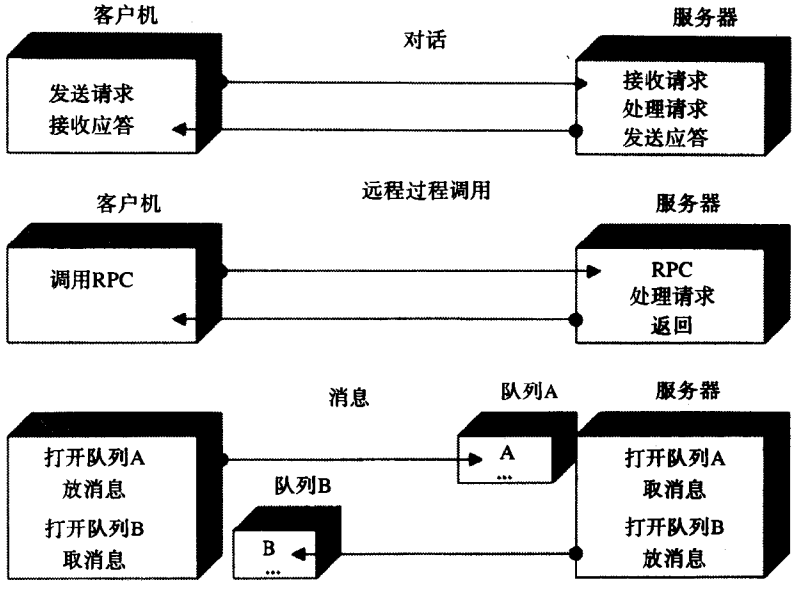


图9-3 中间件模型

9.3 可用技术

中间件提供了进程间通信和执行能力。实现中间件有很多方法和方案。这些技术一般是针对前面讨论过的几种中间件模型之一的。下面将对几个不同的技术进行概括性说明，但是我们并不准备列出所有可用的技术和实现。这是一个快速浏览小节，因此我们强烈建议你使用每种技术进行一下具体的开发。

9.3.1 应用程序通信

分布式处理环境的第一个里程碑是IBM的应用程序间通信（Application Program-to-Program Communication，APPC）。APPC接口允许两个应用程序间的直接同步通信。APPC定义了IBM的系统网络体系结构（System Network Architecture，SNA）下所需的进程间通信协议。APPC的实现是使用6.2类的SNA对等逻辑单元（LU）完成的。LU 6.2是一个事实上的标准，它允许不同处理节点上的应用程序能够建立一个通信会话，并在应用程序之间发送和接收对话消息。使用APPC，可以提供一个完整的两阶段提交。几个软件厂商已经开发了LU 6.2功能，并把它们包括在应用程序和服务中以提供这种分布式的互操作能力。客户机-服务器计算的一些最早的实现使用了APPC。

9.3.2 远程过程调用

使用远程过程调用（Remote Procedure Call，RPC）是另一种在分布式环境中提供进程间通信的方法。通过使用请求服务器进行某种操作的RPC，客户机能够同一个服务器进行交互。客户机会被挂起直至从服务器收到一个响应为止。RPC机制常常用来在一个分布式系统上提供安全服务。

Sun公司提供的安全RPC机制提供了一套系统调用，这些系统调用使用数据加密标准（Data Encryption Standard，DES）加密算法在LAN上传递机密数据。安全RPC使用加密的网络信息服务（Network Information Services，NIS）服务器来存储用户的密钥。安全RPC的一个主要优势在于它能够很容易地同Sun的低级RPC进行集成。这使得现有的基于公共ONC RPC的应用程序能够转化到一个更安全的环境中。

开放软件基金会（Open Software Foundation，OSF）的分布式计算环境（Distributed Computing Environment，DCE）中包括的RPC机制也提供了一套用于进程间通信的调用。DCE认证是基于Kerberos模型的，这是一个来自MIT的可信第三方证明许可模型。Kerberos提供了客户机和服务器的独立校验功能，并且能提供一个更为健壮的控制环境，其原因在于它同分布式文件系统（DFS）的紧密结合。认证完成以后，就可以提供对应用程序访问的授权。并且，分布式文件系统以及它的访问控制是同RFC机制紧紧对齐的。我们将在第16和17章中仔细探讨DCE和RFC机制的实现。

9.3.3 socket

伯克利进程间通信（Inter-Process Communication，IPC）机制（也称作socket）提供了一套用于进程间通信的系统调用。客户机进程建立一个socket，并请求到服务器的连接。服务器在一个预定义的服务号（称做端口）上监听请求。一旦服务器进程接受了客户机请求，那么客户机和服务器上的两个socket之间就可以进行全双工通信。socket机制在大多数UNIX实现中都可用。socket为TCP/IP网络上的网络应用程序提供了当前的事实标准。该机制最适合于对话模型。socket无法提供全面的认证和授权控制。

9.3.4 IBM MQSeries

IBM提供了支持消息中间件模型的技术。IBM MQSeries产品提供了IBM平台和其他平台上的基于队列的接口和管理服务。使用MQSeries技术的应用程序能够同使用队列的分布式组件进行交互。该技术是由一个消息队列接口（Message Queue Interface，MQI）和一个消息队列管理器（message queue manager，MQM）功能组成的。MQI为应用程序提供了接口，并把网络连接和协议向开发人员屏蔽起来。MQM提供了名字和地址解析、消息路由、MQM资源管理以及同步点参与。消息为异步处理提供了一种独特风格的通信和能力。队列机制支持可保证的消息传输、恢复能力以及同步点服务。使用MQSeries的商业实现已经开发出来以支持工作流计算。

9.4 分布式对象

处理的分布化中大量涉及到面向对象系统。当前，分布式对象处理系统是非常复杂的。然而，支持软件正在迅速发展。分布式对象系统中的关键元素之一是实现一个机制以管理对象的分布化和目的地。对象之间必须要维护一个安全关系，即使分布式对象的导航路径不可预测的时候也同样如此。

9.4.1 OMG CORBA

对象管理组织（Object Management Group，OMG）是一个由100多个厂商组成的联盟，

它做为一个标准团体而存在，其目的是解决面向对象技术中的互操作性和可移植性问题。OMG正在建立一个用于创建分布式对象网络的标准，并定义它们之间互相通信的方式。允许对象能够互相通信的机制是通过对象请求代理（Object Requet Broker，ORB）。ORB用来实例化对象、提供对象间的通信以及代表对象调用方法。OMG开发了一套称做公共ORB体系结构（Common ORB Architecture，CORBA）的规范。该规范定义了ORB、对象服务、公共工具以及应用程序对象，并且提供了使得客户机-服务器对象能够交互的接口定义语言（Interface Definition Language，IDL）和API。CORBA的2.0版指定了不同厂商的ORB应该如何进行互操作。

9.4.2 对象请求代理

分布式环境中的对象共享是通过使用对象请求代理来完成的。ORB提供了连接服务以同服务器对象通信，并激活或者保存服务器对象。应用程序对象组件是具体于终端用户应用程序的。这可能包括电子表格或者字处理类型的应用程序。公共工具组件定义了管理对象的方式。如任务管理、系统管理以及用户界面之类的组件也在此部分中定义。对象服务扩展了ORB的能力。这些扩展部分包括命名、事务和并发控制之类的功能。时间和安全服务也包括在此部分。ORB提供了建立对象和客户机-服务器之间关系的中间件。图9-4给出了对象管理组织的公共ORB体系结构使用的ORB体系结构示例。

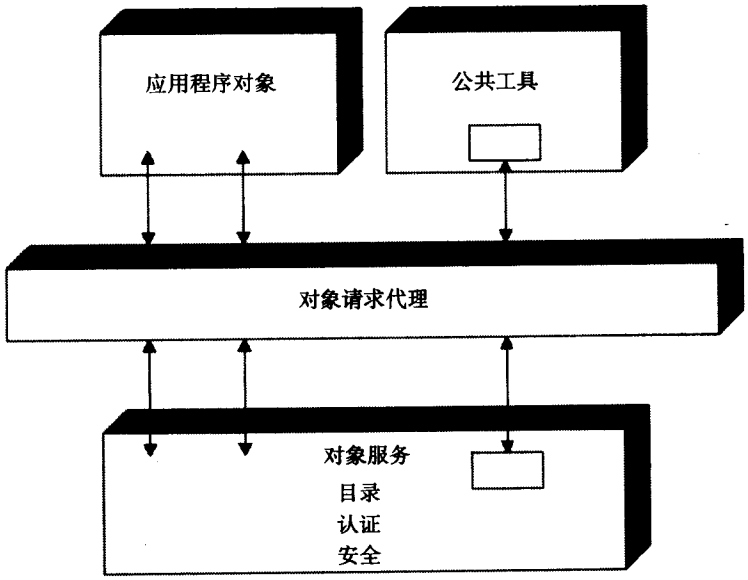


图9-4 公共ORB体系结构

9.4.3 COM/OLE

公共对象模型（Common Object Model，COM）是由Digital和Microsoft为支持对象链接与嵌入（Object Lingking and Embedding，OLE）环境而开发的对象请求代理。对象可以使用多种语言实现，传统语言和面向对象语言都可以。COM模型同CORBA类似，它也为对象接口使用接口定义语言并把对象接口同其实现分离开。COM分布式模型是基于DCE RPC的。在

一个基于Windows的系统上，对象链接与嵌入技术集成了多种应用程序和数据类型。OLE支持超过350个的基于COM的应用编程接口调用，并能够使用专门为网络实现提供的OLE工具在网络上进行通信。分布式对象的COM/OLE实现是OMG CORBA模型的竞争对手，但它是DEC和Microsoft所专有的。

9.4.4 SOM和OpenDoc

在分布式对象领域，IBM的涉入是建立在它的系统对象模型（System Object Model，SOM）和OpenDoc组件软件体系结构的基础上的。OpenDoc提出了一个复合文档规范，它是由CI Labs开发、并在IBM、Apple、Novell、Oracle、Taligent和其他公司的指导之下形成的。复合文档可以包含由许多元素组成的组件，这些元素包括文本、图形、图像、声音、动画甚至电影。OpenDoc定义了所有这些组件应该如何配合和交互以形成一个完整文档。SOM是CORBA兼容对象服务的IBM实现，它提供了OpenDoc组件的本地和远程互操作能力。SOM是CORBA对象请求代理的一种实现。SOM的对应物是COM，而OpenDoc的对应物是OLE。

9.4.5 分布式对象的安全性考虑

分布式对象技术的使用为安全问题提出了另外的挑战。分布式对象具有在通常客户机-服务器环境中所能发现的所有安全问题，并且还有额外的独特问题。我们需要处理面向对象技术和传统方法之间的区别，这是主要的挑战所在。对象间的交互和关系同传统应用程序和数据关系间的交互根本不同。对象自己不拥有一个固有的用户概念。由于一个分布式安全系统的关键支柱是一个认证的标识的使用，所以对象的这种特性可能会导致问题。需要附加元素来满足分布式对象中的这种需求，或者需要一个更高级的信任（这也伴随着更高级的风险）。

对象间的关系可能并不总是明确定义的或者非常容易理解的。封装向程序员或者用户隐藏了对象实现的细节。对象既可以作为客户机也可以作为服务器，这取决于对象的具体调用。信任放在对象中的位置或者为谁而放并不总是明确的或者可预测的。在分布式对象技术中，引入新的对象或者改变过的对象组件可以不需要向其他组件通知，这固然是一个优点，但是它也会引入更多的风险。如果信任机制不能结转并且作为通过对象的导航组件执行，那么信任必须放弃所有对象驻留其中的执行环境中。有可能一满车的人都到达一系列的安全检查点。可以考虑对整个汽车进行授权以允许其前进，或者在汽车继续前行之前应该分别检查车上的每个人。

9.5 密切注意发展趋势

不幸的是，现在客户机-服务器领域正处在变动之中。好消息是它仍旧在发展，并且我们能够看到更丰富和更容易的实现。同时，当你决定如何接近客户机-服务器和中间件的时候，应该进行一些考虑。

在面向对象系统的领域中，中间件的前途将继续光明。面向对象系统已经对信息技术产生了重大的影响，并且还会继续发展。然而，标准之争还没有完成，就像典型的正确（right）和可能（might）之争。有着大量的厂商支持，OMG的CORBA规范似乎会流行起来。许多人把这看成是right（正确）标准——由许多不同的厂商通过一个独立组织（OMG）来支持。COM方案似乎是might（可能）标准，Microsoft通过它的操作系统和其他专利产品来促进它的

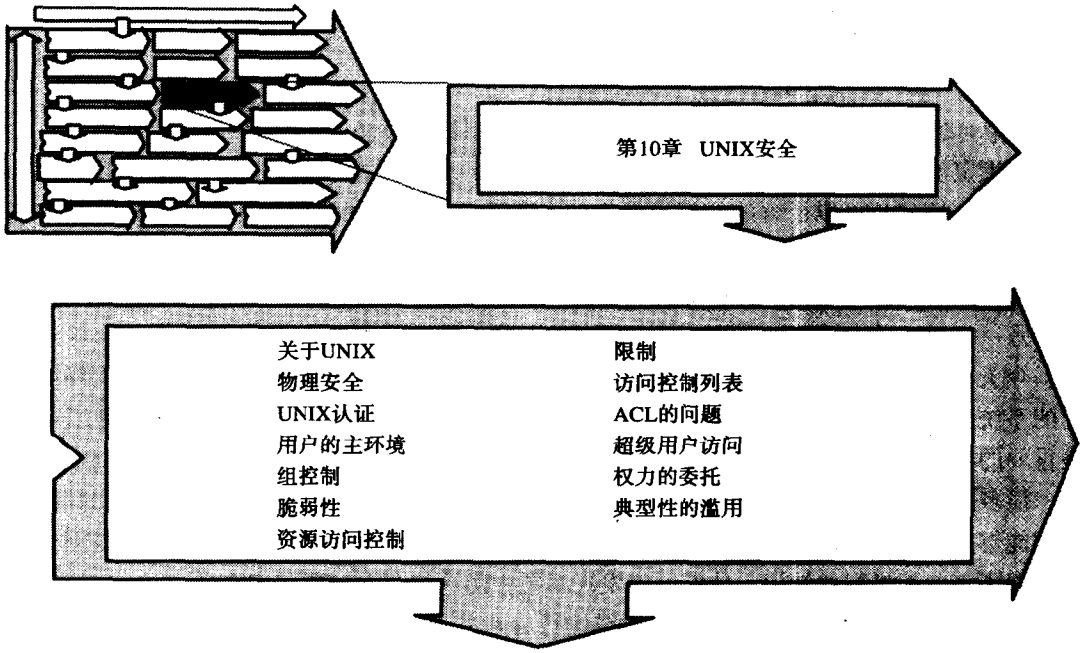
发展。其他竞争者也正在进入这个领域。惟一可预测的就是分布式对象肯定是未来系统开发的一部分。

安全RPC机制在UNIX市场中的许多系统上都可用。对于一个应用程序来说，安全RPC能提供足够的认证和授权控制。但是，安全RPC并不适合于把多个应用程序分布到分布式环境中。安全RPC本身并不对多个应用程序进行管理和控制，它的实施是建立在用户密码控制基础上的。相对于安全RPC，OSF/RPC的主要优势在于它通过一个安全服务器和分布式文件系统提供了集成的授权和认证控制。

9.6 小结

在本章里，我们探讨了客户机-服务器系统的一些考虑事项和实现途径。具体的安全问题和详细信息将在下一章中介绍。当比较几种类型的中间件模型和基于这些模型的技术时，必须要考虑到安全问题。对标识、授权、认证以及机密性的需求不会随着技术和方法的改变而改变。客户机-服务器系统的任何设计都应该支持一个已定义的安全体系结构以保证所需的全部安全机制都到位。这是一句老话，但是它是真的——安全只取决于最脆弱的链接。如果你有最坚固的锁和最好的门，但是房子的后窗却开着，那么一切都是白费。

第10章 UNIX 安 全



近几年来，采用开放式系统的技术已经成为一股潮流。今天，大多数组织对开放式系统操作系统环境的选择都是UNIX，它已经成为工程工作站、数据库以及中等规模计算的事实标准。在几年前还很少听到过开放式系统的许多组织，现在已经成功地部署了基于这种技术的方案。由于有了公共操作环境，软件厂商能够容易地把它们的产品移植到开放式系统上。这使得客户有了更好的选择和更具竞争力的价格。然而，在实现开放式系统之前，安全和信任的问题必须要解决。

我们大多数住在郊区，那里并非安全之地。犯罪的事偶有发生。我们锁上了窗户和大门，一条大狗为我们看门，我们相信它能吓住坏人。然而，如果你注意到了你的卧室窗户后面雪地里的脚印，你的安全意识可能会大大提高。你会采取积极的行动来检查入侵者。雪地里的脚印、地毯上的鞋泥以及门锁上的刮痕都是警示房主人存在安全隐患的线索。

滥用计算机系统的人也会在雪地上留下脚印。问题在于，普通的网络或者系统管理员不会查看后院的雪地，并且即使他们看了的话，他们可能也不知道那些通往卧室窗户的脚印意味着什么。

> UNIX系统可以同其他任何类似操作系统一样安全。然而，达到该目标需要很高的知识和很大的努力。因此，相对于其他技术来说，要更详细地探讨UNIX安全性问题。

在UNIX评论中，要探讨和解释电子化的“雪地里的脚印”。将分析该开放式系统环境的

脆弱性和易受攻击点。在介绍完有关问题的相关知识以后，将介绍并探讨解决方案。我们的一位大型机同事坚持说UNIX是天生不安全的，他戏称术语能够在字典中找到“UNIX安全”是作为一个矛盾修饰法的例子。尽管并不像某些人所想得那么可怕，但UNIX确实向那些希望在UNIX中增加信任的人提出了很多挑战。为了迎接这些挑战，需要掌握基本的UNIX控制。也必须要明确可能出现的问题和必须解决的基础问题。有关可能解决方案的讨论，将在下一章继续进行。

10.1 UNIX安全性名声不好的原因

无论何时，当介绍UNIX安全主题时，我通常都会以对1964大众甲壳虫汽车的讨论作为开始。我肯定没有机械故障，但即使存在机械故障那也是可以找到修好1964年制造的甲壳虫汽车。所有一切都是可以了解的、容易更换的，我甚至能够找到燃油泵！拿一个螺丝起子、一条传输带和一些导线，普通的年轻人就能够让甲壳虫多跑几公里。很多人都非常了解甲壳虫的基本机械故障。如果我不能修理某个毛病，那么我的一个朋友可能行。

另一方面，当打开新福特金牛座的引擎盖时，我发现它的引擎非常复杂，根本没法下手！今天的燃油引擎可以比做专用的操作系统，如那些可以在微型计算机和大型机环境中找到的系统。安全机制的内幕没有详细地检查。操作系统的源代码严格控制在专用环境中。但最重要的是，在大型机系统中，操作系统的集中式特性使得它自己能够严格控制它的资源。

很多人都随着UNIX的使用而长大，因此他们对UNIX内幕非常熟悉。有了公开的源代码，人们能够对系统的机制进行修补，并了解它的秘密。有些最坏的UNIX安全破坏就是因为破坏者掌握了其内部机制的知识。他们可以发现系统中的缺陷，并滥用其安全性。

然而实际上，UNIX中出现的大多数滥用行为是由于标准可用控制的不正确应用所造成的。知识是高效安全应用程序的关键，但是它对于UNIX环境来说作用更重要。本章和下一章的目的是让用户能够对UNIX安全机制有一个基本了解。我们将“打开引擎盖，四处拨弄，然后看看会发生什么”。

10.2 UNIX安全

UNIX已经问世很长时间了。最初它是在20世纪60年代末期作为一个软件开发环境提供的，70年代在学术界变得非常流行。UNIX操作系统的商业应用在80年代得到飞速发展。近几年来，作为客户机-服务器技术中的服务器组件，UNIX被广泛使用。导致UNIX如此流行的一个原因是它的开放性，这使得针对UNIX的软件和硬件解决方案的大量出现。UNIX也为操作数据库应用和快速开发应用提供了一个合算的平台。UNIX流行的最后一个原因是它具有强大的连网能力。UNIX机器能够同其他使用多种协议和服务的机器进行通信。

要知道，尽管UNIX可以使用一个强大的Web控制来配置，但是UNIX一般并不默认安装这些控制。环境需要系统管理员采取有关措施。另外，作为对这些控制结果的检查，UNIX安全控制必须要进行仔细的分析。与采用所有进程集中式控制的大型机环境不同，分布式环境中一定不允许采用既有的控制功能。随着时间的前进，UNIX控制趋向“恶化”，任何安全分析的一个关键组件必须是现有。新软件的添加可能会导致控制的“恶化”，例如，控制结构可能会改变。文件和目录可以用不严格的权限或现有的权限建立，而这种权限可能被改变。

10.2.1 物理安全

对于能够物理访问机器的入侵者，UNIX系统同其他计算机一样几乎没有提供什么保护措施。UNIX机器允许在启动时选择一个替代“引导”设备。入侵者可以把自己的磁盘、磁带或者CD-ROM连接到一个硬件控制器上。通过操纵合适的引导程序加载器，入侵者能够很容易地从他们的设备上引导系统机器，并获得对计算机的访问权。

许多UNIX系统有一个称做单用户模式的“后门”钥匙。单用户模式是进入UNIX系统进行维护的一个方法。UNIX假定任何知道该方法并且有物理访问权的人都是一个合法的系统管理员。从安全角度上讲，单用户模式令人担心，它绕过了所有的认证，并让用户作为权限最大的UNIX用户——超级用户进行登录。不必知道超级用户密码，只需拥有对系统控制台的物理访问权并知道如何使用后门钥匙。有些UNIX系统，如AIX和HP-UX系列，为引导顺序提供了安全控制。例如，HP-UX 10.0引入了可选的控制，当打断引导顺序时，就激活这些控制，并要求用户提供一个密码。

但是对于把信息存储在磁盘上的任何计算机来说，即使解决了非法引导问题，那些能够删除磁盘驱动器的人还是能对计算机进行攻击。与以往不同，现在磁盘驱动器能够很容易地加载到另一个计算机上进行检查，并修改磁盘上的控制和数据。如果修改了驱动器上的配置信息，然后再送回驱动器，那么入侵者就能够得到系统管理特权。如果入侵者能够物理访问系统，那么他们也能够破坏备份介质，从而得到重要的信息。只需制作备份介质的一份拷贝，密码文件就能够提取出来，然后使用密码破解技术就能够得到密码。关于这一技术，我们将在后面讨论。

10.2.2 UNIX认证

下面要介绍UNIX所采用的用户认证方式和UNIX安全的核心——/etc/passwd数据库。对登录到UNIX系统上的用户进行认证一般都是通过校验用户的密码来实现的。通常情况下，密码以加密的形式保存在/etc目录下的passwd文件中，这是一个ASCII文件。当一个用户要登录到UNIX系统上时，他需要提供一个账号名（称作用户名）和一个密码。用户名（或者账号）要同/etc/passwd文件中的值进行比较。如果有一项符合，那么就加密用户提供的密码，然后系统把输出字符串同/etc/passwd数据库中的加密密码进行比较。如果两个加密值相等，那么用户就通过了认证。UNIX使用的加密算法是DES加密算法的一个修改版本（相对弱些）。

表10-1给出了一个/etc/passwd文件的例子。

表10-1 /etc/passwd数据库示例

账号名	加密后的密码	用户ID	组ID	其他信息	主目录	登录程序
rot	8t6yrgbJJrg2d	0	0	,,,	/	/bin/sh
bin	*	2	0	,,,	/	/bin/sh
rdempsey	dJgrt2743ssdr	204	2	Rob D,.242-3387	/home/rdemps	/bin/csh
gbruce	tr56geJPhe34	0	204	Glen B,.242-7825	/home/grbuce	/bin/csh

/etc/passwd文件中的每个记录（行）都是UNIX中的一个账号。账号这个术语在UNIX中同用户名可以互换使用。每个记录中的第一个字段是用户名，这也称做账号——称做账号更为准确，这是因为该字段并不一定代表一个人。例如，账号bin和adm用来管理有关系统管理

的文件和目录。UNIX使用用户名来查找其对应的密码，并指定相应的用户ID。用户名的格式是无关紧要的，这是留给系统管理员的。如何起用户名一般有惯例，如用名字的第一个字母加上姓的前七个字母组成。需要注意的是，登录后，UNIX根本不关心用户名，真正起作用的是用户ID号。

/etc/passwd文件中的第二个字段是账号密码。默认情况下，密码字段以加密的格式能够被任何用户看到。如果密码字段包含一个空值（即空），那么这表明没有与账号相关的密码。所有账号的密码都应该受到保护，这是非常重要的。如果一个用户能够得到对一个不受保护账号的访问，那么控制系统就很容易了。攻击者可以使用这个账号在网络上的其他系统上发起攻击。如果“*”出现在密码字段，那么该账号就被认为是无效的，并且任何用户都不能使用它来登录系统。

密码时效允许系统管理员强制用户在预定义的时间间隔内改变他们的密码。密码时效是一个标准的UNIX特性，它是建立在一个前提上的，定期改变密码能够防止密码被发现。密码时效的实现方法是，在加密过的密码字符串后面添加一个“,”，然后再跟上一个密码时效字符（表示密码有效的周数）。

第三个字段是同用户名相关的用户ID号。访问资源的能力是建立在用户ID号的基础上的。用户名只是间接通过它同用户ID号的关系来授予访问权。按照惯例，HP-UX系统上的用户ID起始于200，然后顺序指定。顺序没有任何专门的意思，201不一定比226有更大的权限。此规则的一个重要例外是超级用户。超级用户一般指“root”的用户名。超级用户是任何其有效用户ID（详情在后面）为0的用户。在我们的例子中，gbruce和root都是超级用户。账号名同此没有任何关系，决定这些用户是超级用户的是数字0。将在后面详细介绍超级用户。

/etc/passwd文件中显示的第四个字段是默认组。UNIX基于用户的组ID号来授予访问权限。默认情况下，你被分派到一个其组ID号在你的/etc/passwd账号项中的组中。在我们的例子中，rdempsey和gbruce都被分派到组ID为110并且组名是users的组中。同样，用来定义访问权限的是组ID号。组名只是间接通过它同组ID号的关系来定义访问权限。

/etc/passwd中的第5个字段称作GECOS字段。该字段是可选的，它用来提供账号的信息。该字段一般推荐使用默认值（三个逗点）。GECOS字段中的用户信息（如电话号码）会带来安全问题。例如，入侵者能够使用用户的电话号码来确定用户是否在度假，而在相反情况下不能够检查他们账号中的行为。

10.2.3 用户的主环境

密码文件中的第六个和第七个字段定义了用户的HOME环境。其中，第六个字段确定了用户的主目录（称作HOME目录）。HOME目录用来存放个人文件、UNIX邮件以及启动脚本。启动脚本是ASCII文本程序（同DOS下的.bat文件类似），当用户登录时，启动脚本会自动执行。尽管没有什么规定不允许多个用户共享一个公共的HOME目录和启动脚本，但是推荐做法是对每个用户都指派一个私有的、安全的HOME目录。如果把未加限制的访问权授予用户的主环境，那么入侵者能够在该目录下放置修改过的启动脚本，并改变文件访问权限、获知密码等等。当用户进行登录时，启动脚本.profile、.login、.kshrc和.cshrc就会自动执行。对于其他用户来说，它们不应该是可读的或者可写的。如果一个入侵者能够在一个用户的启动脚本中输入命令，那么这些命令在用户登录时就会自动执行。这使得入侵者能够提交命令就好像他

们是合法用户似的。

最后一个字段定义了启动shell或程序。该字段包含一个程序名（在我们的例子中是/bin/ksh，这是一个UNIX命令解释器和编程语言），登录成功后，该程序会自动调用。大多数普通用户被自动放到一个应用程序中或者提供了一个应用程序选择菜单。

10.2.4 组控制

在用户登录时，根据/etc/passwd数据库中的组ID号，UNIX会把用户分派到一个当前组中。组名和成员是由/etc/group文件中的相关项确定的。表10-2说明了/etc/group文件的概貌。

表10-2 /etc/group数据库示例

组名	组密码	组ID	组成员
root	*	0	root
other	*	1	root、sys
bin	*	2	bin
users	*	20	gbruce、rdempsey、grants
admin	*	21	gbruce、rdempsey

同用户ID号一样，资源访问权限是在组ID号的基础上授予的。组名只是引用组ID号的一个便利方法。同用户ID一样，真正起作用的是数字而不是名字。组可以有与其相关的密码，但是该功能很少使用，通常通过在组密码字段中放置一个“*”来禁用。在一些UNIX版本上，用户自动是它们指派的全部组的活动成员。在其他版本中，用户通常同时只能是一个组的成员。不支持自动组成员的UNIX版本可能允许系统管理员来配置该功能。这为用户提供了每个组中的活动成员资格，而不要求用户公开改变他们的活动。

系统管理组中的成员可以让普通用户访问某些系统资源，而正常情况下访问这些资源需要高级特权（如超级用户特权）。系统组，如bin、adm和sys，应该具有成员限制并永远不能包括普通用户。

在大型机世界中，在组上重叠一个部门结构是非常常见的，这是因为多重组成员是难以实现的。在UNIX中，情况正好相反。建立附加组（称做特殊兴趣组）是相对容易的，但是如果没有足够的控制来限制组增长，管理工作可能很困难。组数目过多会为管理组成员关系和报告资源访问特权带来额外的管理麻烦。一般来说，推荐做法是把特殊兴趣组的数目保持在一个最小值。

10.2.5 UNIX认证中的脆弱性

/etc/passwd和/etc/group实现中的第一个脆弱性是这两个文件都是普通文件，并由标准UNIX访问控制保护以免遭篡改。如果用户能够得到对其中某个文件的写访问权，那么用户就可以修改文件内容。使用他们最爱的编辑器，能够删除超级用户的密码或者把他们自己添加到一个特权组。如果他们能够得到对/etc/目录的写访问权，那么他们就可以用他们自己的密码或组数据库来代替标准文件，并且能够提供他们自己的密码文件。

加密过的密码字符串是可读的，这也是一个脆弱性。使用加密过的密码字符串来发现密码的方法称做“破解密码”。在后面当介绍有关UNIX操作系统的典型滥用时，将探讨如何进行“破解”。

UNIX并不保持/etc/passwd和/etc/group数据库的一致性。没有任何东西保证在密码文件中引用的组ID一定存在于组数据库中。幸运的是，大多数UNIX系统提供了标准工具来检查组数据库的完整性。另一个问题是连网UNIX系统之间的用户和组ID的同步。例如，如果用户gbruce——在系统A上的用户ID是204——访问rdempsey（其用户ID也是204）所拥有的一个连网资源，那么gbruce能够获得对该资源的访问权。在分布式UNIX环境中，用户ID、组ID以及其他值的同步是一个大问题。将在第16章中探讨该问题及其可能的解决方案。

从前面可知，UNIX是在用户ID和组ID的基础上授予资源访问权限的。现在我们看一下这种授权方式在UNIX下是如何工作的。

10.2.6 资源访问控制

UNIX系统通过一个三级许可访问控制模型来保护系统资源，如文件和硬件设备。每个资源由一个用户ID（称作拥有者）所拥有，并且默认分派到拥有者的组中。请求访问资源的用户分为三种：资源的拥有者、资源所属组的组成员以及所有其他用户。资源的拥有者关系可以转移到另一个UNIX账号，并且组成员关系可以重新分派到另一个组。如果一个用户是资源所属组的一个当前成员，那么该用户对该资源的访问是允许的。

访问权限的类型包括读、写和执行，它们属于请求者（拥有者、组成员以及其他用户）上。根据资源是文件还是目录，这三种访问类型分别具有非常具体的意义，如表10-3所示。

如果对文件有读访问权，那么可以打开该文件并检查其内容。如果有写访问权，那么可以改变文件的内容。执行访问权允许文件（可以是ASCII文本也可以是二进制程序）执行。具有执行权限的ASCII文本文件称之为脚本程序，它等价于DOS下的.bat文件。其内容可以包括标准UNIX命令或程序。

表10-3 文件和目录访问权限

资源类型	访问权限	说 明
文件	r READ	能够打开和读取内容
	w WRITE	能够修改文件内容
	x EXECUTE	能够做为一个脚本程序运行文件
目录	r READ	能够列出目录下的文件名
	w WRITE	能够添加或删除文件
	x EXECUTE	能够列出文件的详细信息
		能够cd到目录

对目录的读访问权可以列出该目录下所包含的文件和子目录的名字。对目录的写访问权可以改变该目录的内容、删除文件以及拷贝新文件到该目录下。执行权限允许用户得到该目录下所包含的文件和子目录的详细信息。它也允许用户移动到那个目录下，并把它作为当前工作目录。

需要指出的是，提供安全性的是文件和目录访问权限的结合。没有另一个的存在，只有一个是没有用的。为了防止某人修改一个文件，必须限制那个人对该文件和对该文件之父目录的写访问权。如果他有对目录的写访问权，那么可以用他自己的文件版本来替换原始文件。

10.2.7 授权中的局限性

标准UNIX资源访问权限模型有很多局限性和脆弱性。它不允许为独立组或者用户定义

不同的访问权限。还有，访问类型的数目太受限制（读、写和执行）。其他的访问类型，如创建、修改和删除，需要包括进来以加强当前的访问控制模型。访问列表机制也是不够的（拥有者、组成员和其他用户）。它强制系统管理员创建很多特殊兴趣组。当系统管理员需要为单个用户定义访问权的时候，这点尤为可怕。另一个不合需要的方面是操作系统文件和二进制程序（包括操作系统本身）由同一个访问权限模型来保护。使用这种体系结构，有太多的目录和文件（一般情况下有几百个目录、几千个文件）需要保护。一种改进方式是添加安全授权的继承功能（可选的，根据系统管理员的判断），即强制父目录的授权被子资源所继承。

因为文件的所有者能够对该文件进行任何操作，所以说UNIX资源许可模型是任意的。文件访问权限可以调整，因此即使拥有者也不能读取它；另外，任何人都可以读取和修改一个文件。通过把拥有关系改变成其他用户，资源的拥有者也可以放弃资源。一旦放弃，最初的拥有者就不能控制那个资源的相关许可。在后面将看到用于操作系统文件和目录的控制结构中的这种脆弱性很容易滥用。包含系统二进制数据、启动脚本和数据的每个目录和文件必须要充分保护，这是非常重要的。因为有这么多的目录和文件需要保护，所以手工扫描一个文件系统中的问题是不可能。系统管理员必须使用一个自动工具来保证初始建立的控制能继续起作用和有效。将在第12章和第21章中介绍控制管理软件的使用。

10.2.8 访问控制列表

对标准UNIX许可的一个常见抱怨是它们不允许选择性的访问。如果一个选定的用户组需要访问，并且用户没有使用一个惟一的组来标识，那么必须要为这些用户创建一个特殊兴趣组。但是更重要的是，标准UNIX访问控制模型不可以为独立组或用户单独定义访问权限。它仅仅为一个用户组支持访问权限。除了资源的拥有者以外，所有不在该组的用户统统归类为其他用户。

为了解决该问题，有些UNIX系统提供了一种称为访问控制列表（Access Control List，ACL）的资源安全机制。通过资源的拥有者为独立用户或者组设置访问权限，ACL比标准UNIX访问权限提供了更高程度的选择性。ACL包含一组项，它关联到一个文件以指定许可。这些项通过文件的inode记录中的指针来引用。每个项为一个用户ID和组ID结合体指定了一套读、写和执行许可。每个文件可以允许多个项，并且也允许通配符。用户也可以专门从文件访问中排除出来，即使可以访问他们的组。当一个文件建立时，标准UNIX许可就自动映射到ACL项。

10.2.9 ACL的问题

每个资源必须实现自己的ACL。这使得ACL的管理变得很复杂、很耗时，因为当添加了新文件或目录的时候，没有任何自动机制能够更新ACL。控制继承——可以把ACL值传递到子目录和文件——使ACL可以在一个逆向目录树的选定级别上实现。默认情况下，所有的新子目录和文件会继承父ACL。大多数UNIX文件转移工具，如cpio、tar以及shar，不能认出ACL的存在。使用一个不支持ACL的工具所转移的任何文件都将丢失其ACL项。这为系统管理带来新困难，ACL必须要不断地查看以保证仍然有效。更糟糕的是，备份软件可能不支持

ACL的使用。如果备份解决方案不支持ACL，那么根据备份做的恢复会丢失当前的ACL。在网络上传输具有ACL项的文件能够导致项的丢失。UNIX厂商在ACL的实现上不尽一致。例如，列举ACL的命令在HP-UX系统上是lsacl，而在IBM的AIX系统上则是aclget。

10.2.10 超级用户访问

在UNIX环境的设计中，设计者有这么一个决定，给一类称做超级用户的用户授予对系统管理活动的绝对控制权。超级用户（任何其用户ID为0的用户）可以访问本地计算机上的所有系统管理和安全功能。这里的指导思想就是把系统管理限定到一个用户上，该用户执行所有的系统管理功能，从假脱机到用户的添加全都包括。

> 用狗来保护家的问题之一是必须要信任这条狗。你仍旧掌管着房子，并且可能相信这条狗不会伤害你，但是总是有那么一点可能性会使这条狗扑过来咬你。你可以把它的保护职责限定在地下室中，但是你需要保证没有小孩会打开地下室的门而让它跑上来。

表10-1表明用户账号root和gbruce的用户ID都是0。强烈推荐严格限制超级用户的使用。不管资源上应用了什么安全和控制，超级用户一概可以访问。超级用户可以删除所有的访问控制、破坏安全策略（如最小密码长度）、查看和改变任何文件、读取任何人的邮件以及删除密码。实际上，几乎没有什么行为是本地超级用户不能进行的。超级用户密码应该限定在最小的范围。那些委托了超级用户密码的用户不应该把超级用户访问权用于常规行为，如读取邮件。可以设置很多陷阱（后面介绍）来诱骗粗心的超级用户。如果超级用户谨慎地使用其权力，那么他们落入这些陷阱的可能性就小得多。

10.2.11 权力的委托

使用超级用户的指导思想是为UNIX系统提供集中式和单独的管理。过去，当系统没有多少用户，并且只能完成有限的功能时，这不是一个重要的考虑。对一个10用户左右规模的系统来说，单系统管理员思想还能够胜任，但是对于更大的系统来说它就不可行了。我们可能希望某个操作员进行假脱机，但是我们不希望他成为一个超级用户。不幸的是，UNIX的基本设计要求只有超级用户权力才能执行最基本的系统管理任务。中心问题是我们如何保证那些授予超级用户特权的用户不会滥用其权力。如果要委托超级用户特权，那么必须要采取措施来保证该特权不会被滥用，并且不会非法假定额外的超级用户特权。

UNIX允许用户使用switch user（su）命令来改变标识。他们必须提供希望改变标识的账号（即user）的密码——如果有的话。如果改变成功，那么UNIX会创建一个有效用户ID（effective user ID，EUID）。有效组成员也可以改变（例如，在HP-UX中使用newgrp命令）。使用su的问题在于一旦用户把标识改变到一个新账号，该用户就拥有了那个账号的全部访问特权。这不是一种权利的委托，而是一种权利的放弃。

UNIX使用一个非常特殊的组来访问一套特殊的系统调用，它称为特权组。在后面，将介绍如何使用商业解决方案来解决权利委托问题。

10.3 典型性的滥用

前面介绍了UNIX操作环境的基本安全结构。探讨了包括文件和目录结构在内的多种控制，

并介绍了它们是如何保护重要数据的。在下面，我们将提出一些问题并分析一下其他的问题领域。

1. guest账号

如果不管账号所给的权力级别就能得到对任何账号的合法访问，那么UNIX系统就可以很容易地被破坏。许多系统把“guest权力”扩大到包括未经认证的用户，这就无意中为黑客的访问打开了方便之门。guest权力可以扩大到包括没有密码的账号，或者具有如guest或者support之类的明显密码的账号。不安全的guest账号为UNIX操作环境提供了“没上锁的窗户”。不仅如此，因为不能识别个体用户，所以guest账号还会阻止正常的审计工作。

2. 欺骗

最早入侵计算机系统的方法之一是使用一个特洛伊木马来窃取密码。入侵者使用另外一个程序代替正常的登录程序，这两个程序都执行登录功能并记录密码。密码保存在一个秘密的位置，或者邮寄给入侵者。欺骗（spoof）就是执行模仿合法程序行为的程序。欺骗在本地系统上执行，它是一种特洛伊木马攻击形式。欺骗和特洛伊木马的区别在于：后者是一个不活动的程序，它需要执行以完成它的不正当功能；欺骗是不活动程序，它向用户提供一个伪造的命令行或者画面。欺骗的一种常见形式是向用户提供一个伪造的登录画面。该画面在各方面看起来都很正常，但实际上它正在执行一个欺骗程序而不是在操作系统的控制之下。输入账号名以后，会提示用户输入密码。密码输入以后，就保存起来或者邮寄给黑客。然后欺骗程序显示一条错误消息，例如“无效的登录”，其意思是表明用户输错了密码。然后，合法的登录程序会初始化，并且再一次提示用户输入账号名。用户的正常登录过程会接着进行，然后欺骗程序从系统中删除。用户很自然地认为他确实输错了密码。如果正确进行的话，这类欺骗行为是很难发现的。欺骗的任何迹象都必须作为一个重大的安全威胁来对待。连网的欺骗扮做一个有效的客户机或者服务提供者的伪造程序。在下一章中，将深入讨论这些内容。

3. 密码猜测

密码猜测是一个很老的学生技术，通过猜测一个账号的密码来尝试登录系统。猜测者输入thor、support以及dilbert之类的字符串做为密码，希望能碰碰运气。短时间内的大量登录失败可能意味着一个“雪地里的脚印”。尽管这有可能是一个蹩脚的打字员在作怪，但是它有可能表明有人在尝试猜测密码。UNIX并不在连续数次无效登录后就禁用账号，那不是它的标准功能。然而，系统管理员可以把无效登录尝试记录到/etc/btmp日志文件中。

可以很容易地开发软件以监视该文件，并在某个给定账号发生密码猜测的时候通知系统管理员。有些管理员自动禁用受影响的账号，但是这可能会使管理员遭到一个恶意攻击。如果入侵者向超级用户账号提供了无效的密码（可能故意），那么自动禁用该账号会使得超级用户不能访问系统。

> 作为UNIX环境中的一项黑客技术，密码猜测已经过时了，这是因为它很容易被检查出来，并且已经有很多可用的替代方法了。然而，有必要指出，黑客正在利用密码猜测技术来对付Windows NT和语音邮件系统。来自皇家加拿大骑警的一份最近的安全公告中报告了“一个称做VMB黑客的程序，该程序能够用来自动拨叫语音邮件电话线路并尝试存放在黑客建立的一个文件中的不同密码”。[RCMP]

4. 密码破解

对UNIX安全的一个更大的威胁来自于标准UNIX认证机制中的广为人知的脆弱性。采用的技术称为密码破解技术。为了理解密码破解的工作原理，我们首先需要介绍一下密码加密机制的工作原理。当创建密码时，密码的前8个字符同一个“salt”的时戳相结合。密码和salt都用做对加密过程的输入。salt是用来搅乱密码的。加密机制的输出是一个加密过的字符串，它包括salt和加密后的密码。该加密过的字符串保存在/etc/passwd文件中，它的前两个位置用来存放salt。当登录时，输入的密码和同输入的账号相关的salt都传送到UNIX加密机制。如果密码文件中的加密过的字符串同UNIX加密机制的输出相匹配，那么输入的密码就通过了检验，然后就可以登录了。

不幸的是，黑客们都非常精通密码机制的原理，并且能够加以利用。尽管加密算法本身能经得起破译的考验，但是标准UNIX密码数据库可以使用一个密码破解器进行攻击。黑客早就认识到，如果他们把字典中的每个条目都交给加密算法，那么他们就能够把加密过的输出同加密过的密码字符串进行比较。如果使用了大量的单词，那么至少匹配一个密码的可能性还是很高的。后来出现了一些算法，它们能够利用字典中的条目从而大大提高成功匹配的可能性。图10-1说明了密码破解器的工作原理。

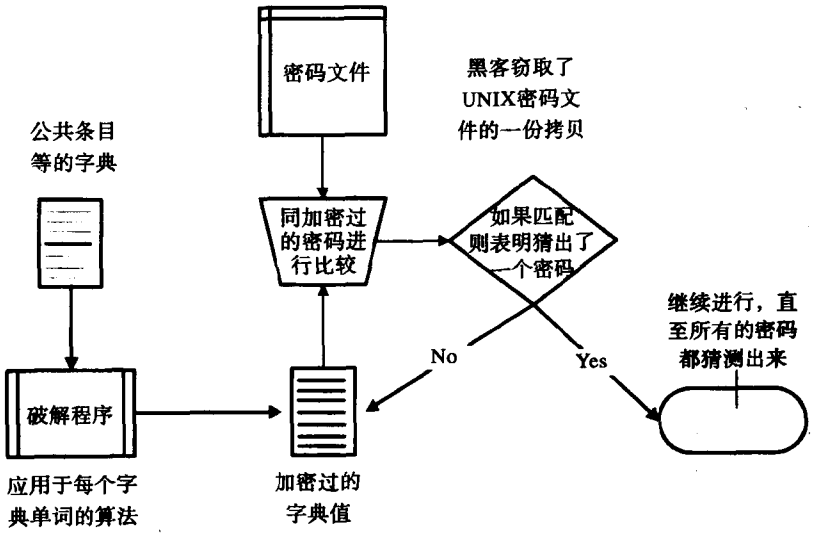


图10-1 密码破解

通常，/etc/passwd数据库只能被超级用户修改，但是每个UNIX用户都可以读取它。因为密码文件是可读的，所以能够拷贝和分析它。然后把加密过的密码同对字典中每一条进行加密后所得到的值进行比较，如果二者匹配则表明密码猜出来了。字典是用户自定义的，它可以包含普通单词、俚语、名字等。也可以调用算法来测试大小写敏感性，以及插入或追加特殊字符和数字。

健壮的密码能抵抗密码破解企图。这样的密码至少需要包含6个字符，其中大小写要混合，并且还要有特殊字符和数字。例如，F>Ru2n0就符合标准。应该指出，UNIX只对密码的前8个字符进行加密。还有，尽管一般情况下终端用户有最小密码长度的限制，但是超级用户并不受此约束。

为了表明密码破解技术的威力，UNIX系统管理员和同安全有关的人员都要了解密码破解工具。经验表明，使用一个不超过10000个单词的字典，在10分钟内就能够破解出一个普通UNIX密码文件中的10%的账号密码！卡内基梅隆大学的Daniel V. Klein对密码安全性做了分析，他测试了大量密码的抗破解能力，测试表明："结果令人吃惊，使用密码破解技术，很大一部分密码都可以破解出来。"应该认识到，通过使用含有几百万个单词的字典以及更为复杂的规则，然后连续数天进行破解，破解出的密码的数量还会更大。

对付密码破解的最好办法是向普通用户隐藏/etc/passwd数据库。由于加密密码字符串不能再拷贝，所以这样能保证密码破解器的失败。将在第12章中介绍的C2安全实现提供了这种能力。

5. 主环境

对直接访问UNIX系统的所有用户在登录时都提供了一个默认目录。该目录有时候称为HOME目录，这是因为变量$HOME保存了该目录的PATH或位置。HOME目录可以是多人共享的，也可以是每个用户独立的。后者是推荐做法，这是因为HOME目录下一般包含了很多在用户登录时自动执行的脚本。如果HOME目录或者这些登录脚本没有受到保护，那么它们就可以修改成自动执行不在用户权限范围之内的命令。其他用户对HOME目录的读访问也是不允许的，因为用户邮件和其他个人信息可能放在该目录下。

很明显，应该保护用户的环境，而对超级用户的环境则必须要绝对地进行保护。对超级用户登录脚本的修改会自动执行，并且超级用户在登录时可能察觉不到。在这些脚本中所添加的任何命令或程序都会以超级用户权限执行。这种攻击形式的一种变体是散布一个雷区。入侵者知道，启动进程是代表一个用户自动运行的，而不仅仅是在登录时。如果能修改启动脚本，那么它里面的任何语句都将在其所属用户的全部权限下执行。

关于布雷的例子可以从使用vi编辑器中看出。当vi编辑器激活时，它会查找一个称做.exrc的启动脚本。如果在当前工作目录下找到，那么该脚本就会自动执行。如果入侵者在很多目录下都放了一个修改过的.exrc（他们一般具有对/tmp和/usr/tmp的写访问权），当超级用户在这些目录下执行vi命令时，地雷就会突然爆炸。

在其HOME环境下，用户通常有很多默认设置，其中包括很多必须要仔细实现的变量设置。一个常见的错误是不正确地实现PATH变量。当提交一个命令时，系统会在PATH变量所设置的目录下查找可执行文件。如果PATH中包括了"."，那么当前目录就会在查找范围内。很明显，UNIX命令只不过是某个目录下的程序。例如，who命令就是/bin目录下的二进制文件who。PATH变量通知操作系统要在/bin目录下查找二进制文件who。如果"."是PATH中的第一个目录，那么当前目录就会在/bin目录之前查找。问题在于，如果黑客可以把一个修改过的脚本或者二进制文件放在当前工作目录下并为其命名为who，那么执行的就是这个who而不是那个真正的命令，这是因为在PATH变量中当前目录位于/bin目录之前。

6. 系统目录和文件中的特洛伊木马

UNIX操作系统包括了几百个目录和上千个文件。而各个厂商和各种应用软件则为环境带来了更大的复杂性。系统启动脚本、ASCII数据文件、二进制可执行文件以及库（所有这些组成了操作系统）都是普通用户可以查看的。只有很少数的操作系统功能是真正由操作系统本身所保持的，大多数功能都保存在普通文件中。UNIX命令只不过是这些文件的名字而已。UNIX使用标准UNIX文件访问权限模式来保护这些资源。对UNIX操作系统资源实行访问控制

要依靠文件和目录访问权限的结合。如果某个操作系统资源能被破坏，那么整个系统都有可能遭受到滥用。普通的UNIX系统一般有几百个目录几千个文件需要保护，所以只有使用自动工具才能保证维护访问控制得以保持，这是惟一的解决方案。

7. 启动文件

UNIX使用一系列的独立启动脚本来启动系统。主启动脚本通常称做/etc/rc，该脚本会调用很多其他相关脚本。在系统启动时，这些脚本会自动执行，并且使用非常高的特权。这些脚本中的所有命令是使用超级用户特权执行的。用于启动的目录（特别是/etc目录）和它们所包含的文件都应该保护以免遭修改。对这些脚本的任何修改都应该受到最大程度的怀疑。

8. 通过设置用户ID程序的权限委托

正常情况下，UNIX下的程序一般是在调用该程序用户的权限下运行。设置用户ID（SUID）程序是一种特殊的程序，它在该程序拥有者的权限下运行。SUID程序可以是二进制程序，也可以是shell脚本（一组ASCII命令）。它们都启用了一个特殊的权限位（称做SUID位）。SUID程序是与安全紧密相关的，因为它们是作为文件的拥有者而不是实际的用户来运行的。SUID程序中的弱点可以加以利用从而得到未经授权的特权。

根SUID程序是超级用户所拥有的程序，它们在超级用户的权限下运行。因此，它们是非常危险的，所以必须要受到严格的控制。如果某个运行根SUID程序的用户能够提交修改文件的命令，那么他就可以作为超级用户来运行命令。如果用户能够从程序中退出，并得到一个UNIX shell提示，那么这就更危险了。

如果一个用户能够在短时间内得到超级用户密码，那么他可以使用下面的命令来创建一个危险的SUID shell程序：

```
$su
password: <Enter the root password>
#
#cp /bin/ksh .anytime_I_want_to_be_root
#chmod 4555 .anytime_I_want_to_be_root
#exit
$
```

为什么这会有危险？这是因为/bin/ksh程序是一个UNIX shell或者命令解释器，它提供UNIX提示并允许用户输入UNIX命令。正常情况下，该shell运行在用户权限下，它没有特殊的权限。然而在上面的例子中，该shell被超级用户所拷贝，因此现在它由超级用户所拥有。由于在上面例子中使用chmod程序启用了SUID权限位（chmod 4555），所以该shell会作为资源的拥有者运行——那是root！每次这个新.anytime脚本执行时，它都会作为超级用户来执行。用户甚至不再需要提供超级用户密码！

UNIX操作系统（有时候还有它的应用程序）会包含很多SUID程序。这些标准SUID程序的出现不会被看做不正常，但是新SUID程序的出现有可能表明系统遭到了攻击。

9. 设备文件安全性

设备文件，也称做特殊文件，是UNIX操作系统用来同其硬件子系统进行通信的。这些子

系统包括磁盘、磁带、网络设备（例如LAN网卡）以及内存。需要认识到，设备文件可以绕过正常的UNIX、网络以及数据库安全控制。如果有对一个磁盘设备文件的访问权，用户可以查看UNIX文件或者数据库的内容，而无需顾及UNIX或者数据库控制。如果有对内存设备文件的写访问权，那么用户可以改变进程的有效用户ID并为其授予非法的超级用户特权。在有些UNIX系统上，LAN卡有可能被误用，这归因于LAN设备文件（称做/dev/nit）的松散安全权限。这是卡内基梅隆大学的计算机应急小组（Computer Emergency Response Team，CERT）在1994年说明的。由于设备文件能够提供绕过正常控制的能力，所以必须要严格保证它们的安全，并且要对新设备文件进行仔细的检查。

10. UNIX调度器

UNIX包括两个批调度器，at和cron。这两个调度器工作方式相同，但是二者之中cron用得更多。实际上，用户可以创建批任务调度表，并把它提交给调度器。提交过程要把调度表拷贝到一个安全目录下。调度器定期检查这些目录的内容，并在需要的时间调用批任务。尽管其他各种问题过去已经很明确了，但是对于调度器特别是at调度器来说，有两个重要的问题需要解决。

第一个问题是有关用于调度的目录的安全性的。提交给调度器的任务是运行在提交任务用户的权限下的。如果超级用户的任务调度表可以修改，那么其中一个未经授权的项就会以超级用户的权限运行。

第二个问题涉及到实际批任务本身的安全性。如果这些任务（实际上是由普通UNIX语句组成的脚本）可以修改，那么任务中的非法声明就会在用户的权限下运行。因此，一定要使用限制性的文件访问权限来正确保证批任务的安全性。访问权限不应该对调度脚本或者它执行的任何程序访问。

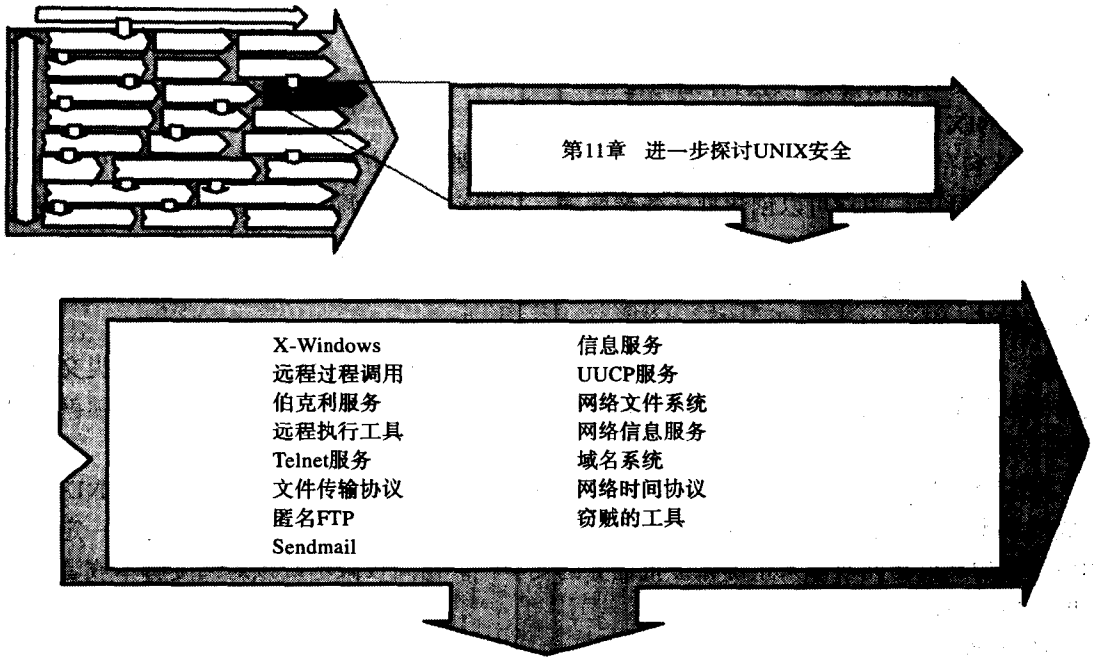
11. 备份

备份磁带的安全性是一个重要问题。除非介质被加密，否则大多数备份磁带都很容易读取。如果用户能够访问系统文件的拷贝，那么在没有保护的情况下访问系统就是非常有可能的。密码文件（甚至隐藏的密码文件）可以提取出来，然后破坏者就可以通过密码破解技术来破解其中的密码。备份磁带的物理安全和记录信息是非常重要的。保护不严密的备份介质敞开了系统的大门。

10.4 结论

保证UNIX系统的安全需要进行很多工作。在开始阶段建立一个正确的控制结构并定期监视控制是非常重要的。然而，积极地检查系统以防出现“雪地里的脚印”也是非常重要的，这是系统管理员的职责。有关黑客活动的所有迹象都应该严格的调查。在下一章中，将继续探讨UNIX的安全性。将进入UNIX最强大但也是问题最多的领域：网络。

第11章 进一步探讨UNIX安全



本章将继续探讨UNIX操作系统的安全性。首先看一下流行的UNIX用户界面——X-Windows系统。但是本章的首要着眼点是UNIX网络服务。通过多种服务，同其他系统进行通信的能力是UNIX系统最强大的功能之一。从安全角度上看，这也是问题最多的一个领域。

> 本章所分析的许多网络服务其实并不是UNIX的专有服务。例如，很多计算平台都提供telnet服务，如Windows、OS/2以及Windows NT。把这些网络服务包括在本章内容中，并不是因为它们是UNIX所专有的，而是因为它们最初出现在UNIX环境中或者是同UNIX环境有关系的。

同第10章一样，本章也将着眼于分析问题。这些问题的解决方案将在下一章介绍。本章最后将讨论一些表明系统中出现安全问题的信号。

X-Windows

MIT开发的X-Windows系统（X）是UNIX世界中的窗口界面的事实标准。X是一个支持用户交互的神奇工具。X非常易于使用，这为远程计算机间信息和图像的共享提供了优秀的功能。不幸的是，尽管X的体系结构是非常优秀的，但是在安全性方面X却有很多的问题。对于那些希望在X体系结构中拥有信任的人来说，他们必须要解决很多问题。

X体系结构的一个缺点是它没有认识到有些动作是特权化的。一旦通过认证，任何用户都能够无限制地访问X-Windows管理器的所有功能。拥有了对X-Window管理器的访问权以后，

不管对其他用户有何影响，用户的请求总是合法的。连到X上的用户可以关闭不归他们所有的窗口。他们可以控制其他用户（包括超级用户）的鼠标和键盘，并破坏X服务器的安全性。阻止未经授权的活动只有通过X-Windows访问控制机制。

对X-Windows服务器的访问控制是由一个称做xhost的程序和一套相关的控制文件（一般命名为/etc/Xn.hosts）来完成的。访问控制是基于主机名或者客户机的相关网络地址的。基于主机名或者网络地址的访问可能在单用户工作站环境中是合适的，但是它不适合于多用户环境。多用户环境需要能够识别独立用户的访问控制。

为了解决基于用户的访问控制问题，MIT开发了一套名为Magic Cookie的代替机制。MIT Magic Cookie（在X11R4发布版中引入）使用一个共享的密钥来对独立用户进行认证。该密钥（实际上是一个由计算机生成的字符串）保存在文件.Xauthority中，该文件位于用户的HOME目录中。如果该文件可以拷贝，那么Magic Cookie就可能被窃取并用于欺骗目的。Magic Cookie在LAN上是可见的，使用包截获技术可以探查到它。

X-Windows的X11R5发布版中引入了另外一套访问控制机制，其术语名称是XDM-AUTHORIZATION-1。该机制使用DES加密算法来保护LAN上的用户密钥。然而对于全球范围的用户来说不幸的是，DES是受美国国防部出口限制的，所以在美国境外使用XDM-AUTHORIZATION-1是受限制的。

通过替换认证策略并允许对所有客户机的访问，任何X用户都可以削弱认证过程。这是因为X缺乏特权用户的概念，所有经过认证的用户都授予对X服务器的全部访问权。尽管存在解决认证机制削弱问题的方案（XDM-AUTHORIZATION、Kerberos或者公开密钥技术），但X的体系结构使得不可能全面解决管理访问控制的问题。

对于X-Windows安全性问题，目前有三种解决方法，其中有一种是可行的。第一种是允许暴露，这肯定是今天最常见的回答。第二种是使用过滤（使用过滤软件或者硬件路由器）把X-Windows客户机限定到具体的网络地址。第三种方法是部署一个提供更为安全的X-Windows版本的商业产品。访问控制机制必须要不断地评审，从而确保它的存在和功能性。对访问控制策略的改变，例如发现对X-Windows显示管理器的通用访问，必须要作为最高的怀疑对象进行对待。

11.1 UNIX网络服务

UNIX服务器可以提供很多种网络服务，其中包括终端连接、文件传输、磁盘和打印共享服务、本地命令的远程执行以及进程间通信。客户机可以自由请求服务器上的任何服务。服务器有能力决定它们要实现哪些网络服务，以及在某些情况下决定同哪些客户机进行通信。需要认识到，每种UNIX网络服务都有某种形式的安全暴露问题。只有那些会被真正请求的服务才应该开启，但不幸的是，顾客购买到的大多数版本的UNIX都默认开启了大量的网络服务。在下面，首先介绍标准UNIX网络服务的工作原理，然后探讨一些主要地网络服务，其中包括ARPA、伯克利和UUCP家族。

11.1.1 标准UNIX网络服务的工作原理

UNIX系统使用一个称做inetd守护神的特殊后台进程来激活向它们请求的网络服务。当去往UNIX机器的流量被LAN接口捕获到时，inetd守护神就会激活。UNIX网络服务可以分为两

类：已知服务和远程过程调用。

广泛使用的服务通常使用一个特殊的数字来标识，该数字称做端口号。客户机发送到服务器的包中包括的是端口号而不是服务名。例如，用于telnet服务的端口号通常是23。当服务器接收到一个以它为目的地的包时，它会根据端口号在/etc/services文件中查询请求的服务。如果找到了，服务器就会启动在/etc/inetd.conf数据库中找到的守护神进程以同客户机对话。如果查询失败，那么就会返回到客户机一个通信错误。网络文件系统（NFS）是个例外，它使用远程过程调用来通信。

图11-1提供了一个简化的概图来说明UNIX网络服务客户机和服务器是如何进行通信的。

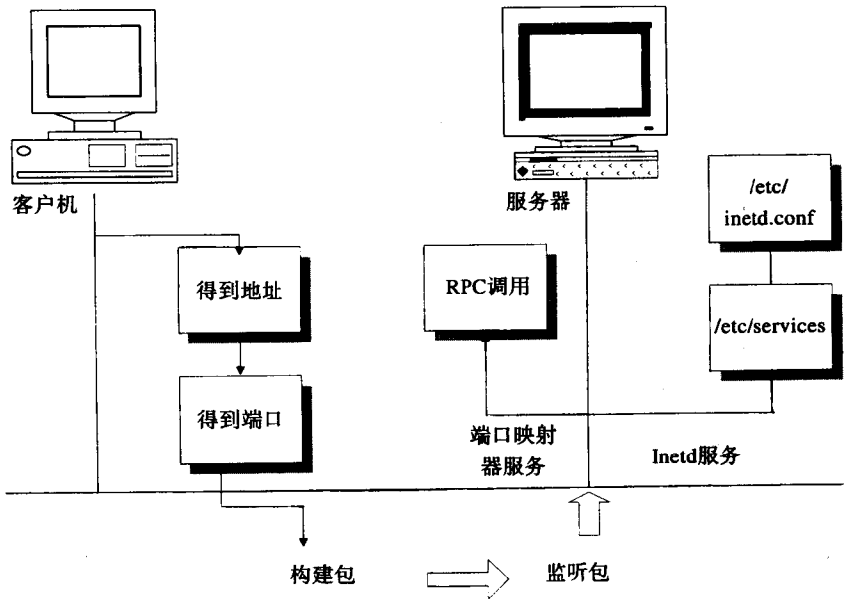


图11-1 UNIX网络概图

11.1.2 远程过程调用

远程过程调用（Remote Procedure Call，RPC）是一个用于客户机-服务器通信的方法。程序员创建一个接口，该接口定义了服务器程序中子例程的位置，并为请求的服务器分派一个特殊的程序号。通常，服务器程序不使用一个已知端口，而是使用一个特殊程序（称做端口映射器）的服务来同客户机进行通信。端口映射器在服务器启动时注册服务器，并同服务器协商一个端口号。客户机通过联系端口映射器来确定使用哪个端口号，然后客户机使用端口映射器返回的端口号同服务器进行通信。

有些RPC机制可以被愚弄，从而认为它们正在同一个已经认证的用户进行对话。UNIX网络服务通过使用序列号来跟踪它们位于客户机-服务器认证过程的位置。有些入侵者可以使用一种称为顺序攻击的技术来进行攻击。该技术的思路是通过向服务器提供一个精心设计的序列号来绕过认证过程。在这种情况下，服务器会认为用户已经通过了认证。包括NFS在内的许多UNIX服务器都曾遭到过这种攻击。

11.1.3 伯克利服务

伯克利服务（包括命令rsh、remsh、rlogin以及rcp）用来进行远程命令执行、远程登录以及文件拷贝。伯克利服务的首要问题是他们允许主机等效性。主机等效性是一个基于如下思想的概念，如果一个用户或进程已经在一个可信计算机（主机）上通过了认证，那么就没有必要在第二台计算机上也对该用户进行认证。这里假定第二台计算机信任第一台计算机，并从而把认证的等效性扩展到文件系统。图11-2提供了一个说明UNIX中主机等效性的概图。

开发主机等效性工具的首要动机是为了它的方便性。主机等效性使得用户可以很容易地在多个计算机上执行任务但不必在每台独立计算机上都进行认证。主机等效性的问题是，如果一台主机能遭到一次安全攻击的破坏，那么同该主机等效的所有系统都会很容易地破坏。1988年的因特网蠕虫攻击所采用的方法之一就是利用主机等效性来绕过安全性。

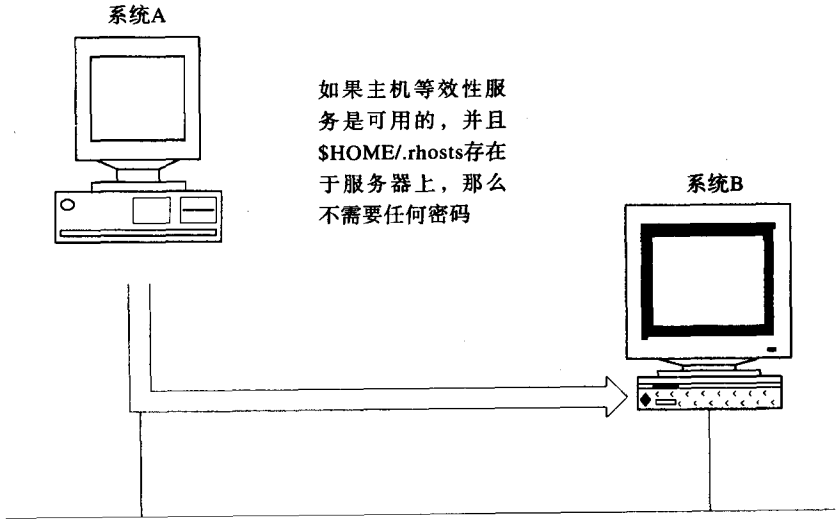


图11-2 UNIX主机等效性概图

主机等效性是通过使用.rhost、.netrc和/etc/hosts.equiv文件扩展的。应该指出，在限制最少的形式下（默认情况），主机等效性可以被普通用户扩展到其他系统。如果一个用户在他的HOME目录下建立一个.rhost文件，那么当从其他系统上访问这台计算机时该用户就会为其自己扩展主机等效性。用户不要求系统管理员的干预。为了解决该问题，系统管理员可以启用另一种形式的更受限制的主机等效性。通过重新配置，可以把主机等效性从普通用户中取消，并且要求必须通过/etc/hosts.equiv文件。正常情况下，普通用户是不能修改这个文件的。

尽管主机等效性是绝对不予鼓励的，但是应该认识到有时候主机间的等效性是一项功能需求，这种情况是存在的。例如，有些分布式备份方案需要客户机的root账号把主机等效性扩展到中心备份服务器。除非有某些原因使得伯克利网络服务非用不可，否则请把它们禁用。所有的.rhost文件都应该仔细检查。它们可能会使系统遭到破坏，并间接地破坏到其他系统。它们可能就是“雪地里的脚印”。

11.1.4 远程执行工具

远程执行工具（Remote Execution Facility，REF）是用来在一个服务器上提供命令的远

程执行。该功能同伯克利remsh（或者rsh）工具类似。REX具有特殊功能，它可以通过使用NFS把用户的本地HOME环境自动加载到服务器上。REX也把本地用户设置传送到服务器上（以环境变量的形式）。图11-3是一个REX概图。

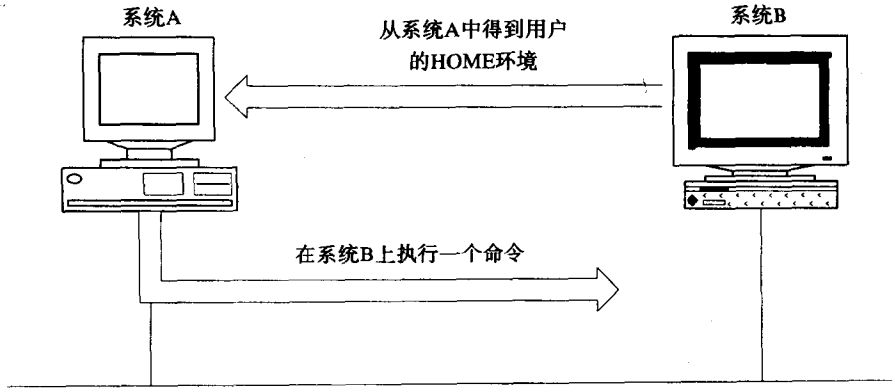


图11-3 远程执行工具

在信任方面，REX有很多问题。REX的默认实现允许每个用户都可以访问服务器，这是建立在用户ID号基础上的。超级用户是个例外，他们不能使用REX。这种模式对提出请求的客户机不进行任何认证。拥有自己工作站的任何用户可以使用任意一个用户ID。即使系统号受限，用户ID号在客户机系统上也是相对很容易操作的。

REX能够把用户本地环境（如他们的磁盘）加载到服务器上，这也增加了安全性的复杂程度。尽管加载一个外部磁盘系统需要具有服务器上的高级授权，但REX用户还是能够使用某种方法把本地磁盘或外部磁盘系统加载到服务器上。

11.1.5 Telnet服务

telnet服务用来提供远程（如在LAN上）终端访问。正常情况下，一个成功的telnet登录会为用户提供一个shell提示，但是它也可以配置成提供一个应用程序或者数据库菜单。UNIX服务器对telnet用户的认证方式也是使用标准的/etc/passwd文件，这同前一章中所介绍的标准方式完全一样。telnet服务不提供LAN上的加密，所以任何telnet用户都可能会受到包截获攻击。如果一个用户使用telnet成为超级用户，那么他就有可能会把超级用户密码暴露给LAN上的监听者，这是因为密码是以明文形式传送的。

因此，许多系统管理员限制超级用户的telnet访问只能在系统所直接连接的系统控制台上进行。系统控制台同主机的通信并不通过LAN，所以密码截获攻击无从下手。在大多数UNIX系统上，这种限制（包括rlogin）是通过一个称做/etc/securetty的文件来进行的。

11.1.6 文件传输协议

文件传输协议（File Transfer Protocol，FTP）是用来在系统间传输文件的。在FTP的标准实现中，用户必须要有服务器上的一个有效UNIX账号。FTP认证使用标准UNIX密码机制。FTP有一个特殊之处，它要求用户在连接之前提供密码。空密码是不允许的，超级用户更是如此。连接以后，用户可以拷贝任何他具有读访问权的文件，包括/etc/passwd文件。在其具有写访问权的目录下，用户可以创建新文件，或者替换已有文件。

同telnet一样，FTP在LAN上传递密码也是采用不加密的明文形式。因此，FTP也会受到密码截获攻击。因此，系统管理员通常把FTP的使用只限定到没有高级特权的账号上。文件/etc/ftpusers可以用于此目的。该文件中所列举的所有账号都禁止通过FTP对本地系统进行接入连接。外出FTP会话不受该控制的影响。

通过一个名为.netrc的文件，FTP也允许使用主机等效性。如果该文件置于用户的HOME目录下，并且包含接入系统的名字，那么用户就不需要提供密码。但是不鼓励使用.netrc及其他形式的不可信主机等效性。

11.1.7 普通文件传输协议

普通FTP（Trivial FTP，TFTP）是FTP的一个简化版本，它允许不进行认证的文件拷贝。TFTP主要用在X-Windows服务器环境下，X终端通过它来下载字体和其他信息。正常情况下，TFTP服务保留了对选定目录的访问权，但是老版本的TFTP允许对整个系统的不受限访问。应该指出，有些路由器把TFTP用做软件配置的一种传输机制。这种方法的可靠性是非常值得怀疑的。一般来说，如果没什么特殊功能需要TFTP，那么就应该禁用它。

11.1.8 匿名FTP

匿名FTP是一种更安全的FTP（相对于TFTP来说），它广泛用于因特网上公开发布软件的下载（对不可信客户）。匿名FTP使用chroot系统调用来伪装用户的文件系统视图。从用户的角度上看，匿名FTP目录是系统上的最高级的目录。用户不能改变目录到更高级或者横向的目录，也不能看到这些级别的目录或文件。操作系统文件和二进制程序受到了保护，这是因为匿名FTP用户根本不能看到或者访问它们！图11-4提供了匿名FTP的一个概图。

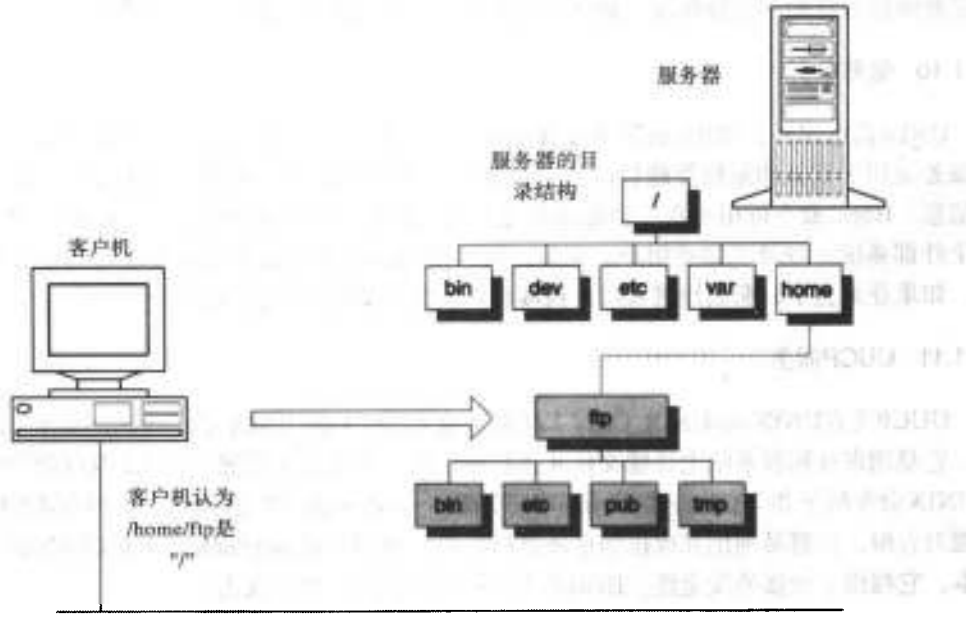


图11-4 匿名FTP

为了建立这种服务，认证文件和程序必须置于匿名FTP目录下。放在这里的文件和程序必

须要受到保护以免遭外部的修改。还有，认证文件不应该反映出真正的系统环境。系统管理员千万不要盲目的把/etc/passwd文件拷贝到匿名FTP目录下。它很快就会被拷走并破解！但是如果正确实现，匿名FTP是很难破坏的。

11.1.9 sendmail

UNIX sendmail程序是负责UNIX邮件服务的，它是一个高度复杂的程序。由于该程序及其伴随配置文件sendmail.cf的复杂性，sendmail程序早就成了入侵者的攻击目标。推荐做法是，除非有特别需要，否则禁用UNIX邮件（禁掉sendmail守护神程序并不会影响系统内部的邮件，它只会影响来自网络的邮件）。如果启用了UNIX邮件，请一定要装上所有最新的厂商sendmail安全补丁。

> 如果你收到了来自Boris Yeltsin和Mother Teresa的下班后去喝咖啡的手写邀请，那么你会立即询问同事。这种怀疑态度也应该应用到每一份电子邮件上。这不仅仅是安全性。发送者标识是很容易伪造的。对于一封在路上的信件来说，它很有可能会被熟悉邮政知识的人截获并修改。邮件通常放在一个公共邮件目录下，或者放在用户的HOME目录下。如果安全措施根脆弱或者根本没有，那么用户的邮件就很容易读取。

第二个问题是UNIX邮件进程本身。主文件及其伴随二进制程序是非常复杂的。看一下sendmail.cf文件可能足以吓倒很多系统管理员。由于它的复杂性，sendmail已经成了大量攻击的目标，其中包括因特网蠕虫和近来的HP/IBM/SUN报告。通过操纵sendmail程序来得到UNIX密码文件的一份拷贝是最早的技术之一。近来的许多报告表明，操纵sendmail仍然是可能的，并且将继续是黑客们的一种流行做法。另外，隐秘的可执行程序或者脚本有可能会作为正常邮件消息的一部分传输。如果不防止的话，其效果可能同收到一个信件炸弹类似！

11.1.10 信息服务

UNIX信息服务，如finger服务或者rwho，可以对未经认证的外部人员提供信息。创建这些服务是用于调试和系统管理目的的，但是攻击者可以使用这些服务来接收关于系统的有价值信息。finger命令可用来在一个远程系统上查找关于一个账号的信息。rwho命令可用来查看一个外部系统上登录了哪些用户。例如，可以使用rwho来检查系统管理员是否登录到了系统上。如果登录上了，那么攻击者可以稍候再来。推荐禁用所有这些服务。

11.1.11 UUCP服务

UUCP代表UNIX-to-UNIX Copy（拷贝）或者UNIX-to-UNIX Communication（通信）协议，它是用来在远程系统上传输文件和执行命令的一个软件子系统。UUCP也向用户提供了一套UNIX命令的子集。UUCP可以用在串行线路、modem或者X.25网络上。旧版本UUCP的安全能力有限，很容易利用其操作顺序来进行攻击。HDB（HoneyDanBer）是UUCP的一个改进版本，它提供了增强的安全性。HDB具有一些控制以防止顺序攻击。

UUCP的目录结构必须要仔细实现，并且其控制文件必须要正确地保证安全。例如，/usr/lib/uucp/Systems文件包含了用于远程UUCP系统的密码。如果安全措施不到位的话，有可能会导致这些系统遭到破坏。UUCP是20世纪80年代的UNIX通信标准，但是随着串行TCP/IP

服务（使用SLIP或者PPP协议）的流行，它的使用已经日趋减少了。如果不需要的话，请禁用UUCP。

11.1.12 网络文件系统

网络文件系统（Network File System，NFS）允许多个系统在网络上共享磁盘和CD-ROM文件系统，如图11-5所示。

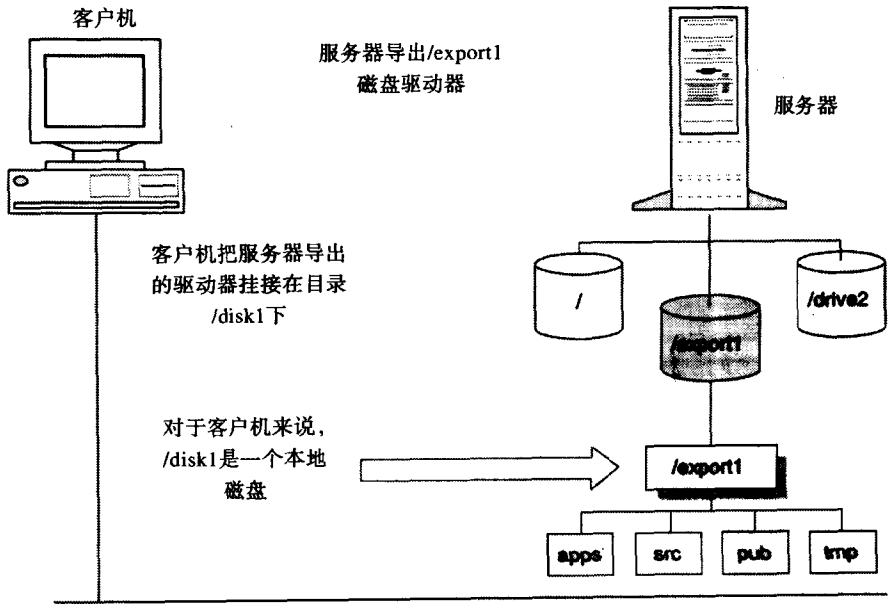


图11-5 网络文件共享

通过NFS，本地客户机系统可以透明地把其他系统的资源加载到本地目录树中。NFS把远程文件集成进本地目录树中不再需要显式的文件传输。客户机连接到服务器上以后，用户对服务器上文件和目录的访问就是完全透明的。NFS的使用并不仅限于UNIX，DOS和其他操作系统也支持。

连网机器可以是服务器或客户机，也可以二者都是。NFS服务器为客户机系统提供了文件共享。NFS服务器通常都是UNIX机器（但是MVS、OS/2以及其他OS也是可以的），而客户机可以是基于UNIX、OS/2或者DOS的系统。服务器是被动的，它们等待客户机请求服务。它们也需要客户机维护状态。换句话说，维护其在通信过程中的位置是客户机的责任。尽管这样会有客户机可以不受服务器重启影响的好处，但是它带来了安全问题。前面提到过，如果客户机能伪造一个当前“状态”的表示，那么NFS服务器可能会相信它们并允许其对目录和文件进行未经授权的访问。

使用NFS，必须要解决下面5个问题：

1) 应该导出什么系统资源？导出给谁？

2) 超级用户访问特权应该扩展给客户机系统吗？

3) 作为客户机，应该允许什么类型的文件进入系统？

4) 如何信任从一个NFS服务器上下载的文件的完整性？

5) 对NFS的认证和授权机制应该信任到什么程度?

在下面几段中，我们将探讨这些问题。

NFS服务器指定了客户机可用的资源。最安全的方法是只导出客户机系统绝对需要的那些资源和功能。如果导出服务器系统上的整个目录树，那么就有可能会破坏系统上的二进制程序，并且也可以访问服务器的密码文件。只有服务器上目录树的一个有限子集才应该导出。并且不管什么时候，服务器都应该使用“只读”形式导出文件系统。这样，客户机系统就不能修改服务器导出的磁盘驱动器。

NFS服务器有一个统一导出文件系统的选项。该选项会“授权”网络上的任何系统都可以访问导出的文件系统。一般来说，该选项是十分不安全的。在大多数情况下，NFS服务器应该直接并简明地指定它的客户机系统。一个可能的例外情况是以“只读”形式导出文件系统。

NFS服务器使用一个基于客户机名的访问控制机制。在这种情况下，一个恶意客户机可以通过改变它的系统名和IP地址来伪装成一个授权的客户机或者服务器。NFS没有任何内在的控制来对付伪装的客户机或服务器。恶意的客户机或服务器可以伪装成一个合法的机器，并绕过NFS的访问控制机制。

有关认证的第二个问题是NFS服务器和客户机互相默认信任对方的ID。许多操作系统命令和文件是由bin的账号所拥有的。如果客户机上的用户可以获得一个特权用户ID，例如账号bin的用户ID（正常情况下用户ID号是2），那么服务器就会为其授予对操作系统二进制程序的访问权。实际上，服务器对客户机的用户ID应该总是表示怀疑，并且永远不信任其认证的可靠性。不管什么时候，服务器应该只导出只读文件系统，或者限制用于导出的目录和文件。

NFS可以把等效性扩展到客户机系统上的超级用户。这明显是一个安全问题，即使你明确信任客户机系统上的超级用户。对客户机的破坏可能会牵连上服务器（除非服务器上的文件系统是只读导出的）。可选的“-root=”导出参数能够扩展超级用户等效性。如果超级用户等效性没有扩展，那么客户机系统上的超级用户会指派一个令牌用户ID（例如以-2作为用户ID号。）

对于客户机系统来说，主要的安全考虑是防止从服务器上导入任何安全性麻烦。如果允许服务器上的用户访问客户机的设备，那么来自服务器的设备文件可能会带来安全问题。前面提到过，SUID程序，特别是根SUID程序，可能会导致安全问题。如果用户能够从根SUID程序中创建shell，那么他就会成为超级用户。NFS可以禁用服务器的设备文件和SUID程序，在客户机上实现该选项是非常明智的。

前面提到过，NFS服务器是无状态的，这是因为状态是由客户机所保持的。客户机保持一个NFS文件句柄，该文件句柄维护了客户机对NFS服务器的访问权限。这种方法的脆弱性在于，服务器不会对来自客户机的请求进行认证，而是相信文件句柄的内容表明了客户机是可信的。在NFS的早期版本中，猜测文件句柄的组成是很容易的，并且可以使用一个伪造的文件句柄来向服务器提出请求。

一个伪造的NFS服务器能够截获NFS客户机发出的文件请求，并以比合法的服务器更快的速度做出响应。因为客户机没有任何办法来确认文件的来源或完整性，所以冒名顶替者就可以下载他们所要的任何文件。如果所请求的文件是可执行程序，这点尤为危险。

11.1.13 网络信息服务

UNIX系统需要很多数据库来实现系统管理和连网。文件/etc/passwd和/etc/group是用于用户认证和授权的。/etc/hosts或其等价方法是用于把系统名映射到IP地址的。由于缺乏集中的存储，所以每个数据库和值都必须要由每个机器进行本地维护。维护这些数据库在网络上的一致性是一个耗时的管理任务。独立数据库值变得不同步的可能性是非常高的。例如，如果用户rdempsey在系统A上分派了用户ID号209，用户gbruce在系统B上分配了同样的ID号，那么当这两个系统在网络上交互时，谁才是具有用户ID 209的用户呢？答案是两个用户都是！任何使用用户ID的网络服务都不能区分用户gbruce和rdempsey。图11-6提供了NIS映射的一个概图。

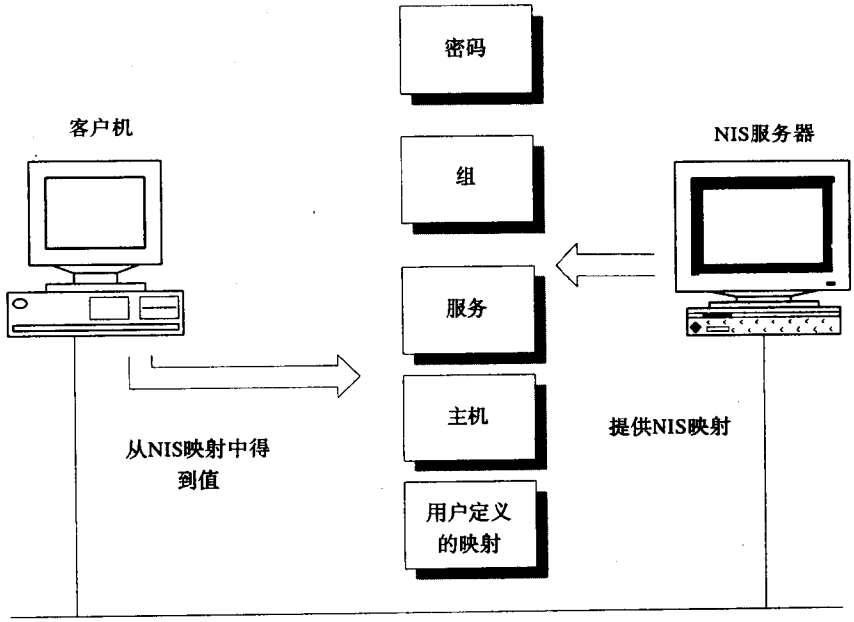


图11-6 NIS概图

网络信息服务（Network Information Service，NIS）是一个分布式的查询服务，它用于向连网客户机提供一致的数据库值。开始它被称做黄页（Yellow Pages）（但是这侵犯了英国电讯的版权）。NIS广泛用来解决系统管理数据库值的同步问题。NIS允许一个NIS主服务器（每个NIS域内一个）把它的表（称作映射）中的数据库值传送到第二个NIS辅服务器。辅服务器用来在NIS主服务器出现故障的时候提供冗余。NIS客户机把它们自己连接（绑定）到辅服务器上以得到所需的数据库值。可被映射的数据库没有数量或者组成限制（只要是ASCII普通文件就可以）。NIS管理员可以创建定制的NIS映射。

表11-1列出了NIS导出的标准数据库。

表11-1 标准NIS服务

NIS映射	UNIX数据库	目 的
密码	/etc/passwd	认证并分派用户ID
组	/etc/group	保存组ID号和名字

（续）

NIS映射	UNIX数据库	目 的
主机	/etc/hosts	把主机名映射到IP地址
网络组	/etc/netgroup	分派用户到NIS组
/etc/rpc	/etc/rpc	提供已知RPC服务的定义
协议	/etc/protocol	提供网络协议的定义
服务	/etc/services	定义已知的连网服务和分派端口

NIS也提供了一个本地查询服务。这用做对提供输出映射的一个完整视图的替代。如果NIS已经实现了，并且在本地/etc/passwd文件中的用户ID前出现一个加号（“+”），那么登录程序将从NIS密码映射中解析用户ID、密码以及用户ID号，而不是使用本地/etc/passwd文件。其效果相当于逻辑上用NIS主密码映射覆盖了本地/etc/passwd文件。

但是为了保护特定的账号，NIS允许选定NIS查询的本地解析。只有/etc/passwd数据库里的那些在第一列中包含“+”的账号才从NIS服务器上解析。所有其他账号仍使用本地/etc/passwd文件。这种体系结构的好处在于它能够在本地基础上维护具有高级特权的账号，如超级用户和数据库管理员。即使NIS密码映射用于普通用户，超级用户可以（必须）有一个每台机器惟一的密码。NIS也允许全局“万能”值的使用。如果实现了“万能”值，那么任何不存在于本地/etc/passwd文件中的账号都将从NIS服务器上解析。“万能”值的实现方法是在本地/etc/passwd文件中添加形如“+:0::::::::”的项。如果账号是有效的并且也提供了正确的密码，那么用户就登录到系统上，即使系统上不存在本地账号。如果HOME目录不存在，那么有些版本的UNIX会给出错误信息提示。

NIS网络组是用户或者机器的一个逻辑分组。这提供了一个系统管理工具（提供了方便）而不是实际增强了安全性。网络组用来使用一个公共的网络组名来对用户进行分组，而不是为数据库中的每个独立用户要求一个项。网络组可用来限制独立个体对NIS映射的视图。然而必须指出，网络组的实现已经证明对PC-NFS NISCAT命令没有任何作用（该命令为个人计算机用户提供了完整的NIS映射）。

NIS提供了能够列出NIS映射的命令。这些映射可以为希望破坏安全性的人提供信息。特别是，ypcat命令（称做NFS PC版本上的NISCAT）可以用来得到密码数据库的一份拷贝。尽管所有的密码都是加密过的，但是密码破解技术能够让它们现出原形。标准NIS不为C2隐藏密码文件提供支持。

NIS的使用不仅为系统管理员带来一些问题（特别是，密码和用户ID值的同步），它也带来了很多安全问题。NIS可能会为组织带来密码泄漏问题。如果NIS传递的密码成为公开的，那么恶意用户就能够访问大量的系统，而在非集中式的密码管理下这些系统本是他们所不能访问的。如果通过一个NIS映射系统管理员或者超级用户账号公开了，那么问题就更加严重了。在这种情况下，用户就可以访问该NIS域内的所有机器。

NIS允许NIS主服务器在系统中被辅服务器所复制。客户机可以直接选择绑定到一个NIS辅服务器，也可以连接到响应绑定请求的第一个辅服务器上。第一种做法有可能会使客户机遇到单点故障问题，这有违使用辅服务器的目的。大多数客户机选择绑定到第一个可用的辅服务器。客户机不对辅服务器进行任何认证。如果将一个恶意的辅服务器引入到网络中，那么用于每个NIS账号的登录控制就值得怀疑。仔细实现主服务器和辅服务器系统是非常有必要

的。如果NIS主服务器能被破坏，那么密码就有可能会被修改。

从客户机的角度上看，最后一个有关NIS使用的考虑事项是，NIS通常配置成为NIS域内的每个系统提供其所要求的最丰富的网络定义。NIS会为UNIX系统所使用的服务、RPC和协议提供一个中心定义。对于希望具有更严格网络控制的本地系统管理员来说，NIS管理员所允许的连网服务可能太丰富了。

前面所提到的许多NIS问题，如辅服务器的认证以及对隐藏密码文件的支持，都由NIS的一个改进版本——NIS+所解决了。NIS+使用了安全RPC机制来认证对NIS的访问和授权用户动作。不幸的是，安全RPC和NIS+并不是在所有版本的UNIX上都可用。

11.1.14 域名系统

在TCP/IP的世界中，系统名会映射成作因特网协议（Internet Protocol，IP）的特殊地址。UNIX系统可以以很多不同的方式来解析这种映射，包括使用一个本地/etc/hosts文件和网络信息服务（Network Information Service，NIS）。第三种方法是使用域名系统（Domain Name System，DNS）。该协议也可以用它的伯克利名字——Bind引用。DNS的使用有很多信任问题，这包括DNS服务器的假冒、命名树选定部分的失效以及暴露DNS提供的内部网络的信息。

11.1.15 网络时间协议

网络时间协议（Network Time Protocol，NTP）广泛用来在多个分布的计算机间为一个整体组织提供一个一致的时间概念。NTP允许客户机系统可以从很多源上同步时间（包括日时和秒）。为了冗余，NTP使用了第二个服务器——它从主服务器更新时间。客户机得到时间的方式是不断地发送正确的时间值（探询），或者通过远程过程调用来向服务器发送连续的查询。

出于很多目的，客户机系统需要依赖一个严谨的和精确的时间定义。这包括为访问控制和多系统事务的同步化建立准确的审计跟踪。对时间服务提供者的大多数攻击通常都是服务拒绝攻击。但是更复杂的攻击可以使用精确时间的中断来辅助破坏一个审计跟踪或者克服时间敏感的安全控制。

11.2 窃贼的工具

有必要指出，你的系统可能并不是入侵的目标，而仅仅是后续攻击的垫脚石。入侵者最初可能只是想把你的系统用做一个安全仓库，而不是想搞什么破坏。如果入侵者的行为很明显，那么本地管理员会采取对策让其住手。在这种情况下，入侵者的主要目标是隐蔽。通常情况下，黑客并不想被发现。他们会采取精心的步骤以防止系统管理员发现他们的行为。他们一般会小心谨慎，从而不进行会引起注意的行为。

攻击者可能需要磁盘存储来存放工具和相关的违禁内容，并且这些东西必须要隐藏起来以防系统管理员看见。他们使用那些用列举文件系统难以发现的空间或特殊字符来建立目录。工具的文件名会从原始名字改成其他不容易注意到的名字。日志文件和数据会归档和压缩。作为附加的安全措施，加密也广泛使用，这会使得系统管理员很难发现隐藏的目录。如果系统中没有加密工具，那么这些目录就会转移并隐藏起来。这种情况在美国之外是很常见的，因为在美国境外操作系统一般不包含DES数据加密工具。磁盘挂接点（挂接外部文件系统的本地目录）可以用来隐藏东西。如果挂接了外部文件系统，那么存储在挂接点下的文件不会

发现。但是当文件系统卸载的时候，那些文件就可用了。记录入侵者行为的日志文件，如wtmp、syslog或者shell历史记录文件，可以修改或删除。

> 每个窃贼都有其特殊需要。他们需要如垫片、撬棍以及锁之类的工具来辅助他们的非法入侵。他们需要安全的仓库来存放这些工具和他们的赃品。黑客都有同样的基本需要。他们需要自动工具来检查系统的脆弱点所在。他们需要具有存储数据（如日志文件）以进行后续检查的能力。他们也需要有安全的地方来存放他们的工具和数据。

入侵者的工具箱中可能包括很多东西。自动检查UNIX系统安全性的脚本是最普通的工具。SUID程序和设备文件、密码破解器、最爱的编译器、网络探测器以及调试器都是必备的工具。黑客们也对诱骗信息非常感兴趣。他们创建网络流量的日志文件，并在其中扫描密码及其他有趣的信息。他们会拷贝密码文件，并且研究如何列举远程过程调用服务。如果系统中出现了包含密码列表、服务器信息或者网络行为日志的文件，那么这表明可能有非法行为。

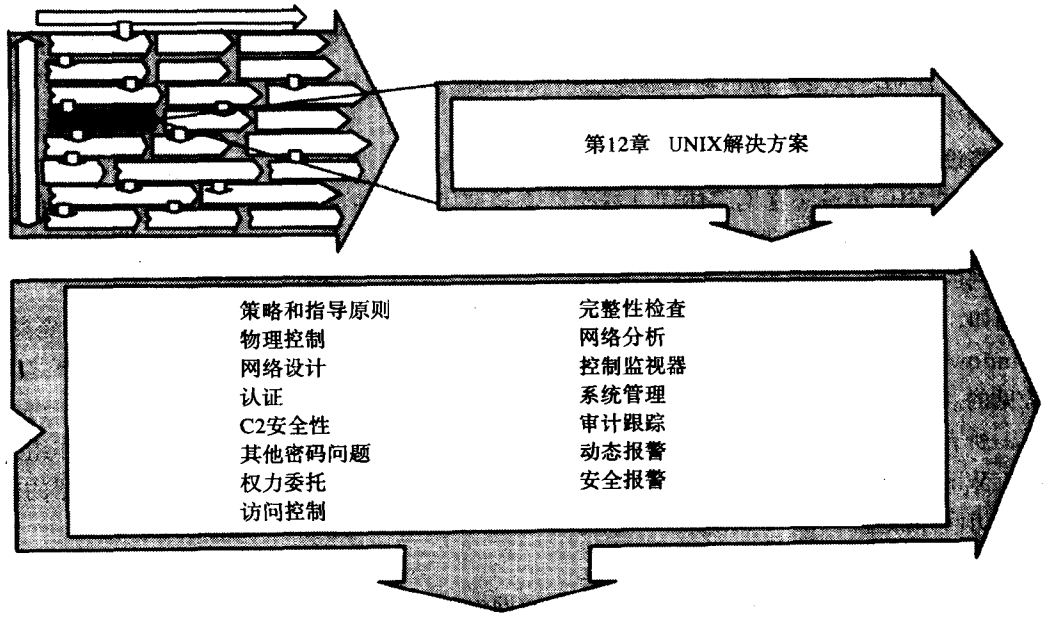
设备文件是操作系统和用户的中介。如果发现系统进行了配置以记录所有的网络流量（称作混杂模式），那么这表明系统可能受到了网络探查攻击。如果设备文件，特别是内存或者LAN设备文件，有了不正确的许可，或者位于正常的/dev目录之外，那么这也表明可能有非法行为。

11.3 结论

新购买的UNIX系统一般默认启用了很多网络服务，其中每种服务都为控制环境带来了复杂性。如果某项服务是不需要的，请禁用它。UNIX网络服务也提供了很多认证和访问控制机制，其中包括统一访问和主机等效性。但是最大的局限性是在TCP/IP传输机制中。因为入侵者有能力查看或者篡改包，所以需要使用新的方法来进行加密，从而保证传输机制的安全性。

现在已经探讨了包括UNIX网络服务在内的很多UNIX高级专题，并且分析了它们所带来的安全风险。在第12章中，将着重介绍一些针对这些问题的解决方案。

第12章　UNIX解决方案



在前面两章中，我们已经详细探讨了一些UNIX安全问题。这些问题都有相关的解决方案，本章我们将对其中一些方案进行讨论。在开始讨论之前，有必要先指出使用安全工具的几点内容。

> 在本章中，我们将探讨各种捐赠的和商业的软件解决方案。这些方案中的大多数都有关于其使用的许可和版权声明。他们不应该看做是免费的。在使用一个捐赠的软件方案时，读者应该特别小心以保证不会违背其使用规定。

第一点，尽管一个方案可能并不完美，但是“聊胜于无”，一个起码的方案总比根本没有强。并不能因为没有完美的方案就可以什么也不做。第二点，网络上最脆弱的系统是最有可能被破坏的系统，然后攻击者可以利用它再破坏其他更重要的系统。因此，正确的做法是全面地而不是有选择地部署解决方案，从而提高整个组织的整体安全水平。第三点，那些希望破坏系统的人会使用自动工具，所以你应该具有自动响应的能力。没有任何一项技术能够独立解决所有的问题，所以需要把多种技术结合起来使用。最后，技术不能代替人的智慧。即使找到了解决方案，有关环境的知识还是需要的。

策略和指导原则

一个定义良好的策略及其指导原则为解决方案的构建提供了正确的基础。如果没有针对UNIX环境的策略和指导原则，那么强烈推荐你开发一个。

物理控制

解决物理安全问题的方案很明显，即无论何时，请把UNIX机器放在一个物理安全的环境

中。敏感数据只应该存储在具有物理安全性的计算机上，除非有强大的补偿控制。谁都知道，一个价值几十万美元的计算机数据库服务器是绝对需要保证安全的。无论什么时候，计算机系统都应该放在一个访问受控的环境中，例如一个专门的机房。包含敏感数据或者需要高级信任的机器必须要重新放在物理安全的环境中。虽然不明显，但备份介质和网络设备，如路由器或者网桥，也是同样重要的。备份介质和存档应该在物理上同计算系统隔离开，从而防止在出现灾难的时候计算系统和备份介质同时丢失。

厂商也已经认识到了提高对计算机启动过程的控制是非常必要的。例如，惠普支持启动过程的密码控制。其他厂商也对问题应用相似的方法。尽管总体可靠性不能依赖于内部控制，但是这确实是对过去实现的一个改进。

网络设计

在第7章中，讨论了在网络中建立安全性需要使用的各种设计技术。例如，探讨了使用路由器或网桥来构建可信子网，从而防止未经授权的流量访问子网。UNIX网络服务也可以利用可信子网的好处。

例如，一个使用X-Windows的UNIX开发工作组可以使用路由器同公司的互联网络隔离开。X-Windows流量一般是在工作组本地的，这是因为使用X-Windows来交换信息在同事之间是最常见的。把X-Windows流量局部化比把其公开到公共网络上具有性能上的优势。在许多情况下，所有的X-Windows流量可以隔离到局部工作组中。在这种情况下，通过在工作组外部禁用X-Windows流量，安全性和信任都可以得到增强。这防止了对X-Windows服务器的未经授权的访问和探查。

认证

UNIX操作系统使用密码和其他方法（例如网络地址）来认证用户。按照惯例，密码应该具有一些特性，从而保证密码机制的安全性。表12-1列出了这些密码特性以及其后面的动机。

表12-1 密码特性及其动机

特 性	动 机
不可预测性	普通术语是很容易猜测的。难以破解的密码应该满足最小长度要求，并且要由多种成分混杂组成
定期改变	密码不能谁都知道，如果密码公开了，那么要减少知情范围
惟一性	对一个系统上密码的破坏不应导致对其他系统的破坏

捐赠的软件，如CrackLib，可以用来防止已被猜测密码的使用。许多版本的UNIX也支持实施一个密码策略，它可以规定可接受的特性。系统生成的密码是难以猜测的，并且禁得起破解，许多系统都支持该功能。但是强迫用户采用难以记忆的密码只会让他们把密码写下来。我们经常会注意到，有些用户把写有密码的“便签”粘在计算机屏幕上，或者把密码草写在纸片上。为了获得对其计算机账号的访问，许多人都这么干。

隔离的数据库、网络服务器以及多种不同操作系统和硬件平台的使用需要单独的认证技术。这种单独性强迫用户需要记住、维护用于不同应用程序、数据库以及LAN服务的多个密码。如果使用了密码时效并且是不协调的，那么用户只能使用同样的密码或者记住每个独立的密码。但最重要的是，这些关于密码的看似惯例的建议并不会解决对密码机密性的主要威胁。这些威胁包括密码猜测、密码破解以及密码在网络上的明文传输形式。现在我们来讨论针对前两个问题的解决方案，而针对最后一个问题的解决方案将在第17章和附录B中介绍。

针对密码猜测问题的解决方案是在它出现的时候向系统管理员报警。当UNIX发现了失败的密码的时候，它们要记录在一个本地审计跟踪中。审计跟踪可以检查，并且如果发现了一连串的失败密码时，系统会产生一个警报。审计跟踪的使用和动态报警将在本章和下一章中进行讨论。

密码破解者是非常非常有战斗力的。前面我们已经看到，密码破解器使用加密过的、但可读的密码字符串作为对一个密码破解算法的输入。该字符串会同字典条项的加密输出字符串进行比较。我们已经使用超过250万个条项的字典进行了密码破解。密码破解器使用很多规则或算法。他们可以测试大小写敏感性，并且在字典条项中插入数字或者特殊字符。在发现密码方面，他们是非常厉害的。尽管可以预防密码破解器使用，并且也有可用的捐赠软件来探查容易猜测的密码，但是对于密码破解的真正解决方案是把加密过的密码字符串隐藏起来。

C2安全性

作为C2可信系统，C2 UNIX系统提高了标准UNIX操作系统的认证和审计能力。一个C2可信系统设计成符合由美国国防部制定的桔皮书所定义的C2安全级别。不幸的是，C2已经成为用于UNIX系统信任和安全问题的万能药。必须认识到，C2的实现只解决了必须要解决的很多问题中的少数。这并不意味着C2的实现是一个坏想法（它是非常好的想法），但是它不能独自解决安全问题。

C2可信系统的实现基本上提供了两个好处。密码文件向普通用户隐藏起来了，这能防止密码破解。并且，操作系统调用的审计启用了。普通用户不能读取加密过的密码字符串。C2安全性要求所有的加密密码字符串都要隐藏起来。这些加密过的密码保存在一个位于一个安全目录（只能由超级用户读）下的文件中，该文件通常称做遮蔽密码文件。修改原始密码文件以反映新密码文件的存在，但是它保留了大多数的基于用户的信息。大多数工具（但不是全部）都支持遮蔽密码文件的使用。

C2也对UNIX操作系统生成的审计跟踪下了一个更好的定义。选定的用户组和系统调用可以结合起来以建立审计跟踪。例如，审计可以限定到用于普通用户和超级用户全部动作的安全相关系统调用。审计跟踪保存在一个安全的目录中，普通用户不能访问该目录。

分布式系统的大量使用突出了对使用公共认证的需要。公共认证可以用在很多系统和环境上。用户不愿意被强迫记住数不清的密码。解决方案是使用单一登录技术，如OSF/DCE所采用的Kerberos安全模型（后面将介绍）。

其他密码问题

前面我们已经探讨了密码破解和猜测问题。让组织头疼的另一个问题是用户和服务器间的密码共享到底是不是一个安全风险。即使加密了，密码也有可能在去往服务器的路上被截获和重放。许多组织正在寻找关于共享密码认证的替代方法以对付这种威胁。这些方法包括一次密码的使用和强大的认证技术，这些将在附录中进行介绍。

表12-2总结了针对上述问题的解决方案。

表12-2 密码问题和解决方案

问 题	解决方案
猜测	系统生成的密码 当发现猜测行为时报警
破解	遮蔽密码文件（C2）

（续）

问 题	解决方案
单一登录	Kerberos或者一次密码
明文密码	Kerberos或者一次密码

权力委托

UNIX系统管理任务的执行需要高级超级用户特权的使用。我们如何才能安全地委托超级用户的权力而同时不会使普通用户变成超级用户呢？幸运的是，对于此问题，捐赠的、厂商提供的以及独立的商业解决方案都存在。

捐赠的软件，如sudo程序，可以用来控制超级用户访问。sudo允许超级用户把高级权力委托给那些需要使用超级用户特权的人。它允许系统管理员建立可以使用高级特权运行的命令和程序表。同所有的SUID程序一样，sudo也有一个问题，如果能够从一个sudo控制的程序中退出并进入一个UNIX shell，那么你就成了超级用户。

厂商提供的系统管理软件，如惠普的系统管理管理器（System Administration Manager，SAM），为权力的本地委托提供了一个解决方案。SAM用来执行很多任务，这包括用户管理、资源配置以及网络服务的管理。SAM过去需要超级用户特权才能执行，但是HP-UX 10.0提供了管理任务委托的功能。

PowerBroker是来自加拿大卡尔加里Freedman Sharp公司的一个商业产品，它提供了安全委托超级用户权力的功能。PowerBroker为权力委托问题提供了一个网络化的解决方案，并且它具有使用一个中心审计来跟踪超级用户行为的功能。Memco软件公司的Security for Open Systems（SeOS）软件（UNIX版）提供了一个替代方法来限制超级用户权力。SeOS使用一个基于访问规则的数据库在操作系统级上截获安全敏感事件。该软件提供了一个积极的访问控制机制，该机制能标识出谁具有系统的访问权，并能在用户登录后马上跟踪用户行为。SeOS实现成UNIX OS的一部分，但是它不改变任何二进制文件或者重新建立内核。尽管通常已降至最低，但截获系统调用会导致系统性能的下降。用户需要对整体效果进行正确地分析。

今天市场上有很多可用的集中式系统管理工具，这也包括备份和假脱机解决方案在内。这些工具也为超级用户权力的安全委托提供了复杂的解决方案。

访问控制

访问控制指的是操作系统在其对授权用户或进程的控制之下对资源访问进行限制的能力。访问控制通常是建立在图12-1所示的一个或多个方法之上的。

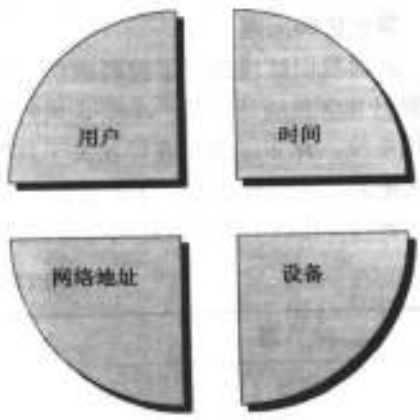


图12-1 访问控制方法

在UNIX中实行访问控制的标准方法是使用文件和目录许可。从前面可知，UNIX使用用户ID和组ID的结合来授予或者拒绝对资源的访问权。UNIX也可以配置成限制对选定设备或者网络服务的访问。例如，某些UNIX系统上的/etc/securetty文件可以用来强制从系统控制台进行登录。/etc/ftpusers文件可以用来拒绝选定用户对FTP的使用。这是防止超级用户使用FTP的最常见方法。密苏里州的圣路易华盛顿大学已经开发了捐赠的FTP

守护神（WU-FTPD）程序，该软件可用来增强FTP的安全性。其增强之处包括为独立用户建立chroot环境的能力，这同匿名FTP所使用的方法类似。同时也提供了更强大的访问控制。

根据系统名或网络IP地址的访问控制也是很常见的。例如，NFS提供的资源访问就是由系统名和IP地址控制的。HP-UX操作系统支持网络服务的访问控制，其方法是通过/usr/adm/inetd.sec。可以建立过滤模式来根据服务类型的选择性访问，这相当于一个软件路由器。例如，telnet访问可以限定到选定的地址；FTP可以配置成除了那些在工作组子网内的用户以外其他用户全都可以使用。

基于时间的访问控制也是很常见的。一般的基于时间的控制可以限制访问只能在正常工作时间内进行，并且在一段无交互时间后锁定屏幕。因特网上许多可用的捐赠软件也可以辅助访问控制。德州A&M大学的Drawbridge软件提供了强大的过滤功能。荷兰的Wietse Venema也捐赠了TCP包装器，它可以用来把过滤访问控制扩展到很多UNIX网络服务上。

完整性检查

硬件、操作系统和应用程序，当使用完整性来实现的时候，就结合在一起形成了可信计算基础（Trusted Computing Base，TCB）的计算环境。将在第20章中深入探讨这个概念。很明显，如果未经授权的改变能够作用到TCB的组件上，TCB就不再是可信的了。我们已经看到，计算基础可能包含几百个目录和几千个文件和程序。如何保证这么多资源的完整性呢？

第一步是通过强大的控制来正确地保证基础的安全，并检查这些控制（文件和目录许可）仍旧到位。第二步很重要，它涉及到完整性检查器的使用。

完整性检查器是建立在如下思想上的。在操作系统和应用程序安装完成以后，可信计算基础的一个快照就建立起来。该快照为TCB元素的一些经过选择的特性建立了一个数据库。过一段时间后，当前计算基础的当前特性就同保存的特性进行比较。对计算基础的未经授权的修改会成为一个“雪地里的脚印”，这是因为它们有可能表明系统中有时间炸弹或者特洛伊木马程序。对TCB的任何不清楚的改变都应该受到怀疑。图12-2说明了如何用一个外部CD-ROM来校验操作系统二进制数据的完整性。

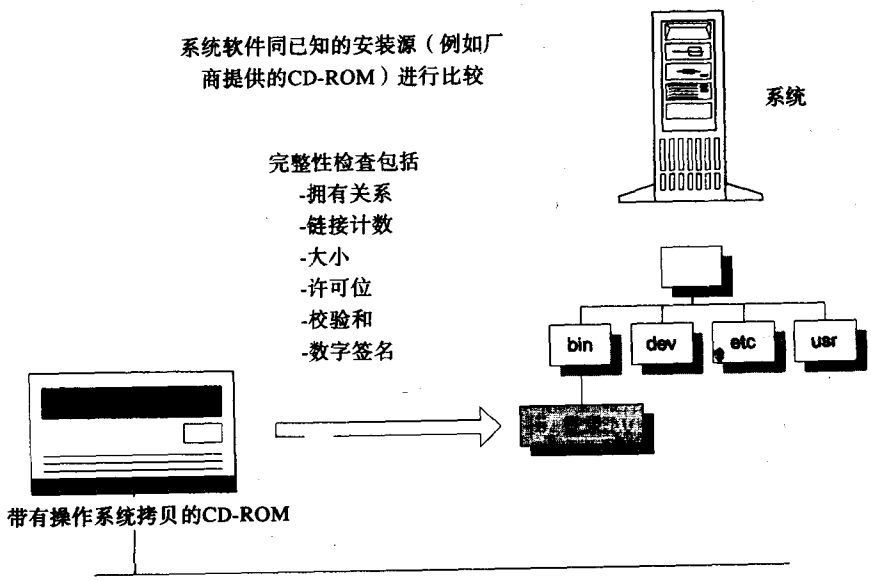


图12-2 完整性检查

必须认识到，完整性检查的使用有很多局限性。一个计算基础不是静态不变的，而是处在一个不断变化的相对稳定状态中。软件修正（例如补丁）、应用程序的新版本以及操作系统的升级都会改变TCB的特性。只有在保存计算基础特性的数据库是可信的情况下，完整性检查器才是有价值的。如果该数据库保存在一个基于LAN的系统上，那么它的数据就有可能会被攻击者修改，从而反映出计算基础修改了。一个更安全的方法是把TCB保存在可移动介质上，或者使用一个厂商提供的CD-ROM。

完整性检查器只有在能够保护它自己的数据库的情况下才是有用的。一般情况下，完整性检查器所保存的特性包括拥有关系、组成员、访问权限（许可位）、建立日期、文件大小以及一个校验和。UNIX系统上常见的一种校验和类型是循环冗余检查（Cyclic Redundancy Check，CRC）。CRC校验和相同并不是意味着完全可信，这是因为它们是建立在文件之特性的基础上的。黑客可以使用复杂的技术来模仿原始文件的特性，并生成一个伪造的、但却与原值相同的校验和，他们已经这么做过了。

一个更安全的方法涉及到使用复杂加密技术来为资源（例如一个文件）生成一个数字签名。他们使用文件中的真正数据作为算法的输入，因此对输入数据内容的任何改变（例如二进制代码修改）都会检查出来（即使很小）。RSA数据安全公司建议使用MD5校验和作为数据认证标准（在因特网RPC 1321中）。MD5产生的数字签名通常认为是抗伪造的。我们将在第15章中探讨整个加密技术领域，其中也包括数字签名在内。

必须要实施规定以保证：当经过授权的改变（如补丁或者版本更新）应用到计算基础上时，完整性检查器的数据库会更新。提供完整性检查的方式可以是把其作为其他安全工具（例如控制监视器）的一部分，也可以是作为一个单独的工具。一个非常流行的用于进行完整性检查的工具是Tripwire。

TRIPWIRE

Tripwire是普度大学的Gene Kim和Gene Spafford开发的一个捐赠软件产品。该软件的目标是探查对计算基础的未经授权的改变。Tripwire已经移植到了很多不同的环境中，并且它还提供了源代码。该软件对MD5校验和的使用提供了可选的支持。使用MD5校验和为探查伪造的程序提供了一个强大的功能。

脆弱性的网络分析

一类特殊的控制监视器，通常称做网络分析工具，可以用来对UNIX网络服务的脆弱性（有可能会被攻击者滥用）进行评估。这类工具提供了环境的一个网络（相对于系统来说）中心视图。它们测试UNIX网络服务在一套已知的脆弱性方面的强度和能力。一般的检查包括尝试获得对X-Windows管理器的控制和探查UNIX sendmail服务。其他行为包括尝试窃取密码文件、检查主机等效性以及非法使用FTP。

SATAN（Security Administrators Tool for Auditing Networking，用于审计网络的安全管理工具）是Dan Farmer和Weitse Venema开发的一个捐赠软件工具，它是在1995年4月5号发布在因特网社区上的。该工具结合使用了shell程序、PERL脚本以及二进制代码以测验几个已知的脆弱性。SATAN也有一个非常容易使用的GUI界面，它同许多因特网WEB浏览器所使用的界面类似。SATAN的引入带来了大量的争论。尽管SATAN只着眼于已知的脆弱性，但是由于其容易使用的特性，这类软件促进了黑客行为，这引起了广泛的关注。

12.1 控制监视器

监视软件的使用使得管理员可以定期、肯定地确认UNIX控制的存在并且功能正常。这类工具的主要功能是保证UNIX系统的“大门和窗户”都正确地锁上了。实际上，它们定期或者临时运行批任务，这些批任务可以扫描客户文件系统以查找脆弱的文件和目录许可、可写的系统二进制程序、网络服务中的不规则型等等。然后生成报告来警告安全或者系统管理员系统中出现了符合标准的问题或者其他破坏行为。图12-3说明了控制监视器的工作原理。

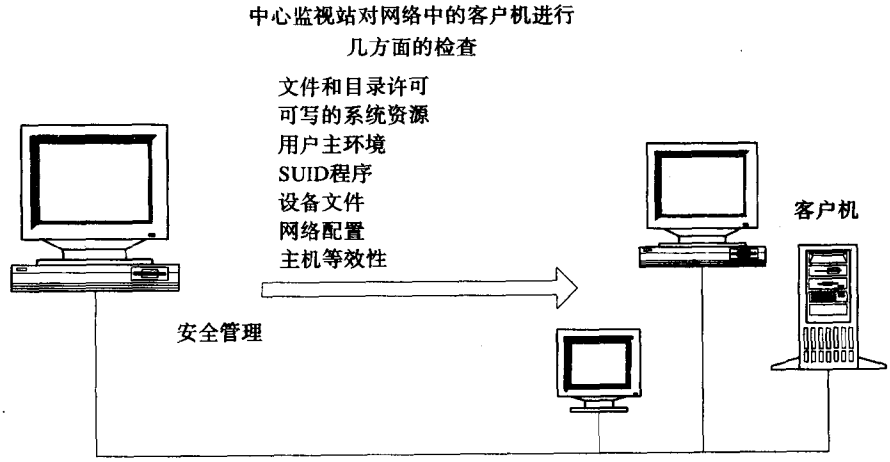


图12-3 控制监视器

对于这些工具来说，同探查一致性的能力一样重要的是发现“雪地里的脚印”的能力。对安全机制的未经授权的改变是系统被破坏的明确指示。控制监视器有能力探查UNIX系统中的已知脆弱性，因此它们可以起到报警系统的作用。

寻找这类软件时需要注意的一个重要功能是它们允许使用模板的能力。模板使得同客户机系统上的可接受标准的一致性可以惟一指定。如果不使用模板，控制监视器在其安全问题报告中就有可能会非常罗嗦冗长。雪地里的重要脚印可能会被忽视、跳过或者在辨别之前又被踏上几脚。这里的思想是，对被报告的信息进行剪裁，从而只报告那些同客户机系统相关的重大安全事件。控制监视器所做的检查包括：

- 密码和组数据库的拥有关系。
- 密码和组数据库的许可。
- 用于每个账号的密码或者账号会被禁用。
- 用户ID和组ID一致性。
- $PATH和$UMASK变量的正确设置。
- 其访问权限是松散的用户主环境目录和文件。
- 扩展主机等效性的文件。
- 其他用户可写的操作系统目录和文件。
- 所有SETUID和SETGID程序的报告。
- 所有可写设备文件的报告。
- 所有失败的访问尝试（登录、特权访问或者资源访问）。

- 如果UUCP启用了，那么/usr/lib/uucp/Systems和/usr/lib/uucp/Permissions文件必须存在且安全，/usr/spool/uucpublic目录同样如此。
- 对所有NFS全局和超级用户启用的导出的报告。
- 所有允许SUID和设备文件导入的挂接。

上面这个列表列举了控制中可能出现的一些脆弱性，但它并不是完全的。控制监视器分为三类：捐赠的软件、厂商提供的产品以及商业产品。

1. 捐赠的控制监视器

使用最广泛的控制监视器之一是计算机甲骨文和密码系统（Computer Oracle and Password System，COPS）。COPS是由Dan Farmer和Eugene H.Spafford开发的一套捐赠（自由）软件。该软件可以对UNIX系统上的安全脆弱性进行全面检查，其检查内容包括各种文件和目录的错误许可以及SUID/SGID检查。COPS提供源代码，并且它已经移植到了几乎每一种UNIX环境中。COPS当前没有GUI界面，它是由一个命令行shell程序调用的。报告可以直接打印出来，或者邮寄到系统管理员。因为COPS实际上是一个免费软件包，所以对它的支持是本地系统管理员的责任。

德州A&M大学的TAMU Tiger安全工具箱也是一套捐赠的软件包。该软件提供很多安全检查，其中包括对关键二进制程序进行各种加密校验和的测验。TAMU也检查安全补丁的正确应用和已知的安全暴露。

2. 厂商提供的产品

许多硬件厂商把控制监视工具作为其系统管理工具集的一部分来提供。这些监视器通常是由硬件厂商的支持组织支持的。一般来说，这些监视器在探查其硬件平台所特有的脆弱性方面非常有效。其负面作用是它们通常不能移植到其他UNIX实现上。

3. 商业控制监视

几个厂商也提供了商业的UNIX控制监视产品，这些产品相对于非商业的软件来说有一些先进之处。它们可以适用于很多混合的UNIX环境（如AIX、SUN/OS、Solaris以及HP-UX）。这些软件通常允许一个系统起到一个安全控制器的作用，负责监视一个安全域内的系统。它也促进同客户策略（如密码长度检查）的一致性。报告可以使用一个模板来生成，这样能消除无效的或者不适当的警告。商业UNIX控制监视产品通常都有一个容易使用的GUI界面。另一个重要的功能是它们可以为特殊计算环境定制安全性检查。同安全相关的报告可以很容易地定制。最后，软件支持是可用的。

商业控制监视解决方案包括BrainTree Technology的AuditorPlus以及Axent Technologies的Intruder Alert。系统管理解决方案提供商，如Computer Associates，也在其软件产品中提供了控制监视功能。

4. 商业访问控制解决方案

这类解决方案通过自己的认证、授权以及完整性控制机制补充了本地系统控制。这些方案不同于系统管理工具，这是因为它们的主要着眼点是安全性。一般情况下，它们不提供其他的系统管理功能。属于这类解决方案的商业产品包括Los Altos Technologies公司的Armor、瑞典斯德哥尔摩Dynamic Software AB的BoKs、Axtent Technologies公司的OmniGuard。这些类型的解决方案可用于UNIX、Novell Netware、Windows NT以及其他环境。它们一般提供如下优点：

- 集中式管理。
- 密码策略实施。
- 网络访问过滤。
- 基于时间的访问控制。
- 系统生成的密码。
- 并发访问限制。
- 针对具体设备的访问。
- 密码长度检查。
- 自审计机制。
- 集中式审计跟踪。

这类解决方案也为许多安全功能提供了集中式管理。例如，Armor能够起到为数百个UNIX工作站进行用户认证的单一认证源的作用。改进的密码管理和审计也是常见的功能。例如，BoKs和OmniGuard通过密码组成规则和密码时效性的使用增强了对健壮密码的需求。

12.1.1 系统管理

商业系统管理解决方案也可用于分布式UNIX环境。这些方案为分布式系统管理问题提供了一个全面的解决方法，这些问题包括备份和恢复、任务调度以及用户管理。许多同安全相关的功能，包括控制监视和完整性检查在内，系统管理方案也都提供了。商业系统管理解决方案包括Computer Associates的CA Unicenter、惠普的OpenView Systems Management、IBM的SystemView、Axtent Technologies的Ommi-Guard ESM以及Tivoli Systems的Tivoli Management Environment。将在第22章中深入探讨访问控制和系统管理解决方案。

12.1.2 审计跟踪

UNIX不使用一个单一的、综合的审计跟踪来记录系统行为。操作系统把审计数据记录在许多地方，其中每个都使用不同的格式和报告技术。这些独立的审计跟踪都被给与了不同的位置和名字。它们在各种UNIX平台上也不是标准的。表12-3总结了HP-UX操作系统上可用的主要审计跟踪。

表12-3 UNIX审计跟踪

审计跟踪	审计跟踪的种类
.history & .sh_history	用户命令行项
/var/adm/wtmp	成功的登录
/var/adm/btmp	失败的登录
/var/adm/syslog/*	网络连接日志文件
/.secure/etc/auditlogX(C2)	操作系统调用
/var/adm/acct	用户行为
/var/adm/sulog	改变到用户ID

商业访问控制和系统管理解决方案一般提供了一个集中式审计跟踪和高级报告功能。

12.1.3 动态警报

北美的许多公司为家庭用户提供了电子运动监测器。这些监测器能探查入侵者、听到警报，甚至能够拨叫本地警察局。表明它们出现在家庭中的滞销货的存在是足够防止许可犯罪行为的。通过把对网络管理的需要和安全性结合起来，也可以把一个类似的功能扩展到UNIX系统上。网络管理系统主要用于复杂计算机网络的管理。它们能够记录和处理警报，甚至达到为全体员工分页的程度。它们的使用可以扩展到安全领域中。我们将在第22章中介绍动态警报的使用。

12.1.4 安全警报

卡内基梅隆大学的计算机紧急反应组（Computer Emergency Response Team，CERT）和美国能源部的计算机事故报告能力（Computer Incident Advisory Capability，CIAC）跟踪各种操作系统的已知脆弱性。当安全性暴露已知的时候，他们提供安全警报和公告。报告被转发到说明安全事故的安全社区，并推荐为动作教程。

几个计算机厂商也提供了安全报告的公开通知。表12-4列出了部分关于安全报告、公告和支持的邮件地址。

表12-4 安全报告、公告和支持的邮件地址

厂 商	邮件地址
CERT	cert@cert.org
CIAC	ciac-listproc@llnl.gov
Cray Research	support@cray.com
Hewlett-Packard	security-alert@hp.com
IBM	services@austin.ibm.com
Next	ask_next@next.com
Santa Cruz Operation	security-alert@sco.com
Silicon Graphics	security-alet@sgi.com
Sun MicroSystems	security-alert@sun.com/var/adm/wtmp

HP安全公告

可以向support@support.mayfield.hp.com发送主题为security_info的电子邮件来订阅HP安全公告。如果想得到所有HP安全公告的索引，可以向上面地址发送主题为security_info_list的电子邮件，或者访问惠普公司的网站www.hp.com。有关安全问题，请发送邮件到security-alert@hp.com。

12.2 结论

有很多用于UNIX环境中的安全问题的解决方案，其中包括捐赠的和商业的解决方案，它们都为认证、访问控制、完整性检查以及控制环境的监视提供了高级工具。

前面我们对基本的UNIX控制结构、有关它们的实现问题以及网络功能进行了讨论，并了解了UNIX环境中的安全性问题。关于UNIX中的安全性问题，有三个关键点。

第一点，有很多方法都能破坏一个UNIX系统的完整性。若要正确地实现安全性，掌握有关UNIX工作原理的知识是非常重要的。安全和系统管理员都必须要深度掌握UNIX的工作机

制，并且能做到“温故而知新”。必须认识到，由于环境的复杂程度，手工检查控制中的脆弱性是不可能的。所以必须要使用一个自动工具来进行安全性检查。

第二点，总体可靠性绝对不能置于控制环境的持续功能性上。必须要认识到，当控制分布化的时候，它们会随着时间而恶化。对应用程序的改变、升级操作系统或者新系统管理员的行为都可能会影响到一个给定控制的持续功能性。对于那些负责UNIX系统的人，或者同此相关的任何计算机系统来说，他们必须要积极地检查现有控制并查找控制结构中的脆弱性。

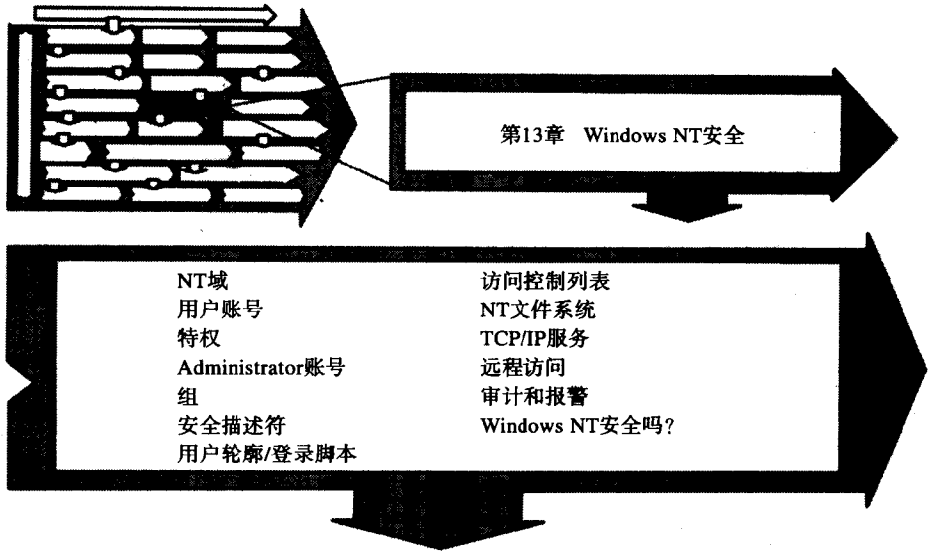
第三点，对积极检查UNIX环境以寻找安全性征兆的需要会改变。但是如果控制正确地实现了，并且定期评审，那么为什么不能简单地依靠这些控制呢？原因就是，同大型机环境不同，分布式环境中的控制通常是由本地管理员实施的。因为没有中心权力，所以分布式的控制在过一段时间后可能会削弱或者绕过。

上面几点为我们对集中式安全管理方案的使用和分布式计算中的审计任务所进行的最后讨论构成了基础。但是在能够完全信任环境之前，开放式系统的基础需要得到支持。另外的有关需求和领域包括：

- 对可用于很多应用的单一登录技术的需要。
- 客户机和服务器之间的相关安全认证，以及对在一个不可信传输上保护数据（包括密码在内）流量的需要。
- 对认证和授权特权的集中式安全管理的需要。

OSF的分布式计算环境（DCE）为开放式系统计算提供了一个坚固的基础。我们将在后面章节中探讨OSF的中间件方案。但是首先要探讨一下Windows NT——一个相对较新的操作系统——是如何解决计算安全性问题的。

第13章 Windows NT安全



Microsoft的Windows NT操作系统是在1993年首先进入市场的，现在它已经成为一个部署客户机-服务器应用的流行平台。Windows NT是基于32位体系结构的，它提供了许多高级功能，其中包括多任务、高可用性以及资源共享。Windows NT提供了一个容易使用的管理工具，并且对许多工业标准网络服务提供了支持。Windows NT也结合了动态主机配置程序（Dynamic Host Configuration Program，DHCP），DHCP允许网络地址动态分派给客户机。Windows NT也是使用Microsoft的SQL Server数据库引擎的流行平台。

从安全角度上看，相对于包括UNIX在内的一些较老的操作系统来说，Windows NT环境结合了许多高级安全功能。这些功能包括密码在网络上的保护、委托管理特权的能力以及对安全管理的一个全局方法。最近，Windows NT收到了美国国防部的国家计算机安全中心颁发的C2证书。Microsoft也为包括安全性在内的很多Windows NT功能发布了一套应用编程接口（API）。这些API允许第三方的和定制的应用程序结合进Windows NT环境中。

Windows NT分为客户机版和服务器版。两个版本在功能上基本类似，但是客户机版同服务器版比起来功能稍弱。例如，远程访问服务器现在支持最多256个同时连接，而客户机版只支持一个连接。尽管两个版本所提供的控制的核心功能是相同的，但是客户机系统不支持全局安全管理功能。因此，我们将主要讨论Windows NT操作系统的服务器版。

同前面几章一样，我们的目的是提供对Windows NT使用的安全机制的一个良好理解。在掌握了以后，将探讨一些突出的安全性问题。首先将对Windows NT安全性的整体特性进行一下探讨，下面先从介绍Windows NT域这个概念开始。

NT域

Windows NT安全性提供了管理用户账号和组的能力，并提供了从一个中心观点上实施安

全策略的能力。为了管理目的，Windows NT机器（包括客户机和服务器在内）可以合并到一个称做域的组内。一个域内的多个机器可以共享一个公共的用户账号数据库——其术语是全局账号域。这为用户提供了一个公共用户ID，而无需考虑他们登录的是域内的哪台机器。用户也可以指派到为整个域所定义的组中。

同通常的UNIX环境不同，安全标准可以实施到一个域内的所有系统上。在域中，有一台机器被定义成域的主要控制系统。从该系统上，可以建立公共的安全策略，并实施到整个域上。对这个系统的访问权通常（但并不总是）限定给域管理员。也可以指派用于域控制的备份机器，从而在主域不可用的情况下提供冗余性。

Windows NT也支持可信域的概念，它允许那些已经通过其自己主域认证的用户可以访问其他域中的服务——只要第二个域信任主域。若要这么做，二级域的域安全管理员必须同意把信任扩展到用户的域。Windows NT也可以扩展支持在二级域中没有一个实际账号的用户。这称做传递确认，它允许用户可以临时访问外部域而无需管理员的任何干预。需要注意的是，对一个域的访问权并不提供对任何资源的访问权。本地系统管理员完全控制着哪些资源允许来自本地域或者外部域中的用户访问。图13-1给出了Windows NT域模型的概图。

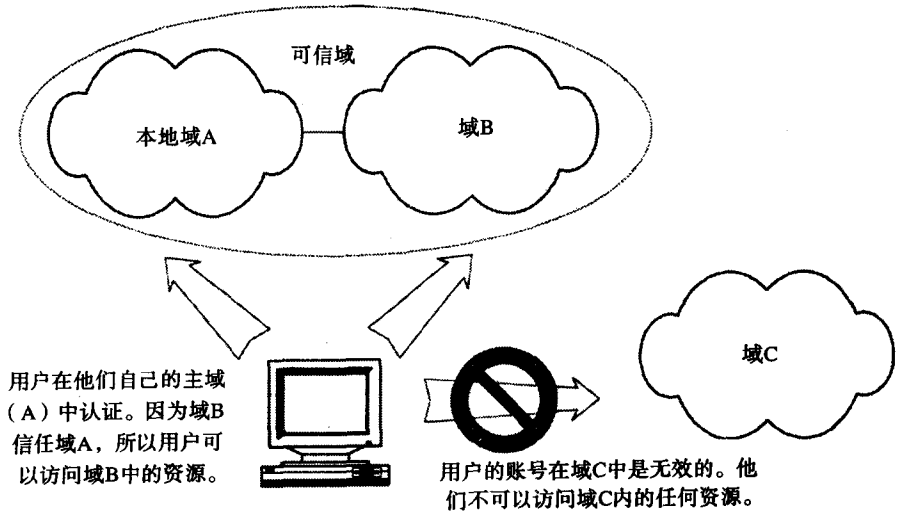


图13-1 Windows NT域模型

使用域模型主要有两个好处。第一个是安全标准（如密码时效、最小密码长度以及在一段无交互期后锁定账号）可以实施到域内的所有机器上。因为组织的安全策略集中实施了，所以由本地系统管理的不一致所带来的安全脆弱性得到了消除。这提高了域的整体安全性。但更重要的是，域为用户提供了一个单一参考点。对于辞职的或者被解雇的员工来说，系统可以在一个集中管理的地方取消他们的访问权。对于账号已取消的用户来说，除非他们知道GUEST账号，否则将拒绝他们对Windows NT资源的任何访问。

域模型的另一个重要方面是包括合作伙伴和合约人员在内的非员工可以很容易地标识。当一个非员工添加到域中时，他（她）可以指派到一个标识特殊状态的全局组中。本地管理员可以在非员工组指派的基础上明确接受或者拒绝对敏感或者机密信息的访问。

从前面可知，Windows NT允许域管理员为域内的用户建立和实施标准。这些标准主要实施到用户的账号上，其中包括要求密码长度和最小密码长度的能力。针对密码长度和组成，

公司标准也可以实施。在一段无交互期以后，或者在连续数次尝试登录失败以后，用户也可以连接断开。

用户账号

正常情况下，Windows NT用户是由安全管理员来指派账号的。用户账号可以在本地的基础上进行定义也可以在域的基础上进行定义。本地账号只能由本地系统确认，而域账号是域安全服务进行确认的。账号信息保存在一个安全账号数据库中，该数据库是由安全账号管理器（Security Account Manger，SAM）功能所管理的。SAM可以存在于一个本地系统上，也可以放在网络上从而为整个域提供安全信息。SAM的域版本可以复制到多个服务器上以提供冗余。

Windows NT的客户机和服务器系统都支持GUEST账号的使用。GUEST账号用来向临时用户提供访问权，这些用户在登录时使用GUEST做为账号名。在Windows NT客户系统上，GUEST账号是默认启用的，而服务器则恰好相反。如果启用了的话，GUEST账号应该通过一个密码来保护。聪明的管理员应该定期检查该账号的访问权，并保证尽可能限制它们。

特权

特权是一套允许一个用户或者组成员进行具体动作的权限的集合。特权可以直接由管理员指派给用户，也可以根据一个已定义组的关系来继承。当用户想访问某个对象的时候，就会检查用户的特权。这些特权包括访问网络上的Windows NT机器、进行备份以及管理审计系统的权力。表13-1列出了一些可以指派给用户或组的可能的特权。

表13-1 NT特权示例列表

特 权	说 明
SE-SYSTEMTIME_NAME	重设系统时钟
SE_RESTORE_NAME	从备份中提取文件
SE_DEBUG_NAME	进行系统调试
SE_SHUTDOWN_PRIVILEGE	关闭和重启系统
SE_TAKE_OWNERSHIP	传递对象的拥有关系

Windows NT中有很多可用的特权。表13-1只提供了其中的少数几个示例。需要认识到，Windows NT安全机制是在两个因素的基础上决定是否允许一个用户进行某个动作，它们是用户所拥有的特权和用户希望访问的对象的访问控制。即使用户给与了访问网络上Windows NT服务器的特权，他们可能也会被拒绝访问该服务器上的任何或者全部资源。特权也可以为管理整个Windows NT域而定义。这些全局特权通常同一个域用户管理器相关。

Administrator账号

Windows NT中的Administrator账号同UNIX超级用户有点相似，它具有对Windows NT系统的完全控制权。二者之间的明显不同是Windows NT管理员已经赋予了能力，这是因为它们已经指派了用户权限的全部集合。如果这些权限删除和指派给其他用户，这种能力就会消除。在UNIX上，如果不使用第三方产品或者修改操作系统的话，这是不可能做到的。

组

同我们已经介绍过的其他操作系统一样，Windows NT也支持组的使用。使用一个称做组的复合实体，访问权限和特权可以指派给多个用户。Windows NT为本地机器和域预定义了很多默认组，其中包括管理员、访客以及用户组。系统管理员可以按照意愿来建立和维护新组，他们可以在部门结构、工作职能或者其他所需的链接用户的原因基础上创建新组。创建组的

主要原因是为了简化管理。从安全角度上看，指派到组中的用户继承了同该组相关的权限和特权。

Windows NT支持两种类型的组，本地组和全局组。本地组只在本地系统上是有效的，它不能授予其他NT系统上的特权。然而，它可以包含由已经全局定义的域或组所定义的用户账号。全局组在整个Windows NT域（或者其他可信域）上都可用。全局组不能包含本地组，也不能包含其他全局组。本地组和全局组之间的最大区别是本地组中的成员只在本地系统上是有用的，而全局组中的成员可以在许多不同的系统上授予访问权。同其他操作系统中的组的使用一样，控制和监视组访问特权和组成员是非常重要的。

安全描述符

Windows NT允许访问控制置于很多系统和域资源上。保护资源的类型包括：文件、目录、打印机、程序以及网络连接。实现资源安全性的方法是，在一个称做安全描述符的数据结构中指定特殊的安全性信息，然后把该安全描述符同需要保护的资源或对象关联起来。安全描述符包含拥有者的名字和用于对象的相关组名，并且也保持了可以访问该对象的用户列表、如何访问该对象以及用于该对象的审计需求。用户和组的列表加上相关的访问许可称做访问控制列表。

13.1 安全控制

成功登录到Windows NT环境上的用户会收到一个访问令牌。系统授予用户对一个对象的访问权所根据的不是用户ID，而是这个访问令牌。访问令牌包含三项关键内容，用户ID、用户所属的组以及已经授予用户的特权。用户所运行的每个程序或进程都给了访问令牌的一份拷贝。

通过使用用户ID（如账号名）和相关密码的结合体进行登录，就认证了用户，然后得到访问令牌。用户ID是用来标识用户账号的。用户ID最多可以包含20个字符，但是大小写无所谓。在Windwos NT环境中，用户ID glenbruce和GLENBRUCE是一样的。密码最多可以包含14个字符，但是同用户ID不一样，密码是区分大小写的。这种异常曾让许多用户受挫，包括作者在内。登录过程本身是通过强迫用户输入ALT+CONTROL+DELETE组合键进行保护的，如果在登录程序之外使用该组合键，这会导致一个Windows工作站进入重启画面。这种方法能够阻止特洛伊木马程序，但是有人说基于DOS的特洛伊木马程序能够绕过这种控制。

图13-2提供了Windows NT认证和授权机制的一个概图。

本地安全管理器（Local Security Authority，LSA）是一个运行在本地计算机上的安全子系统，它负责标识用户、创建访问令牌和维护审计跟踪。登录进程激活的是LSA。然后LSA联系安全注册表（如SAM），并由SAM校验用户账号和密码。SAM也向LSA提供关于用户组成员关系和特殊特权的信息。一旦SAM确认了用户的标识，LSA就会为用户创建一个生命期受限的访问令牌。实际上，访问令牌只在用户会话期内有效，它会因为用户无交互或者时间限制而自动消除。SAM和LSA之间的所有通信都使用LSA和SAM共享的密钥进行加密。

接着，访问令牌传递到监视和控制Windows NT进程的Win32S子系统。Win32S接着为用户开始一个初始的进程。Windows程序管理器启动，它显示用户的初始图标和桌面。用户启动的每一个后续进程都有与其相关的访问令牌。当用户请求访问一个对象时，这些访问令牌（不是账号名，如用户ID）就会检查。负责检查的是第二个安全功能，安全参考监视器

（Security Reference Monitor，SRM），它来确定是否授予对Windows NT资源的访问权。SRM对访问令牌和与对象关联的安全描述符进行比较。如果安全描述符所请求的要求满足，那么就会授予对该对象的访问权。

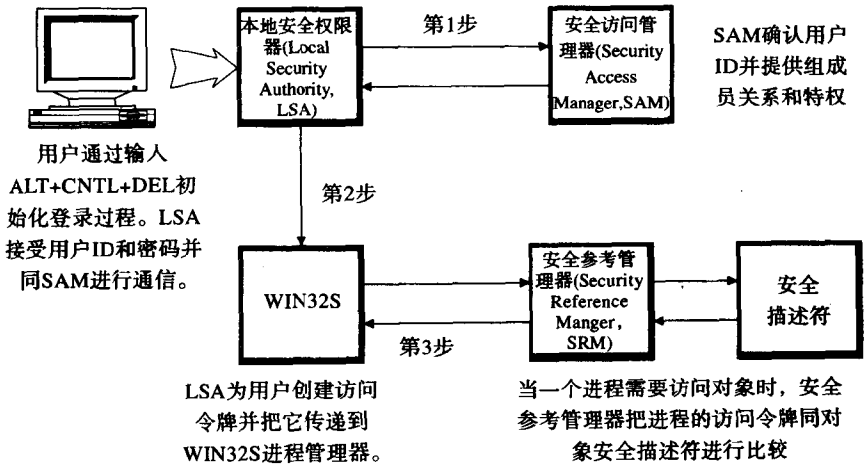


图13-2 认证和授权

Windows NT所使用的认证和授权方法有几个优点。首先，密码在LAN上是以加密的形式而不是以明文的形式进行传输，因而它不能直接查看到。第二，认证和授权进程之间有一个非常牢固的连接。因为访问一个对象需要一个访问令牌，所以入侵者不能简单地使用一个假的用户ID来提交一个系统调用。通过检查用户的特权和标识，访问控制也提高了安全性。

13.1.1 用户配置文件/登录脚本

用户的桌面环境可以在用户配置文件（profile）中进行定义。用户配置文件用来设置默认打印机和显示公共屏幕布局和颜色。它们对于安全最重要的方面是能够防止用户改变环境。也可以用它们来为用户指派默认HOME目录（该目录可以用来保护建立的文件，或者限制用户可以访问的程序和信息）。

用户配置文件可以指派到独立用户，也可以指派到组。用户配置文件的一个重要限制是它们仅能为Windows NT客户工作，而不能用在Windows 95或者Windows for WorkGroup客户机上。Windows NT也支持登录脚本——在用户登录后被自动执行的批处理文件。尽管没有用户配置文件全面，但脚本可以被所有的Windows客户机使用。登录脚本可以用来启动更进一步的安全性检查、控制环境设置，或者初始化同其他机器的网络连接。

13.1.2 访问控制列表

Windows NT使用访问控制列表来限制用户对资源的访问。ACL主要同Windows NT文件系统一起工作，它可以限制用户访问、维护或者删除资源的能力。访问控制列表实际上由很多条项组成，这些项称做访问控制项（Access Control Entries，ACE）。每个ACE包括了有关对象如何被特定请求者访问以及需要进行什么审计的信息。对于一个文件，一个ACE可以向某个特定用户授予读写特权。还可以建立第二个ACE来允许一组用户只能读取该文件，等等。

ACE项所授予的权限也称做是访问掩码。受到控制的访问类型取决于ACL所控制的对象类型。表13-2列出了一般的用于文件的访问权限，它们可以指派到用户或者组。

表13-2 用于文件的访问控制权限

访问权限	说 明
READ	进行用户管理、重新配置服务器以及管理访问控制
WRITE	允许用户修改文件的内容
DELETE	允许用户删除文件。在一个NTFS文件系统上，被删除的文件不能使用磁盘工具来恢复
CHANGE PERMISSION	允许拥有这种特权的用户改变文件的许可
EXECUTE	执行文件——如果文件是一个程序
TAKE OWNERSHIP	把一个文件的所有者关系改变到另一个用户
FULL OWNERSHIP	用户拥有上面所有的访问特权

Windows NT为NTFS目录使用了独立的访问控制权限，如表13-3所示。

表13-3 用于目录的访问控制权限

访问权限	说 明
NO ACCESS	无论如何用户都不允许访问该目录
LIST	可以列举该目录下所包含的文件或者子目录
ADD	用户可以建立新文件或者新子目录
READ	用户可以显示关于文件或者目录的信息
ADD&READ	用户可以建立新目录，显示谁拥有目录以及目录的相关访问权限
CHANGE	可以删除目录
FULL CONTROL	用户可以拥有上面全部访问权限

当把用于文件的和用于目录的访问权限结合起来使用时，管理员可以定义一套非常有限制性的访问权限。同前一章所介绍的UNIX文件和目录许可比较起来，Windows NT的实现具有一些优点。首先，许多UNIX系统不支持灵活的ACL，它们只能根据用户的分类、组以及其他因素来指派文件或者目录许可。UNIX用来保护文件和目录的许可也只限三种，READ、WRITE和EXECUTE。对于那些支持在目录上使用ACL的UNIX系统来说，目录的ACL一般是不可继承的，即不能传递到子目录上。更值得注意的是，UNIX只能把这些许可应用到文件和目录上，它不支持对大范围的系统资源使用访问权限控制，而Windows NT可以做到。实际对比结果是UNIX资源访问控制同Windows NT的比起来：全面性不够、粒度不够，并且更难以管理。

13.1.3 NT文件系统

Microsoft为Windows NT创建了一个新文件系统，称做NT文件系统（NTFS）。设计NTFS的目的是为了解决Microsoft使用的传统文件系统中的一些问题。Microsoft在NTFS中添加了从系统或者电源故障中进行恢复而不会导致破坏或者损坏磁盘驱动器的能力。NTFS有一些保证其数据的一致性和可恢复性的高级功能。并且，NTFS支持大文件和大磁盘。对数据冗余的支持和用于快速文件查找的一个改进的索引方案也实现了。但相对于Microsoft所使用的传统文件系统来说，NTFS中最重要的改进可能是它的安全性。

NTFS使用一个非常类似于UNIX的方法来保护它的数据和目录。NTFS文件系统中的每个对象，包括文件或目录，都有与其相关的安全信息。该信息存放在安全描述符里面。对NTFS中一个对象的每次访问请求都会引发一次安全性测试。用户（或程序）必须提供一个安全访问令牌，该令牌证明他们是有效的用户。用户的标识和访问令牌所持有的特权，同安全描述符中的访问许可进行比较。如果访问令牌所表示的授权同安全描述符的需要相一致，那么就会授予对对象的访问权。如果不一致，那么用户的对象访问请求就会失败。

同传统的UNIX文件系统比起来，NTFS文件系统在安全性方面有几个优点。首先，前面已经提到过，Windows NT支持很多访问许可。这使得系统管理员可以为NTFS对象使用比UNIX更严格的安全访问设置。UNIX支持传统的RWX（读、写和执行），这些许可应用到文件所属的用户、组成员以及所有其他用户上。Windows NT定义了7个许可和多个访问控制列表。另一个优点是审计已经设计进NTFS的基本功能里了。但最重要的优点是NTFS评价用户请求所基于的是一个访问令牌而不是用户ID。在UNIX中，伪造一个用户ID（在前面关于UNIX的几章中我们已经给出了几个例子）然后向文件系统提交未经授权的请求是相对很容易的。Windows NT所使用的访问令牌是很难伪造的。它们也包含另外的安全信息，如用户特权，这使得管理员在是否授予访问权这个问题上可以做出更好的决策。

删除的对象不可重用是C2安全性的一个要求。这一规则还适用于内存的使用。当一个进程在内存中结束了的时候，其内存区域就被清洗，并且不能被其他用户检查。Windows NTFS文件系统不能恢复删除，但在DOS文件系统下可以。

尽管NTFS支持一个先进的安全模型，但Windows NT系统也存在着一些缺点。Windows NT系统可以支持很多不同的文件系统，其中包括MS-DOS系统下的传统文件分配表（File Allocation Table，FAT）文件系统。FAT文件系统相对来说安全性很差。一个Windows NT系统所使用的FAT或者其他类型的文件系统一般不支持前面所介绍的NTFS安全功能。这些文件系统所保存的数据或者信息很容易破坏。

13.2 连网

13.2.1 TCP/IP服务

Windows NT对UNIX、Novell Netware以及因特网环境中的许多TCP/IP连网服务提供支持，其中包括对通信、打印、管理协议（例如SNMP）以及包括telnet和FTP在内的传统UNIX通信服务的支持。另外，Windows NT还提供了对因特网 gopher、域名服务以及WWW服务的支持。将在第11章和第14章中介绍有关使用这些服务的安全性问题。一个重要的事实是大多数TCP/IP服务都以明文的形式发送密码。必须要认识到，当使用了TCP/IP服务的时候，这个问题以及其他问题在Windows NT环境中也是存在的。

13.2.2 远程访问

Windows NT为希望连接到Windows NT环境的远程用户提供了支持。远程访问服务（Remote Access Service，RAS）提供了很多不同的远程连接，其中包括通过电话线的modem访问和X.25访问。串行线路因特网协议（Serial Line Internet Protocol，SLIP）和点到点协议（Point-to-Point Procotol，PPP）支持modem访问。用户通过了RAS服务器的认证以后，就可

以完全访问Windows NT服务，就好像他们通过LAN连接着系统一样。一个RAS连接可以使用很多LAN协议，这包括TCP/IP、IPX以及NETBEUI。

内建的安全机制是RAS服务的一个强大特性。同Windows NT操作系统的许多功能一样，RAS支持很多选项，为计算环境选择最合适的选项是系统管理员的责任。用户的认证可以通过使用PPP挑战握手认证协议（PPP Challenge Handshake Authentication，CHAP）或者密码认证协议（Password Authentication Protocol，PAP）来提供。这两个协议都可以保护密码，它们不使用明文形式在线路上传输密码。二者之中，CHAP使用了一个强大的认证机制，它是首选的。应该指出，CHAP是一个PPP机制，它在SLIP上不受支持。它可以同DES或者MD5加密一起使用。

RAS接收到用户ID和密码以后，用户就会被Windows NT安全服务以正常的方式进行认证。即使客户对于本地域来说是未知的，他们也可以登录——如果他们是一个可信域中的成员。RAS的一个更好的功能是安全管理员可以确定一个RAS用户的活动范围。管理员可以决定独立RAS用户是否应该对完整Windows NT域具有访问权，或者限制到那些由本地RAS服务器提供的服务。Windows NT支持回拨控制——它要求RAS中断认证过程、断开用户、并对一个预定义的电话号码进行回拨。也支持所有流量的数据加密。RAS使用RC4数据加密算法，它可以以一个安全的方式同一个客户机处理RC4能力（如Windows NT客户）进行通信。

Microsoft也提供了很多应用编程接口（API），这些API使得第三方产品可以结合进RAS中。例如Security Dynamics的一次密码SecureID产品可以集成进RAS中。Windows NT操作系统的C2类审计功能已经扩展到RAS。很多事件的成功或失败，包括连接、由于认证失败或者无交互导致的断开连接以及协议失败都可以审计。定义适当的审计级别是RAS管理员的责任。

13.2.3 审计和报警

Windows NT操作系统为C2类审计（由美国国防部桔皮书定义）提供了支持。C2规定了审计跟踪，它可以对很多事件的成功或失败进行日志纪录。这些事件可以包括用户的登录或注销、文件或资源访问、对安全环境的改变或者安全策略改变。在Windows NT系统上可以维护很多审计跟踪，它们包括同操作系统活动相关的事件、对安全环境的改变以及同应用程序相关的信息。需要注意到，同大多数审计跟踪一样，Windows NT审计跟踪也存在于本地系统上。如果保护审计跟踪的许可是松散的，或者一个用户可以得到ADMINISTRATOR特权，那么本地审计跟踪就可以篡改，从而破坏其完整性。

对其他安全机制的支持

Windows NT提供了很多应用编程接口，它们允许定制的应用程序和第三方产品集成进环境中。然而，这些API是专有的，它们不是基于工业标准的。Windows NT为OSF/DCE环境提供了有限的、但非全部的支持。尽管Microsoft提供的远程过程调用同OSF规范是兼容的，但是它们并不是直接从OSF规范创建的。并且，Windows NT不支持全部的DCE服务。将在第16章和第17章介绍DCE服务。

13.2.4 Windows NT安全吗

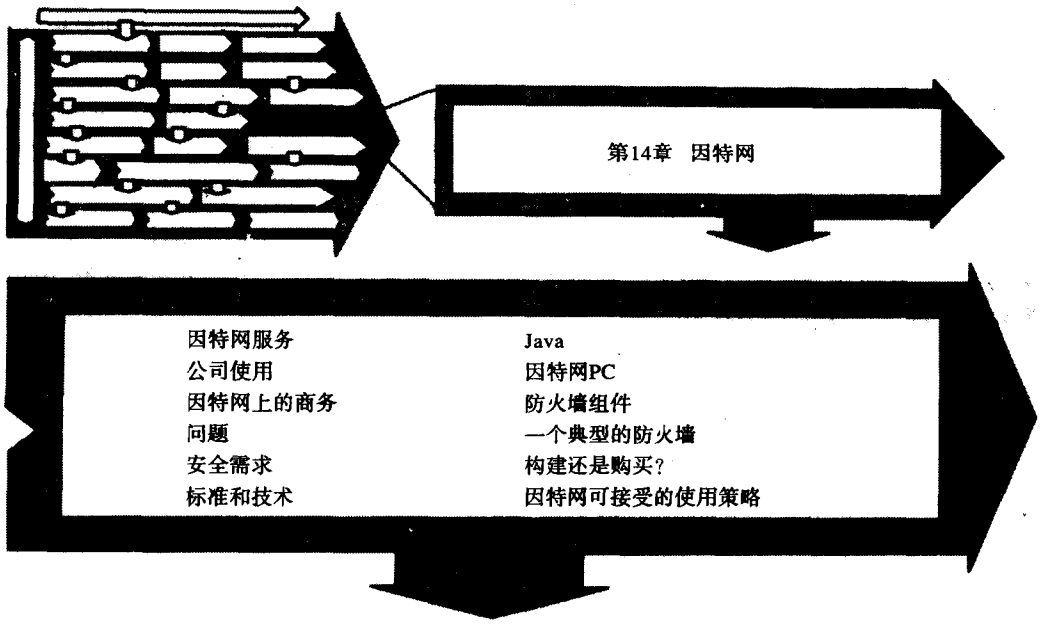
在Windows NT的安全功能设计中，Microsoft有能力理解许多老式的操作系统所面对的安全问题。通过理解这些问题，Microsoft能够解决Windows NT的设计中的很多问题。这也说明，

一个实现拙劣的Windows NT服务器同任何其他类型的服务器一样易受攻击。不安全的GUEST账号的使用、不正确的访问特权和松散的访问控制许可会导致一个很容易受攻击的计算机系统。第三方已经认识到了这个问题，并正在提供解决方案。例如，Intrusion Detection公司的Kane Security Analyst能够分析和报告Windows NT系统的整体安全性。密码猜测和特洛伊木马攻击仍旧威胁着Windows NT环境。Windows NT上的TCP/IP服务，包括telnet和FTP在内，同其在其他平台上一样容易遭到密码探查攻击。

13.3 结论

Windows NT环境专门设计来克服其他计算机平台上特别是UNIX系统上的许多普遍的安全问题。Windows NT达到了C2安全级别，但是许多其他的计算环境也达到了或者支持该标准。对于Windows NT的整体安全性来说，一个重要的因素是该操作系统的源代码仍然受到Microsoft的严格保护。但是这不会阻止有兴趣的人们去发现缺陷或者bug并利用它们。虽然Microsoft已经提出了一些标准，但是Windows NT无疑并不全部符合它们。Windows NT控制的专有性是个问题。关于Windows NT的一个关键问题是新兴的安全标准是否能轻松地结合进该产品中。如果能，那么Windows NT应该能够保留它的作为一个安全的和开放的计算平台的声明；如果不能，Windows NT可能会成为另一个孤立的操作系统，它会复杂化而不是帮助解决分布式计算中的安全性挑战。

第14章 因 特 网



因特网在许多方面都类似于蛮荒之境。基本上，它是一片没有法律的地方，所有的只是用户间的互相约束。因为没有法律没有看守，所以安全暴露的可能性是非常大的。为得到连接的大量可用信息和狂热成了得到窃听的强制理由。当连接到一个像因特网这样的不可信网络时，所涉及到的具体安全问题，必须要实现。

许多组织或者已经连上因特网了，或者正计划准备连上因特网。据报道说现在有超过2千万到3千万的人已经上网了，但是估计数据还会大幅变化。因特网已经成为世界的商业管道，但不幸的是，它也成了非法入侵计算机系统和网络的一个焦点。对于因特网的使用，组织有很多问题需要回答。

一个组织如何能够对公司内部网和因特网之间的访问做到最佳控制呢？哪类公司信息允许流向因特网呢？业务伙伴、客户与或者公众允许访问内部网络上的应用程序和系统吗？如果可以，那么应该对他们的访问进行什么限制？中心问题是一个组织如何能够有效地利用因特网开展业务。这个问题涉及到很多方面，但其中必须要解决的关键是安全问题。

在本章中，将讨论一些使用因特网的业务问题，介绍一些使用这种不可信网络的具体风险，并提供一些可以用来解决这些问题的方法。

14.1 因特网的概念

首先我们来解释一下什么是因特网，并说明一些涉及到的术语和技术。因特网最初并不打算设计成今天这么流行的世界网络。因特网的开始是在1969年，当时美国国防部委托高级

研究计划署（Advanced Research Projects Agency，ARPA）进行网络协议研究。1973年，传输控制协议（Transmission Control Protocol，TCP）/因特网协议（Internet Protocol，IP）作为一项用于计算机通信的标准网络协议提出。1983年，加州大学伯克利分校的UNIX操作系统发布版本包括了该网络协议，TCP/IP开始迅速流行。因为ARPA在其网络中所使用的许多计算机都使用UNIX，所以TCP/IP成了事实上的标准。

美国国家科学基金会（U.S. National Science Foundation，NSF）在1986年开始涉足网络领域，并为超级计算中心之间的快速互连提供了资金。这为网络建立起了一个高速主干。因特网开始演变成一个连接大学、研究中心以及公司的共享网络。截止到1995年，因特网已经快速增长成一个包含几百万台机器和数千万用户的网络。因特网的控制已经移到私人兴趣上。因特网已经从一个政府资助的小型研究项目长成世界范围内的计算机通信网络。因特网的兴趣和增长是不可阻挡的。

术语因特网用来描述公共网络通信，实际上它是由多个使用TCP/IP协议的服务组成的。TCP/IP网络协议实际上是由多个用于不同目的的协议组成的集合。TCP/IP的某些性质使得它对军事应用非常有吸引力。TCP/IP是基于动态包路由协议的。一个大型TCP/IP网络的主段可以破坏或者禁用而不会破坏仍旧在进行通信的网络节点的能力。在第7章中，我们已经看到了使用TCP/IP协议来建立通信是非常容易的。在主要用于学术研究目的基于TCP/IP的网络中，安全性从来不是一个重要的设计考虑。事实是这种连网能力是普遍的和强大的，但它也会带来安全问题。

14.1.1 因特网服务

因特网主要是因为一些利用网络的服务而出名。最著名的服务是万维网（World Wide Web，WWW），它是最近流行的服务之一。做为把散布在几台计算机上的技术文档组织起来的一个方法，WWW技术在1989年开发出来。使用超文本标记语言（Hypertext Markup Language，HTML），这些文档链接起来（HTML提供了从一个文档链到另一个文档的指针）。基于HTML的浏览器软件的开发使得人们可以通过引用这些指针来访问和显示文档。沿着嵌入的链接，基于HTML文档的信息可以根据许多路径来引用。浏览器软件的发展为用户带来了图形、声音、甚至动画。大多数近来的商业因特网入口开始使用WWW技术来使得信息可用。初始的业务因特网是建立HTML网页以使得可以公开访问公司信息。

文档是使用超文本传输协议（Hypertext Transfer Protocol，HTTP）而获得的。HTTP是在网页和浏览器之间传递信息所需要的协议。它使用统一资源定位符（Uniform Resource Locators，URL）来定位HTML文档。URL定义了一个引用具体文档、图像或者其他Web资源的统一地址。HTML文档可以把指向其他HTML文档的URL内嵌进文档中。HTML定义了链接的格式，而HTTP定义了如何从Web服务器提取本地Web文档。

可以把URL想像成一个用来指引图书馆中一本书位于什么地方的索引卡。我们可以使用这个卡来定位（HTML）文档的物理位置。如果有一本书包含了更多的索引卡，这些卡片指出了另外的相关信息的位置，那么会如何呢？这些另外的信息其形式可以是另一本书、一幅图画或者一盒磁带。我们可以沿着这些URL而得到这些信息，而不总是回到主图书馆索引。如果这些信息分布在不同城市的几个不同馆中呢？HTTP可以跟踪信息线索而无需考虑它在哪里，并以一个可以理解的格式把它提供给我们。

其他使用因特网的流行服务还包括文件传输协议（File Transfer Protocol，FTP）、域名系统（Domain Name System，DNS）以及电子邮件。FTP是用来把文件从一台主机计算机传输到另一台机器的协议。当前的WWW浏览器也支持FTP服务。DNS允许客户机确定，或者说解析，一台服务器的IP地址——使用DNS，同服务器联系可以只使用它的因特网名。电子邮件是因特网上另一个可用的常见服务。把电子邮件地址印到名片上已经成了很平常的事儿，电子邮件地址已经变得同通常业务地址、电话以及传真号码一样重要。因特网上还有其他一些可用的服务，限于篇幅关系，这里不做介绍了。

有关管理因特网服务的更详细信息，我向读者推荐两本非常流行的书，Cricket Liu：Liu，Peek，Jones，Buus以及Nye[1994]和Liu、Albitz[1992]。

14.1.2 公司使用

对于想提供和访问信息的公司来说，因特网已经成为一个非常有趣和非常有吸引力的媒介。WWW提供信息的能力犹如为组织打开了一道研究如何才能利用因特网的大门。大量的宣传和支持因特网使用的技术产品和服务的出现推动着这一切。关于因特网的使用，公司应该注意两个基本问题，公司如何把因特网应用到自己的业务上？公司如何使用因特网来赚钱？第二个问题并不总是随着第一个问题。尽管因特网为业务增长表现出了一个有吸引力的前景，但是它也带来了新的挑战、机会和考虑。

因特网的主要好处之一是访问大量在线信息的能力。因特网能够提供其他地方没有的多种类型的交互式信息。公司可以利用这一点来提供关于其产品和服务的多种信息。这可以包括文本、图形、声音记录、甚至视频。当前的挑战之一是知道你正在向谁提供信息。你不会总是知道谁正在访问你提供的信息或者他们如何使用这些信息。

在因特网上，提供者和信息及服务的使用者之间的关系是多对多的和任意对任意的关系。因特网常常说成提供A4服务：访问（access）、任意时间（anytime）、任意地点（anywhere）和对任何人（anybody）。在很多方面，如何使用因特网的决定是由用户群体做的。用户在指示提供者提供什么信息或者如何提供信息。例如，业务应该支持Web浏览器的功能而不是指示人们应该如何访问信息。

根据组织具有的或者希望同用户保持的关系类型，因特网的业务使用可以分成几类。这些类别说明了信息提供者和预期访问者之间的关系。提供给用户的信息或应用程序的类型同它是如何提供的是不同的，这取决于预期的关系。对于因特网关系来说，一个基本问题是是否需要单独标识访问信息的人。

1. 公司信息

第一类一般说明了大多数组织对因特网的初次使用。组织使用WWW服务器来提供有关公司和公司产品及服务的信息。这类使用可能包括对普通公司信息、办公位置、如年度或者季度报告之类的官方公司信息以及产品说明和可用性信息的访问。使用因特网来提供这类信息要比使用使用传统的打印和散发更迅速更便宜。对于深入的产品技术信息来说，使用因特网来提供要比使用传统的产品手册更加详细。公司网页也可以用来提供相关的业界信息，并且可以链接到其他对用户有价值的因特网地点上。但在使用因特网来发布特定类型的财政数据之前，公司需要考虑一些问题。例如在某些国家里，在线提供财政计划书可能会违反法律。

2. 公司对新客户

有些公司扩大了他们的因特网使用范围，他们在网上提供产品分类，并允许客户通过因特网来提交购买那些产品的订单。业务关系现在已经扩展到了标识的个体上，而不仅仅是组织的已有客户。在线产品分类和产品订单对于服装店、书店或者其他传统邮购业务来说是非常流行的。把产品分类提供给公众的能力在公司的当前邮件列表之外提高了销量。如果要有效地开展这种类型的业务操作，需求之一是建立一个配送网络来处理这些通过网络收集的订单。如果要把业务推到因特网上，同样需要考虑一些问题，如税收和出口控制等。

另一个流行的地方是提供客户支持的能力。几家公司为他们的产品提供具体的客户支持服务。这种支持服务可以在很大程度上自动化。软件可以用来提供自动支持信息以回答客户的问题。另外，软件可用性和发布也是因特网的一个使用领域。几个计算机硬件和软件公司通过因特网来提供如驱动程序或者补丁和更新包之类的软件。

3. 公司对现有客户

这种类型的因特网关系扩展到了公司的现有客户上。因特网可以为现有客户提供一个对公司系统的替代访问方法。这种访问可以扩展到个人财政账号和事务执行上。几个银行提供了对客户账号信息的访问，并且一些银行已经提供了对其客户提交银行事务的能力。另一个例子是得到依靠时间的客户具体信息的能力。例如，一个信使公司可以向客户提供跟踪其包裹传送过程的能力。

4. 公司对公司

因特网正在用做提供公司到公司通信（如加入伙伴工程或者支持业界组织）的通信机制。许多组织为了有关其工作的信息而维护网站。如对象管理组（Object Management Group，OMG）之类的标准体使用WWW来提供有关标准发布文档、标准过程进度以及会议备忘等信息。这些信息是为标准体的参与者或者其他感兴趣的人而提供的。

因特网也用来支持一些经过选择的技术的联合开发工作，这甚至是在竞争对手之间进行。例如，一个大规模的开发工程可能需要几家转包商的参与，其中每家都提供原料或者组件。因特网可以用来共享工程信息并提供一个消息通信途径。使用电子数据交换（Electronic Data Interchange，EDI）来交换商业文档也是因特网使用的应用领域。

5. 公司内部“Intranet”

公司可以利用因特网技术的优点，但不连接到外部因特网上。企业内部网（Intranet）是用来描述因特网技术的内部使用的术语。许多公司已经把因特网WWW技术建立成提供和维护内部信息以供自己员工使用的一个标准方法。这种使用的例子包括访问公司每日新闻、内部电话目录、包括津贴程序和工作应聘在内的人力资源信息、专门设计的内部支持应用程序、技术信息、内部软件可用性和发布、部门通信（内部网页）以及到其他分公司或者部门主页的链接。同传统的纸张文档比起来，使用这种方法来提供和访问信息开销更低，这是这种方法的一个主要好处。

14.1.3 因特网上的业务

对于因特网的业务使用来说，一个很自然的扩展是把这种不可信网络上的商业事务包括进来。这是一个非常激动人心的前景，但是它有很多的安全问题。当我们谈论网上商务时，一般指的是财务或者同财务相关的事务的传输。商业应用要求事务双方能够互相认证并能够

机密地进行业务事务。对于这些商务事务来说，一个更重要的安全元素可能是认可。我们需要确信这些事务是真实的和有效的。事实上，当使用Web浏览器和其他因特网业务时客户的安全性必须同与自动柜员机或者其他银行机制打交道时的安全性相似。

当使用因特网来传输那些使用已建立的付款方法（如使用信用卡或者支出卡）的事务时，一般需要涉及到5个方面，信用卡/支出卡的发行者、出售商品的销售商、购买商品的顾客、事务的捕获者（例如销售商使用的银行）以及一些安全管理局使用的保证事务安全的方法。因特网服务需要同一性和机密性，这些事务类型中快速增长的利益促使有能够满足这种需要的方法。公共安全管理局的可用性有助于促进因特网上的商业事务。

14.1.4 问题

当计算机黑客使用因特网进行非法系统访问时，因特网的安全问题就引起了广泛的注意。1995年1月，一个叫做Kevin Mitnick的计算机黑客（也称做“Condor”）在卡来罗纳州逮捕了。他被怀疑非法使用蜂窝电话系统获得因特网的访问并为其非法入侵许多计算机系统提供机会。他是在一个非常著名的计算机安全研究人员Tsutomu Shimomura的辅助下逮捕的。使用他的知识和特殊工具，Shimomura帮助安全局跟踪Condor。据说Condor访问了Shimomura自己的系统并偷取了有价值的信息。研究人员通过监视Condor的行为，然后在电话技术员的辅助下，并同FBI一起工作，他们能够跟踪Condor到一个特定的房间。因特网最有趣的方面之一是没有任何安全中心来拥有、管理、控制或者维治因特网。这常常是因特网最令人费解的一方面。当出现了安全问题或者安全情况时，问题“为什么某人不做某事呢？”就会经常提道。虽然计算机紧急响应组（Computer Emergency Response Team，CERT）和事故响应论坛和安全组（Forum of Incident Response and Security Teams，FIRST）在出现安全事故的时候可用于因特网用户，但是没有任何可以求助的管理局。这种中心控制或者管理中心的缺乏带来了一些额外的挑战，这使得许多业务难以开展。

对网页的链接和网页的可用性都是很脆弱的。如果他们变得不可用，那么哭天喊地根本没有用。改变管理是不存在的。WWW链接和网页可以在没有任何警告的情况下出现或者消失。使用因特网发送的消息是否会传送是根本没有保证的。除非实现了特殊的机制来提供一个确认，否则根本没有任何方法来知道一条消息是传送成功还是失败。

14.1.5 安全需求

把网络连到因特网上跟不在家时把门敞开是非常像的。可以邀请任何人来欣赏你们家卧室中墙上的艺术品，或者浏览书房中的书。可以锁上其他房间的门以防止别人进入。甚至可以把进入上锁房间的钥匙拿给那些经过挑选的人。如果有个人拾到了其中一个钥匙，接着进入房间，打了个转，然后再重新锁上门，那么你如何知道呢？你如何才能证明只有具有钥匙的人才进入过房间？除非有一个来客簿，否则你不会知道谁曾来过你那敞着门的家。我确信，如果你曾经把家门敞开过，那么肯定会施以严格的控制以保证你不在家时的安全。

当使用因特网但却有一些新方法的时候，安全性就是必须的了，这包括认证、机密性和授权。当使用一个不可信网络时，对认证、机密性和认可的需要变得更重要了。安全性在网络消息的内容和传输两个方面都是必需的。因特网就如同一个黑色幽径，在那里，你不知道谁正潜伏在阴影里。如果消息或者数据没有通过一个机密机制（如加密）来予以保护的话，

那么你不会知道谁能看到你在网上所发的东西。

使用因特网的另一个问题是一个人或者东西可能不是它所表现的那个人或者那个东西。在第7章中，我们已经看到了操纵TCP/IP包信息是多么的容易。并且特别提到了在欺骗的攻击中修改包地址。欺骗已经用来入侵安全系统，其所凭借的是模仿一个目标系统所信任的系统的能力。邮件发送者的标识可以被欺骗，并且生成邮件的地点也难逃厄运。如果没有严格的认证和认可，那么你不可能证明信息的真实性或者个体的同一性。

使用数字签名来验证标识并保留消息的完整性是解决这个问题的一个方法。管理和校验这些签名的标准方法是必需的。如果你不熟悉某个人的签名，不知道它是不是可信的，那么它不会给你带来什么好处。提供签名的认证需要使用公共证明管理局。在下面的一章中，将详细介绍证明管理局的使用。

当连接到因特网上后，需要能够保护你的内部网络免遭未经授权的入侵，这是一个明显的安全需求。本章第二节的内容是关于防火墙的，这一节能够给你一个如何继续的思路。能够为员工提供对因特网的访问，或者把公司网络向客户、顾客、业务伙伴以及合约人员开放，是有很多好处的。你可能想允许对特殊信息或应用、或者对特殊个人的访问。

14.1.6 标准和技术

新技术应用的快速增长和发展所带来的负面效应之一是导致了大量的标准冲突，这是很常见的。由于因特网的大小和增长，标准竞争中的获胜者可能有让人吃惊的影响，并且是一个极好的机会。为了提供用于所有安全问题的解决方案，正在进行大量的开发和配置。做为结果，正在同时宣传几个不同的标准。究竟哪个标准将广泛接受，目前还有待观察。

安全HTTP（S-HTTP）是对HTTP协议的一个扩展，它为建立会话提供了认证和加密功能。S-HTTP实际上支持很多用于客户机和服务器的安全选项。客户机和服务器协商使用何种加密机制来保证消息的安全性。这种方法能提供端到端加密，并且对于安全事务来说非常有用，但是它受限于HTTP协议。

安全套接字层（Secure Socket Layer，SSL）在传输层上提供了服务器认证、数据加密以及消息完整性。SSL的一个优势在于它能够保护几个因特网服务的安全性，而不仅限于HTTP。SSL协议在一个TCP/IP连接开始的时候建立一个安全会话。S-HTTP和SSL的使用不是互相冲突的。许多因特网事务方案同时使用了SSL和S-HTTP。安全IP（Ipv6）是一个包括附加安全功能的IP扩展规范。因特网现在正在从第4版的IP过渡到第6版的IP。Ipv6包括两个安全机制，一个认证头和封装安全负载（Encapsulating Security Payload，ESP）协议。认证头包含了在消息的基础上计算得到的认证信息。消息的完整性可以使用此信息来验证。ESP协议提供了加密部分或者全部消息的能力。安全IP协议能够极大地提高网络协议层的安全性。

正在开发和促进另外的安全协议，其目的是解决在因特网上传输金融事务的特殊问题。很明显，在因特网上提供信用卡号是非常不明智的。两个主要的信用卡Visa和MasterCard为支持安全电子事务（Secure Electronic Transaction，SET）协议进行了一场标准之战。选择金融事务协议需要考虑到谁正在支持它。对于那些已经把当前的信用卡和支付卡处理系统同现有的事务捕获和支持系统结合在一起的银行来说，它们正处在一个有力的位置上来定义因特网金融事务将走向何方。

14.1.7 Java

Sun公司的Java技术至少在该概念上已经有了分布式环境中的一些引起关注的考虑。Java是Sun公司在C++的基础上开发的一种面向对象语言，它可以与网络设备和商业工具一起使用。Java的最初设计目标是提供可以在可移植的、高度分布的异构环境中执行的程序。Java是一种解释型语言，这意味着同一代码可以在多种平台上执行。Java程序（比如Java Applet程序）首先要编译成字节码的可传输代码。字节码或者applet可以接着发送到目标平台，并由本地Java解释器进行解释和执行。使用这种解释方法，同一Java程序能够以同样的方式在任何平台上执行——只要平台上有Java解释器就可以。

Java带来的激动是它具有从互联网上下载和执行程序的能力。当考虑到安全问题时，这可能是一个梦魇。许多人对从网上下载的代码抱有戒心，这些代码可以在本地机器上加载和执行。Java包括了几个安全功能以保护applet和其外部执行环境的完整性。applet本身由Java解释器所校验以探查在执行期内可能发生的任何潜在问题。applet不能创建新进程。附加的检查保证了Java applet不会访问本地文件系统。applet只能同其原始主机建立网络连接。

然而，上面所有这些功能并不能保证不会出现安全问题。有很多闲人都非常乐意寻找安全漏洞。尽管Java限制了一些可能的情况，但是在早期的参考实现中还是发现了一些安全缺点。这表明，要让一切一下子全都完美，如果不是不可能的话，至少也是非常困难的。在类似Java这样的解释器里开发安全机制是一个反复的过程。

14.1.8 因特网PC

近来另一个有关因特网的开发是几家主要的计算机厂商宣布了廉价的因特网PC。这个概念是提供一种廉价的、功能受限的设备，只为了上网目的。这种机器由一个处理器、一个功能受限的操作系统、支持图形化和网络访问的功能以及执行Java applet的能力组成。当使用这种机器时，几乎所需的全部处理功能都可以从网上下载。这种机器一次只能执行小的applet，并且只在需要的时候执行。

这种方法具有吸引力的一个原因是它以很低的成本提供了网络访问能力。如果支持新应用软件和操作系统需要更新机器，并且需要对用户进行使用培训，那么支持一个分布式的、全功能PC网络的成本和复杂性就会增长。大型PC网络上软件的发布和改变管理也是导致问题和开销的一个主要原因。因特网PC的受限功能和低成本被认为是降低系统整体成本和避免改变及更新问题的一个方法。对于正在积极筹划实现公司企业内部网的大型公司来说，这是非常有吸引力的。

14.2 因特网防火墙

顾名思义，防火墙是用来阻止火势扩散的一堵墙。网络防火墙通过试图隔离和阻止安全问题来提供同样的基本功能。其目标是把可信网络同不可信网络隔离起来以保证前者的安全。二者之间的所有流量都强制通过防火墙，在这里流量会进行分析。会自动拒绝未经授权的流量，并且甚至两个经过授权的网络地点之间的流量也可以检查。不仅仅进入流量，而且外出流量也可以受到控制。防火墙也可以保证有关可信网络的信息（例如地址列表，这些信息可能对于黑客非常有用）不会泄露到不可信网络中。到目前为止，防火墙最常见的用途是保护

可信的内部网络在连接到因特网时免遭问题。

本章对防火墙的讨论只是提供一个对基本概念、防火墙问题和术语以及因特网服务的总体概括。有关这些领域的深入信息，向读者推荐两本有关防火墙的优秀著作[Cheswick & Bellovin,1994]和[Chapman,1995]。

14.2.1 防火墙组件

当一个组织宣称自己拥有一个防火墙的时候，这可以意味着很多事情。防火墙是一个概念而不是一个具体的模型或产品。防火墙可以构建、购买，可以有多种形式多种尺寸。防火墙可以由单一一个网络设备组成，如一个路由器。也可以包含许多设备，其中包括路由器和计算机，并具有防火墙的全部功能。下面将对典型防火墙的体系结构进行介绍。要记住，这是对典型防火墙的一个概念视图。把功能分成几个分立的组件，可以相对容易地阐述防火墙的工作机制。然而，这种典型防火墙的全部组件可以囊括在一个黑盒子解决方案中。下面分析防火墙的组件来展开对防火墙的探讨，首先从过滤路由器开始。

1. 过滤路由器

在第7章中已经看到，路由器可以用来过滤TCP/IP网络流量。这种过滤是建立在两个主要因素基础上的，包中的网络地址，引用应用程序的类型——使用一个端口号来表示。通常情况下，目标端口号表明了所涉及到的应用程序，但是从安全角度上看，让端口号等同于应用是错误的。过滤器可以在包源网络地址和目标网络地址的基础上对其进行过滤。所请求应用的类型——由源端口号和目标端口号定义——也可以过滤。协议类型（如TCP和UDP）也可以做为一项过滤内容。防火墙管理员可以控制通过过滤设备（在多数情况下是路由器）的包的类型，这些包可以是去往也可以是来自可信和不可信网络的包。包过滤器通常通过访问控制列表来启用。访问控制列表定义了由允许或者拒绝的地址、端口和协议所组成的组合体。

使用过滤技术有很多局限性。尽管路由器可以在包头信息的基础上对流量进行限制，但是它不能对包信息的内容进行任何判断。例如，一个配置为允许邮件通过的路由器是不能够检查出邮件中的时间炸弹的。第二，路由器访问控制列表在其配置中可能会含义模糊。对于一个没有经验的或者粗心的防火墙管理员来说，他很可能会不小心犯下错误，从而允许未经授权的服务通过防火墙。最后的安全考虑是路由器不应该配置成提供不必要的内部网络信息，也不应该从因特网上接受有关内部网络的信息（例如，内部路由信息）。最后，许多基于TCP/IP的服务都是在一个已知端口号上打开通信的，但是其后面的通信都是使用所指派的更高的端口号。这些端口号并不总是可以预测的，因此当应用过滤规则的时候一定要留出一段端口号范围。某些防火墙具有有状态包过滤的功能，它们允许在同客户机会话进行通信的过程中可以打开一个更高的端口号。当通信结束时，更高的端口就会关闭。

2. 屏蔽子网

在第一次世界大战中，西方前线中使用了战壕系统，这种系统的设计目标是提供一个称做深度防守的概念。总共设计了三个战壕，其思想是在第一个战壕被敌人突破以后，前方军力可以撤回到第二个预备战壕。这些战壕被设计成锯齿弯型。即使敌人突破了一段战壕，它们也不能控制整个战壕。第一套战壕为前线内部提供了边界防护。

一个屏蔽子网起到了内部网络和因特网之间的边界防护系统的作用。它进一步把内部网络同因特网隔离保护起来。这种特殊子网的使用也可以引入一类特殊的计算机，这种机器称

做桥头堡主机。这些主机和内部网络以及外部网络之间的流量可以由路由器来控制。路由器不仅仅可以限制网络流量的性质，而且还可以强制进入流量和外出流量必须通过屏蔽子网上的特殊计算机系统——即桥头堡主机。

3. 桥头堡主机

桥头堡主机（桥头堡是一个要塞的外部部分。通过它，防卫者可以控制许多不同的通道）是直接同不可信网络打交道的任何计算机系统。因为桥头堡主机是入侵者的主要攻击点，所以它们必须要受到严格的安全保护。桥头堡主机可以用来控制有效的进入流量，并把这些流量转发到内部网络上的适当地点。另一方面，桥头堡主机起到了去往因特网的内部网络流量的单一控制点的作用。桥头堡主机也能够适应一些TCP/IP服务的需要，即在一段时间内通过有选择地开放所需端口，可以在不可信端口上进行通信。它们也可以为内部用户提供对应用程序或者TCP/IP网络服务的访问，这种访问是基于代理之上的。代理允许在这些应用上施以另外的控制。

4. 双穴主机

许多计算机系统可以使用多个网卡。这种系统的术语名称是“双穴主机”，它们可以在多个接口之间移动流量。在一个因特网防火墙实现中，一个接口通常连接到内部网络上，而第二个接口则连接到因特网上。这里的窍门并不是自动传递两个网络之间的流量，而是强制所有通信都通过双穴主机。内部网络上的系统同外部因特网系统进行通信必须要通过双穴主机，反之亦然。这里不允许有任何直接的通信，所有的通信都会监视。另一方面，双穴主机要比其他类型的方案更难以建立和管理。

5. 代理服务

常识上，一个代理就是授权代表你的一个人。例如，在公司年度会议上，一个代理可以指派代表一个股东进行投票。与此类似，防火墙代理服务也是代表希望得到因特网资源的用户进行动作。代理服务提供者对用户来说是透明的，当用户访问因特网服务的时候，他们不知道有个中介正代表着他们进行访问。

使用代理服务器有两个主要好处。第一，用户不需要登录到桥头堡主机上，也不需要有其上的账号。这使得桥头堡主机可以尽可能地保持“瘦”和简单。从安全角度上看，把东西尽可能地保持简单总是一个好的原则。第二，代理服务器的使用使得用户行为的审计跟踪可以记录下来。对代理服务器使用审计跟踪可以探查那些访问不当网站（如提供色情内容或者非法内容）的员工。许多代理也透明地“清洗”内部IP地址，即把你的内部地址空间隐藏在桥头堡主机地址之后。

代理服务器可用于许多常见的因特网服务，这包括telnet、FTP和HTTP。然而，使用代理服务也有一些缺点。尽管大多数常见的因特网服务都有其可用的代理版本，但是你所希望的某个特定服务可能就没有。例如，找到对SNMP协议的支持可能就是个问题。然而，一般来说，在可能的时候使用代理是值得推荐的。下面将探讨两种不同的代理服务实现方法——电路中继和基于应用网关的代理。

6. 电路中继

电路中继是一种代理服务，其通常驻留于桥头堡主机上。它对来自客户机的请求进行检查，如果有效，就转发到因特网的相关服务器上。使用电路中继，客户机上必须驻留有特殊的软件。电路中继通常要求客户机上安装和配置了特殊的软件。SOCKS是一种常见的电路中

继服务包，它是由David和Michelle Koblas开发的。电路中继可以控制一个请求的源地址和目标地址，但它不会检查请求的性质或内容。例如，对于一个在自己的工作站和一个因特网服务器之间使用传输工具FTP的用户请求来说，请求本身会被控制，但是使用FTP传输的文件不会检查。用户可以传输未经授权的内容，但是电路中继代理对这种行为无能为力。若要在通信内容的基础上实行智能控制，那么必须要使用一个应用网关代理。

7. 应用网关

应用网关代理不仅仅有能力控制连接，而且能够对连接的性质进行检查。应用网关把用户请求传递到一个真正的应用服务，在那里，所请求的应用会把判断标准应用到用户请求上以决定是否允许其通过。有一种情况不很常见但却是所需要的，即这种代理类型的使用涉及到电子邮件的控制。邮件消息可能会携带特洛伊木马程序、病毒或者其他有害的内容。尽管进行这种任务的技术仍旧在改进之中，但是一个应用网关代理可以截获所有的邮件消息。应用网关代理可以打开消息并检查其内容以防有非法内容。它们也可以在审计跟踪的使用中允许一个更好的粒度。如果FTP连接通过应用网关来实现，那么它可以对用户连接的每个基键进行记录。

14.2.2 典型的防火墙

现在我们来看一下防火墙的一般体系结构（记住，实际上没有什么典型的防火墙）。这种体系结构涉及到了一对路由器和两个计算机系统的使用。其中一个路由器称做外部路由器，它通过一个广域网（Wide Area Network，WAN）接口同因特网相连；第二个路由器控制着到内部网络的连接。同这两个路由器相连的是一个屏蔽子网，它由两个系统组成。第一个系统称做双穴桥头堡主机，它用来过滤和处理进入网络流量；第二个主机用来执行选定的因特网服务，如电子邮件、NTP和DNS等。

图14-1说明了这种防火墙的典型体系结构。

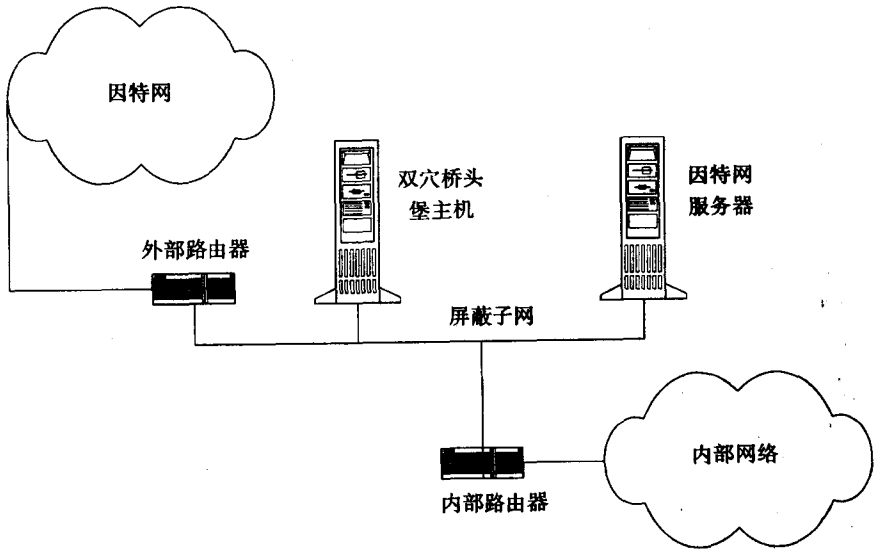


图14-1 一个典型的因特网防火墙

外部路由器提供对因特网的初始连接。它用来过滤桥头堡主机和因特网之间的所有未经授权的流量。所有的进入流量都定向到桥头堡主机以进行处理，并且所有的外出流量也必须从桥头堡主机流出。内部路由器限制了来自内部网络流量的性质和起源。它只转发那些去往桥头堡主机或者因特网服务主机的流量，并且只允许来自这些系统的进入流量。因此，网络流量被强制只能去往经过选择的目的地，在那里它们可以受到更好的控制。[值得怀疑的是，图14-1中所示的子网不是一个“屏蔽子网”。它需要另一个路由器来控制对其主机的访问——其主机被“官方”看作是一个“屏蔽子网”，但是大多数人认为防火墙是足够的]。

内部和外部路由器不仅仅根据源地址和目标地址来限制流量，而且还要根据流量的类型——即TCP/IP源端口和目标端口来限制。进入流量通常限定为电子邮件、因特网新闻服务、域名系统以及网络时间协议服务。所有这些服务的流量都定向到因特网服务器，在那里，它们会在转发到任何内部系统之前先处理。内部路由器保证了只有这些服务可以允许，并且它们必须要通过桥头堡主机。桥头堡主机也可以提供一些代理因特网服务，如Telnet和FTP。在一个位于防火墙之外的系统上提供HTTP服务是很常见的（如果防火墙不能充分支持进入HTTP流量的话）。从前面可知，代理服务器可以代表合法用户来执行服务，并且不需要用户真正登录到网关系统上。

防火墙体系结构是一种“皮带加挂钩”的方法，使用它来为员工提供对因特网服务的访问时，组织的“裤子不会掉下来”。如果入侵者获得了从因特网上访问进入主机的权力，那么内部路由器会阻止其对内部网络的访问。同样，外部路由器可以阻挡内部网络和外部网络之间的未经授权的对话。在这种体系结构中，防火墙中一个组件出现问题不会导致整个系统被破坏。

1. 虚拟专用网

可以使用加密来保护防火墙之间的流量，这种做法产生了一个概念：虚拟专用网（Virtual Private Network，VPN）。如果两个公司网络需要通过一个不可信网络（如因特网）来建立连接，那么它们之间的流量应该保护起来。如果这两个网络都受到防火墙的保护，那么这些防火墙就是放置加密机制的合理地方，这是因为所有的流量都必须要通过它们。很多防火墙厂商都把VPN功能作为其产品的一部分。

2. 对防火墙的攻击

在过去5年中，因特网用户的网络连接已经遭受了很多攻击。这些攻击使用了很多破坏防火墙系统的技术，使用这些技术，攻击者可以进入组织的内部网络。路由器和桥头堡主机的正确配置是非常关键的。

例如，对于一个检查外部包的防火墙路由器来说，它必须能够阻止任何声称自己来自内部网络的IP包的进入。而且，它应该永远不会把来自外部的路由信息接受到自己的内部网络中，并且也不应该允许由一个外部给定的路由器所发送的返回包的路由。在网络上接收到的任何声称自己来自环回地址的包也应该拒之门外。你可能仍旧想使用环回接口。很明显，配置路由器是一个敏感的安全问题，这应该在路由器控制台上进行。路由器绝对不能接受任何来自因特网的控制或者路由信息。关于如何把包路由到内部网络上，路由器绝对不能接受外来的信息。并且，路由器应该尽可能地把有关它所试图保护的内部网络的信息隐藏起来。

与此类似，桥头堡主机也必须尽可能少地泄漏内部网络的信息。特别是域名系统的配置，它绝对不应该把内部网络的信息泄露给外部网络。桥头堡主机应该有少量的用户账号。如果可能的话，对堡垒主机的直接登录和FTP访问都应该限定在系统控制台上。每个桥头堡主机都应该尽可能地接近“裸机”，也就是说，编译器、编辑器、调试器以及其他任何可能会为攻击者提供帮助的工具都应该禁用或者删除。不用的网络服务和功能，如UUCP和IP转发，都必须要删除或者禁用。应该使用监视安全控制的软件（在第12章中介绍过），另外值得推荐的是使用完整性检查程序来定期检查对桥头堡主机的任何改变。所有同安全相关的补丁都应该尽可能快地安装上。

14.2.3 构建还是购买

构建一个能够抵抗来自因特网的反复攻击的防火墙绝对不是一个可以交给新手去做的简单工作。我们曾看到过很多人喜欢花上半天时间来尝试访问连上因特网的系统。很多装备了自动探查网络工具的“高手”们都能看到你的防火墙并试图访问它。毫无疑问，为组织构建防火墙并保证其安全是一项很复杂的任务，它绝对不是可以“边学边做”的。对于大多数组织来说，如果其防火墙被攻破了，那么那些高级管理人员肯定谁也笑不出口。

强烈推荐，除非你的组织有这个领域的技术专家，否则请利用厂商或者第三方提供商来为你构建和实现防火墙！如果把这个工作委托给新手去做，那麻烦可大了，他必须要学习很多东西。但是这并不意味着你在其中没有任何责任。对于一个组织来说，不管是购买一个防火墙还是请个顾问来构建，组织都应该努力学习关于这项技术的尽可能多的知识。厂商或者顾问离开以后，有关防火墙的操作和安全的责任会一直落到组织自己的头上。

最后，防火墙技术仍旧在发展，所以组织应该跟上所有最新技术的步伐，这是非常重要的。可以预见的是，不断增长的客户需求会导致出现新的安全暴露问题。毫无疑问，组织应该及时采用新的防火墙技术以解决这些暴露问题并满足新的需求。

对因特网技术的讨论到这里就全部结束了，下面我们探讨一下控制员工使用因特网的标准。

14.2.4 因特网可接受的使用策略

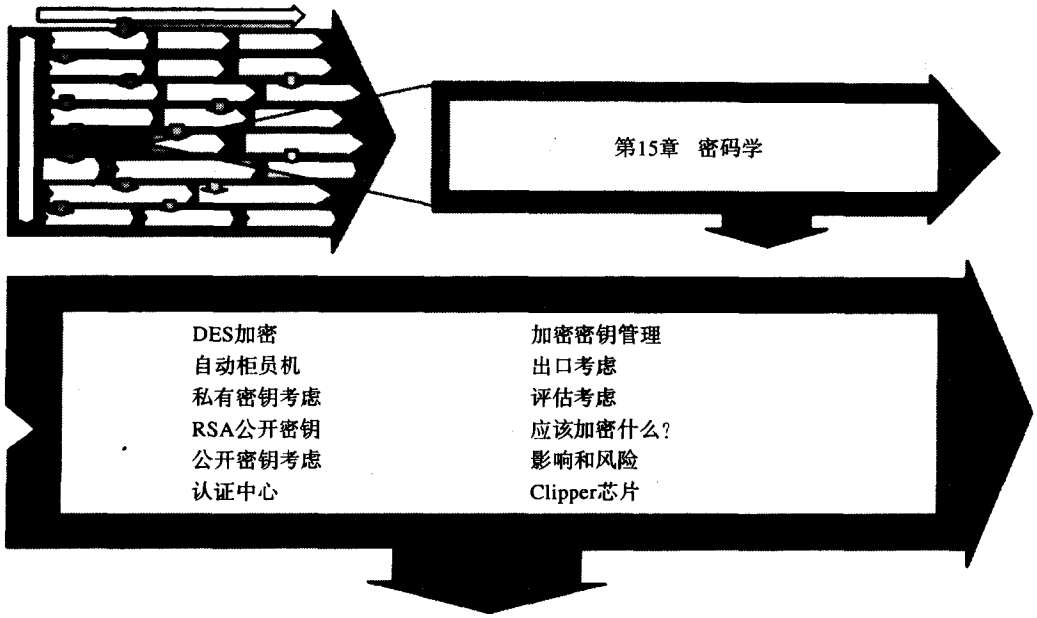
关于员工使用因特网，有很多问题都是公司应该解决的。员工明白他们在使用公司因特网连接时所承担的责任吗？员工知道电子邮件如果不加密的话就有可能会被截获、篡改和伪造吗？电子邮件地址欺骗是一种使用特殊工程攻击来获得未经授权信息的方法。如果员工把公司因特网连接用于个人业务目的，或者使用对公司有害的或违反公司规定的公司电子邮件地址在因特网新闻组上大放厥词的话，那会怎么样呢？对于访问种族歧视或者色情内容的员工来说，公司应该拿他怎么办呢？

员工必须要知道，当使用公司资源来访问因特网的时候，他们所代表的是公司的大众形象。公司信任他们会以一个合理的方式来进行工作，但是组织不会批准对这种信任的滥用。有很多问题都是公司的管理部门应该解决的，只有这样才能“防患于未然”。解决这种需要的一个有效办法是实现一个因特网可接受的使用策略。该策略应该对员工在使用公司资源进行因特网访问的时候可接受的行为和应该承担的责任进行明确的定义。

14.3 结论

如果有正确的技术、员工培训程序以及一个由管理部门签署的明确策略，那么因特网的使用可以带来很多好处。如果没有正确地开发这么一个策略，那么组织就会有很多问题。因特网服务的安全性在近几年来有很大的提高和改进，但是时至今日，安全性仍然是计算领域的一个大问题，并吸引了广泛的注意。不管将来的开发如何，公司的因特网使用都必须要进行仔细的计划。关于预期会有什么好处以及需要什么工具，公司都应该对目标进行明确定义。但是不要忘记解决了技术问题以后，人为问题仍然存在！

第15章 密 码 学



自从人类开始互相通信以来，就有了对保密的需要。密码学的定义是把信息打乱成不可理解的形式、然后再把加密过的信息还原成可以理解的形式的原则、手段和方法。换句话说，密码学是秘密书写的科学。最早的加密例子是在4000年前的埃及手迹中发现的。古希腊、中国以及罗马文明都曾使用过让人惊奇的先进密码技术。在中世纪的阿拉伯世界里，密码学和密码分析学的使用是非常活跃的。尽管情人们和神学家们也有秘密传信的悠久历史，但在历史上密码学主要还是用在外交和军事方面。

美国之所以介入第一次世界大战，一个起作用的因素是英国军事情报部门对一封电报的截获和部分解密。这封电报被称为齐莫尔曼电报（齐莫尔曼是德国外交部长），它暴露了德国和墨西哥的秘密协商。德国承诺在战争成功结束以后帮墨西哥夺回被美国占领的土地。

第二次世界大战促进了密码学和密码分析技术的发展。在20世纪30年代后期，波兰军事情报部门的工作人员复制了一台称做Enigma的德国加密机器。在战争爆发以后，这台机器送到波兰的法国和比利时盟军。Enigma机器类似于一台打字机，它使用了一套称做转子的加密轮。这台机器的高级版本使用了附加的转子，它被认为是不可破解的，但是英国军事情报部门能够破解这种加密方案。从1940年以后，越来越多的德国海军和陆军情报对盟军都成为可用的了。

美国军事情报部门在1942年6月的中途岛之战中起到了重要作用。在一个称做PURPLE的秘密计划之下，对日本通信（称做MAGIC）的解密把日本进攻中途岛的意图预先警告给美国海军。这份情报使得美国海军突袭了山本的舰队。美国的这次胜利是第二次世界大战的

转折点。

数据加密通常认为是为数据存储和传输提供保密性的最佳方法。加密是使用一个算法把数据从一种形式到另一种形式的变换，在变换过程中要使用一个或多个加密密钥。如果不使用正确的密钥来解密数据，那么存储或传输的加密过的结果数据是没有意义的。只要数据需要在一个不可信网络上保持机密，那么就应该对数据进行加密。

通常在两种情况下需要对数据进行加密。当需要安全的存储或者传输秘密的或者需要高度机密性的信息时，需要使用加密。加密应该应用到一段事务消息上，并且不应该在数据字段边界对齐。一方面，数据需要保护；另一方面，事务需要保护。

加密方法主要有两种：私有密钥和公开密钥。对于前者来说，加密和解密都使用同一个共享的秘密的或者私有的密钥；后者使用一个秘密的或者私有的密钥来进行解密，但不同的是它使用一个数学上配对的公开密钥来进行加密。根据具体需求和实现的不同，这两种方法都有自己的优点。

15.1 私有密钥加密

私有密钥加密在一个或多个参与方之间使用一个共享的秘密密钥。对于各个参与方来说，拥有该密钥意味着他们之间可以互相认证。私有密钥技术的一个常见用途是用来对用户密码进行认证。用户密码的加密版本保存在计算机上。用户在登录时提供用户ID和相关的密码。然后密码加密，如果加密后的密码同所保存的与用户ID相关的密码相匹配，那么用户就会通过认证。

私有密钥技术也可以用来对数据进行加密。密钥保存在加密过的文件里，并且解密数据必须要提供密钥。私有密钥加密的缺点在于密钥是在两个参与方（用户和计算机）之间共享的。由计算机过程生成的会话密钥可以用于使用加密的安全通信，它不需要用户的任何参与或者知道。这些密钥可以保存在内存中。用于加密数据（以加密格式保存）的密钥是由用户而不是由计算机系统所保管的。

15.1.1 DES加密

最常使用的私有密钥加密标准是IBM在20世纪70年代早期开发的数据加密标准（Data Encryption Standard，DES）。它是用于加密系统的事实标准，并且是世界上使用最广泛的加密机制。这种私有密钥系统广泛用于包括自动柜员机和零售网络在内的金融网络中。1977年，它被美国政府采纳为联邦信息处理标准（Federal Information Processing Standard，FIPS PUB 46），并在1981年成为美国国家标准（American National Standard，ANSI X3.92）。关于该算法使用模式的更进一步的分类包含在ANSI标准X3.106中。这种加密对个人标识号（Personal Identification Uumber，PIN）的具体应用是在ANSI X9.8和国际标准化组织（International Standards Organizaiton，ISO）9564中提出的。DES算法使用一个56位的私有密钥（另加8位用于密钥完整性检查），并操作64位的数据块。

图15-1说明了这种基于私有密钥的加密过程。客户使用一个私有密钥和一个加密算法把消息变换成不可理解的形式。服务器使用同一密钥和同一加密算法来逆转该过程，并重新建立可以理解的消息。该过程的安全性取决于私有密钥的秘密性。

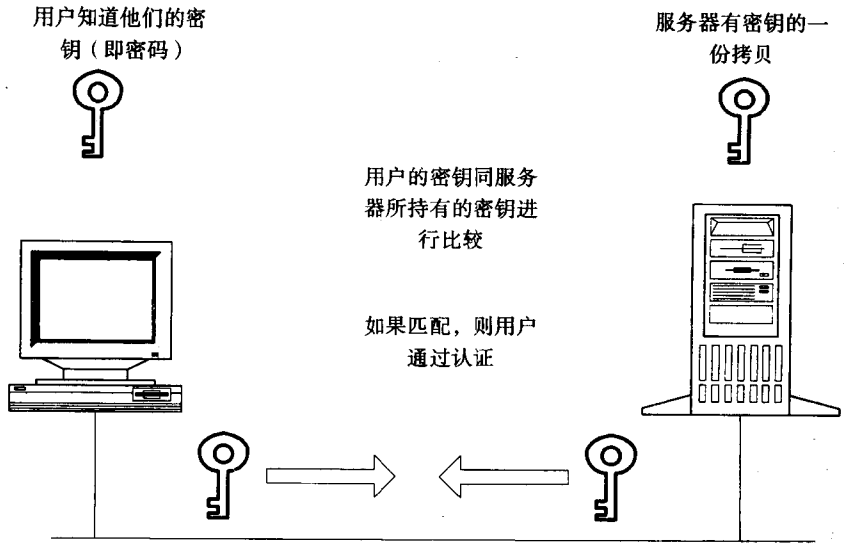


图15-1 私有密钥加密

这种加密过程可以使用软件也可以使用硬件来处理。几家厂商提供了可以运行在许多不同平台上的加密产品。另外，也有专门为执行加密算法而设计的特殊硬件。有些产品在智能卡或者PCMCIA卡中提供DES加密能力。在这种情况下，算法和加密密钥驻留于卡上，并且能够随处移动。如果速度要求非常重要，或者软件解决方案不能提供足够的处理能力，那么硬件解决方案是更有吸引力的。

DES加密算法至今还没有被成功地破解过，至少是就我们所知的而言。然而，可用计算能力的飞速增长可能会在将来提供破解该算法的能力。据估计，一个每秒能进行一次DES加密的计算机需要2000年的时间才能破解一个密钥。然而，DES毕竟是很老的技术。理论研究已经表明DES是可以攻破的。破解该算法的能力是建立在加密密钥的大小基础上的。三重DES（Triple-DES，使用两个密钥来运行三次DES算法）的使用能够延长该算法的寿命。

15.1.2 自动柜员机

我们可以使用自动柜员机（Automated Teller Machine，ATM）来得到现金或者把钱存到自己的银行账户中。大多数的ATM都使用DES加密算法来提供所需的事务机密性。事务中的一些部分实际上会加密两次（使用两个不同的密钥），所产生的机密信息至今尚未被破解过。当一项ATM事务必须通过网络边界时，它的机密性就变得更加重要。图15－2说明了一个典型的ATM事务。

对于典型的ATM事务来说，其中有两个部分是高度机密的。第一个是用户使用ATM时提供的个人标识号（PIN），另一个是事务本身的详细说明。插到机器内以启动事务的银行卡提供了所需的认证。用户在事务期间输入的PIN号提供了用于认证的基础。该PIN使用一个专门用于PIN的密钥来加密，并且加密后的PIN插入到事务消息中。

然后，该消息中的一些部分使用一个专门用于消息安全传输的密钥来加密。在这种方式下，PIN实际上加密了两次。当消息到达主机系统时，消息就使用传输密钥来解密。如果消息

必须要通过网络边界，那么它会使用原始网络传输密钥来解密，然后再使用目标网络传输密钥来加密。为了降低主机的负担，该过程一般是由特殊功能的硬件来执行的，并且这也提供了一个安全的、防篡改的能力。在从ATM中得到现金之前，上面所有这些处理都要先进行。

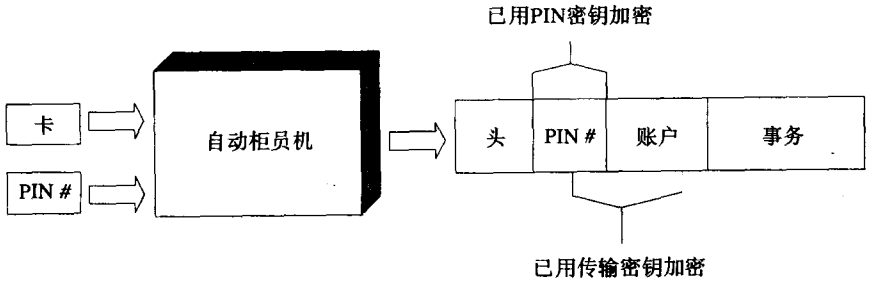


图15-2 ATM交易

15.1.3 私有密钥的考虑

私有密钥DES算法的使用已在金融界生根。几乎所有的自动柜员机都使用DES来保护从机器传输到处理中心的数据。加密和解密使用同一密钥，因此必须要安全保存密钥以免泄漏。如果泄露了一个密钥，那么使用该密钥的所有地点都会有很大的损失。密钥的保存和使用必须要在一个安全的方式下进行。

密钥管理是DES加密的一个重要问题，这是因为对于处理两端来说加密和解密密钥必须是已知的。必须要使用一个更安全的方法来改变和保存密钥。执行加密过程需要计算能力。需要加密或解密的数据量越大，所需要的处理能力就越大。私有密钥方法非常适合这种情况，一个地点需要同相对较少的几个应用程序或用户进行安全交互。加密密钥的管理可以在一个安全方式下完成，并且密钥的机密性能够得到保证。

15.2 公开密钥加密

1975年5月，由Whitfield Diffie和Martin Hellman开发的公开密钥技术成了密码历史上的一个转折点。公开密钥加密——也称非对称加密——使用两个分离的但数学上相关的密钥。每个用户有一对密钥，其中一个是应该严格保持机密的（私有密钥），另一个是同其他用户或者计算机共享的（公开密钥）。这两个密钥在数学上是相关的，二者都要在加密/解密过程中使用。一条使用公开密钥加密的消息只能使用私有密钥来解密。这种方法较之私有密钥技术的先进之处在于私有密钥永远不能同其他参与者共享。公开密钥技术的另一个优点是它可用于创建数字签名。数字签名是用来校验电子消息发送者和内容的。

15.2.1 RSA公开密钥

RSA（由Ronald Rivest、Adi Shamir和Leonard Adleman提出）是一种公开密钥算法，它是建立在Diffie -Hellman最初开发的算法之上的。这种系统使用两个相关的、互补的密钥，其中用于加密的那一个是需要保持机密的，而另一个用于解密的是公开的。对于加密过程来说，只有私有密钥是已知的，它必须要保持机密。RSA算法的使用正在迅速扩大，特别是在

电子消息和电子邮件领域。这种算法的缺点在于它需要很大的计算能力，因此其速度要比DES的慢。

DES加密密钥的大小已经固定为64位，而基于RSA密钥的大小是可变的。RSA算法可以使用一个比DES长得多的密钥（常用的是512位）。因此，RSA被认为比DES更健壮。然而尽管这样，429位（RSA 129）密钥在1994年4月被一个由数百个研究人员组成的队伍使用许多小时的计算时间破解了。

RSA技术也可以为文档认证提供数字签名。一个消息摘要是通过使用一个作用在文档上的散列函数产生的，然后该文档使用私有密钥进行加密。这种摘要是附加到消息上的数字签名。数字签名的解密需要使用公开密钥和同一散列函数，它应该产生同消息摘要一样的结果。1985年，ElGamal在RSA系统的基础上提出了一个替代的公开密钥加密系统。这种系统用于认证，并形成了被提议为美国数字签名标准（U.S. Digital Signature Standard，DSS）的基础。图15-3说明了公开密钥加密过程。

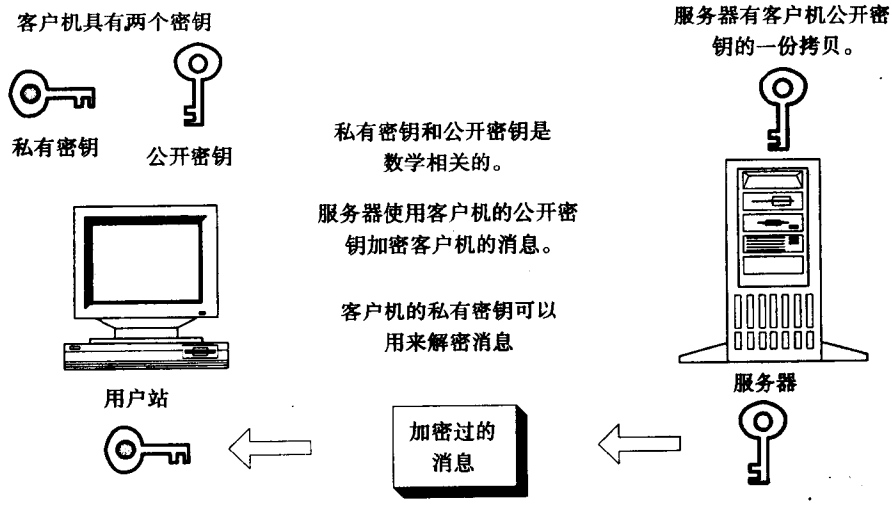


图15-3 公开密钥加密

在图15-3中，客户机有两个加密密钥，公开的和私有的。公开密钥对于服务器也是已知的。发送到客户机的消息是使用客户机的公开密钥进行加密的。然后，客户机使用私有密钥来解密消息。使用这种方法，只有具有私有密钥的客户机才能成功地对该消息进行解密。

参与双方的安全通信可以通过公开密钥的一次交换来实现。例如，如果Beth想把一条安全消息发送给Ted，并且要保证只有Ted才能读取该消息，那么他可以使用Ted的公开密钥来加密这条消息。只有Ted才能解密这条消息（使用他的私有密钥）。Ted可以使用Beth的公开密钥来加密他对Beth的响应，然后Beth可以使用他的私有密钥来解密Ted的响应。如果Beth想让任何人都可以读取该消息，那么他可以使用他的私有密钥来加密消息，知道Beth的公开密钥的任何人都可以解密该消息。如果Beth想证明只有他才发送了一条消息，那么他可以使用他自己的私有密钥来为消息（Ted将使用Beth的公开密钥来解密该消息）创建一个数字签名。关于使用谁的密钥可能会有点迷惑。必须要小心谨慎以确保安全性得到正确地维护。

15.2.2 公开密钥的考虑

公开密钥加密算法要比私有密钥算法要进行更多的计算。所需的计算能力总量是基于所使用的加密密钥的大小的。密钥越大，破解它的可能性就越小，但加密和解密过程所需的处理能力就越高。如果大量的用户都需要独立的密钥，那么密钥管理就成了一个问题。每个用户位置都必须维护或者能够访问用于用户的公开密钥的或者他们希望与之交互的应用程序的目录。公开密钥方法适合于这种情况：用户要求同其他几个应用程序或者用户进行安全交互。这定义为用户到应用程序的多对多的关系。公开密钥算法通常包括可用于确认的数字签名功能。

> 大多数加密机制的强度都是基于两个因素的，算法的完整性和所用密码的长度。大多数加密机制，如RSA和DES，支持不同的版本，其基础是在特定实现中使用密钥的长度。例如，RSA 129使用一个129位（或者一个429位）密钥长度。DES有一个称做弱DES的实现，由于它使用了一个更短的密钥，所以它不受标准美国出口限定的限制。

15.2.3 认证中心

为数百个用户管理公开密钥是一个大问题。每个用户如何能够知道他们可能与之通信的每一个其他用户或服务器？这个问题的一个解决方案是使用由认证中心所管理的证书。证书是一个签名过的消息，它指定了名字和公开密钥。认证中心是这些证书的安全仓库。一个证书是一份数据记录，它一般包含了用户的公开密钥、所用加密算法的类型、证书所有者的名字、证书有效时间、证书颁发者以及用来确认证书的数字签名。

如果一个用户想使用公开密钥加密把一条消息发送到另一个用户或者服务器，但不知道预定目的地的公开密钥，那么他可以向认证中心请求得到这个密钥。用户必须要首先知道认证中心的公开密钥，并使用它来重新获得该认证中心所服务的那个用户的公开密钥。认证中心也可以用来检验用户的认证凭证，并且可能会需要为用户使用其他网络应用和服务生成认证证书。

认证中心有责任保护和管理证书，这包括建立、截止和撤销证书。用于证书和认证中心处理的具有统治地位的标准是开放系统互联（Open System Interconnect，OSI）标准X.509。X.509认证框架是一个在计算业界广泛接受的标准，而不仅仅是OSI网络。该标准定义了证书的结构、管理证书的协议以及可用来提供认证的方法，其中包括数字签名。

一个主要的问题是：谁提供认证中心？现在已经存在有许多认证中心，其中每个都有不同的用户群。一个认证中心可以代表一个用户同另一个认证中心进行认证事宜。因特网请求注释（Internet Request for Comment，RPC）1422包含了认证中心的组织和层次结构应该遵循的建议。银行、邮局甚至信用卡处理者都将争夺运作认证中心的机会看来是自然的。电话公司也已开始宣布他们对提供网络认证中心服务的宏伟大计。

15.3 加密问题

当选择加密技术时，加密算法所需要的密钥管理是涉及到的一个复杂问题。如果使用一个私有密钥算法，那么对消息或数据进行加密或解密的所有地点都需要安全地使用和保存同

一密钥。为了维护数据所需的安全性，对私有密钥的任何改变都需要在所有这些地点上同步地以一个安全的方式来实现。公开密钥方法要求为系统的所有用户选择和实现独立密钥对。这可能会导致一种需求，即需要生成、分派和使用大量的密钥。公开密钥加密的基本问题是密钥的增殖问题，私有密钥加密的首要问题是用户必须知道私有密钥。

15.3.1 加密密钥管理

加密密钥的管理要求必须解决几个具体的密钥管理问题。密钥生成和注册指的是把密钥同其预期应用绑定在一起的能力。问题在于如何把密钥同一个用户关联起来。如何把加密密钥传送到所有需要加密的地点？这是密钥发布所涉及到的问题。密钥激活/停用指的是启用或者禁用密钥的能力。密钥可能会在一个同间隔或时间相关的过程中改变。特殊的密钥可能只用于改变的加密密钥的安全发布。密钥更新或者替换是从一个加密密钥到另一个加密密钥的组织改变。密钥废除或终止指的是把加密密钥标记成无效的能力。如果怀疑一个现有密钥受到了破坏，那么在这种情况下需要进行密钥废除或终止。

美国国家标准协会（American National Standards Institute，ANSI）已经定义了一个标准来管理金融部门的密钥发布过程。金融单位密钥管理（批发）标准——ANSI X9.17是在1985年定义的。加密密钥的安全改变需要使用多层密钥。X9.17标准为建立新密钥和替换现有密钥定义了一个协议。

当考虑到对密钥契约的需求时，另一个密钥管理问题变得很重要。即如果有额外的目的，如何把一个加密密钥保存在第三方契约中。如果一个具有公司加密数据的用户离开公司并拿走密钥，那么在这种情况下上面的问题就变得非常重要。如果不能拿到加密密钥，那么访问这些数据是不可能的。预先把密钥保存在契约中可以解决该问题。另一方面，公司也需要具有一个要求员工秘密保管加密密钥的策略。如果钥匙挂在门外的把手上的话，那么门再坚固的锁也形同虚设。

15.3.2 出口考虑

美国可用的加密技术归类为军需品。除非可以得到一个专有的出口许可，否则这些技术是不允许出口到北美境外的。许可证的批准通常是依赖于具体加密算法和所针对的数据类型的。如果加密需求同一个包含环境中的特殊数据有关，那么获得许可证就相对容易些；如果申请许可证的实现可用于通用加密而不是局限于特殊数据，那么获得许可证的难度就很大。对于出口许可证应用来说，从应用提出到批准可能会花数个月的时间。针对金融机构的出口限制有一些宽松之处。

如果要执行加密算法的应用程序所使用的算法是针对于预定义的数据和预定义的功能的，那么获得出口许可证可能会容易些；如果要出口用来开发加密能力的通用技术而不是应用程序，那么获得出口许可证就困难得多。近来的开发已经表明，有些公开密钥技术的发布和使用涉及到了版权问题。同任何捐赠软件一样，捐赠加密机制的用户也应该检查这些机制使用的版权或者官方限制。

15.3.3 评估考虑

根据所选择的加密算法，计算机资源会受到不同程度的性能影响。公开密钥算法通常要

比私有密钥算法需要更多的处理资源。资源的利用情况也取决于所选择的用于公开密钥加密的加密密钥的大小。加密密钥的位数越多，加密过的消息就越安全，但是也就需要越多的计算资源。加密算法的强度也是很重要的，它必须要同加密密钥的大小一同考虑。

对加密技术支持的访问应该是可用的。密钥生成和分派的易使用性是一个因素。所需密钥管理过程的可用性也是一个考虑事项。密钥管理本身必须能以一个安全的方式进行处理。对现有应用和过程所使用的加密技术的完整性的影响也应该是一个考虑事项。使用硬件来加密要比使用软件方法更昂贵，但是这样可以避免性能问题。只有在需要一个完全处于应用系统之外的解决方案的情况下，硬件解决方案才值得考虑。

15.3.4 应该加密的内容

从前面可知，加密应该用于两个不同的目的。加密的第一个目的是为认证或者授权组件（例如密码、个人标识号PIN和任何消息认证标记）的存储和传输提供机密性，第二个目的是为了网络传输的机密性。

对于加密服务的一个最小或者推荐实现来说，其变化会是在加密的信息量上而不是在加密技术上。加密应该总是用到登录密码或者其他用于认证或授权的信息的安全存储和传输上。它应该用于客户指派信息（如PIN）的安全存储和传输。当其包括在一个传输消息中时，任何消息认证代码或者数字签名都应该加密。

在网络上特别是在一个不可信网络上传输的消息应该加密。加密过的数据不应该限定在事务字段边界，并且应该包括每个消息中可能改变的字段部分。这样会使得推测或者重放消息中的值变得非常困难。例如，每个消息的11～27字节都应该加密。这能防止攻击者重放一条截获的消息。任何归类为高度机密的信息在存储或者传输时都应该进行加密。

15.3.5 影响和风险

加密实现对支持加密算法的处理资源的影响是最明显的。加密算法需要CPU周期来完成加密过程。需要加密的数据字段越多或者越大，支持加密所需的CPU资源所受到的影响就越大。增大公开密钥的长度也会增加所耗费的处理周期。

私有密钥加密方法的主要风险是加密密钥可能泄漏。因此，加密密钥的安全存储是必需的。如果每个用户都使用单独的加密密钥，那么密钥泄漏的可能性就会受到限制，但是这会增大密钥管理问题的难度。管理加密密钥所受到的影响取决于密钥的数量和所要求的地点。必须要小心谨慎以保证会使用正确的加密密钥。

有一种方法可以提供不会影响系统处理的加密，那就是使用硬件加密设备。通过使用特殊的硬件设备，加密可以完全在应用程序之外进行。如果所有地点都使用了加密设备，那么这种方法可以保证客户机和服务器间每一个事务的安全性。然而，这种方法会带来很大的花费，这要取决于使用地点的数量和对支持和备份单元的需求。

15.3.6 Clipper芯片

克林顿政府对硬件加密标准的提议引起了广泛的争论。这项消息数字加密标准提议是基于Clipper加密计算机芯片设计和Skipjack加密算法的。Clipper芯片的推荐使用包括了其在计算机硬件和无线数字通信领域的应用。

关于这种芯片的大部分争论都是这种芯片的一个安全后门，该后门允许美国政府访问芯片的保密数据。该后门称做LEAF（Law Enforcement Access Field，法律强制访问字段），访问它需要一系列的密钥，而这些密钥会由美国政府机构所持有。有关Clipper芯片的问题包括芯片的成本和性能、担心侵犯隐私权、芯片在美国境外的可接受性以及对美国计算机制造业和LEAF访问控制机制强度的影响。AT&T贝尔实验室的Matt Blaze博士在1994年6月的《Protocol Failure in the Escrowed Eneryption Standard》一文中描述了LEAF契约加密标准（Escrowed Encryption Standard，ESS）[NIST94]算法中的可疑脆弱性。

15.4 数字签名

数字签名是一种证明电子文档的来源和内容的方法。数字签名是建立在如下思想上的，如果把一份文档的整个内容作为对一个加密算法的输入，那么即使对源文档进行最小的修改也会导致加密输出的明显变化，这样就可以很容易地检查文档是否完整。用于文档的加密机制称做散列算法，其输出称做消息摘要。如果散列算法也使用一个仅由发送者才知道的加密密钥，那会如何呢？结果将产生一个文档，对文档的最小改变也可以检查出来，并且文档创建者可以惟一地识别。

消息摘要是使用发送者的私有密钥创建的，该密钥为源文档产生一个惟一的数字签名。把发送者的公开密钥给接收者，它用来验证文档中的数字签名，该过程称为签名验证。图15-4说明了如何使用公开密钥加密来生成一个消息摘要。接收方要重新建立该消息摘要，并将其与消息中附加的摘要进行比较。如果二者相符，那么该消息就确实来自发送方并且没有篡改过。

如果一个消息摘要的密钥只有发送者才知道，那么该摘要的创建就称做数字签名。消息摘要会同消息的原始拷贝一起发送到文档的接收者。接收者使用源文档和一个相似的散列算法来创建第二个消息摘要。然后接收者把重新创建的消息摘要同原始摘要进行对比，这样即使是最小的数据或者签名变化都会检查出来。据说，数字签名为消息的内容和创建者提供了不可改变的证据。图15-4概括说明了使用数字签名来保护消息的过程。

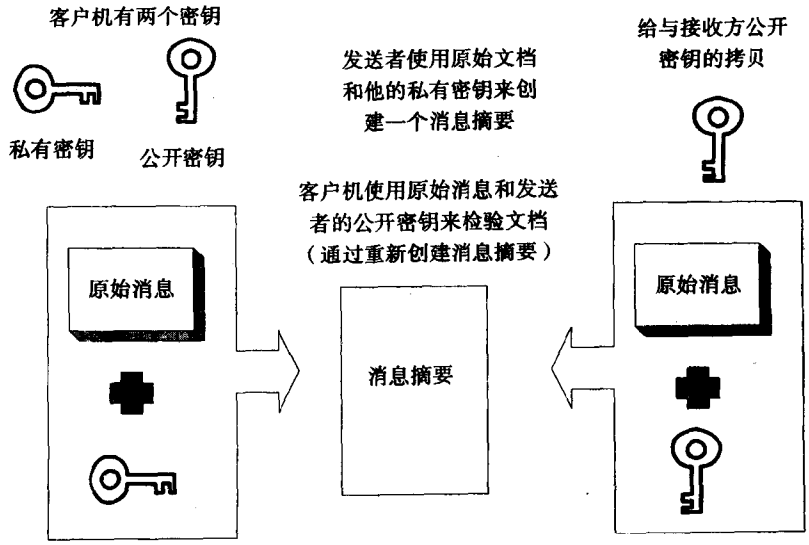


图15-4 数字签名过程

如果原始消息必须要保持机密，那么发送者可以使用一个对消息独特的私有密钥对原始消息进行加密。接着发送者使用接收者的公开密钥对该密钥进行加密，而接收者可以使用他的私有密钥来提取该密钥。这种方法看起来很复杂，那么为什么不只使用公开密钥来加密和解密整个消息呢？使用私有密钥方法（如DES加密）的原因在于：使用公开密钥来加密和解密文档要比使用私有密钥慢数个数量级。

最近，商务部国家标准与技术研究所（Department of Commerce's National Institute of Standards and Technology）发布了FIPS标准186。该文档提出了一个用于数字签名的新标准，称做数字签名标准（Digital Signature Standard，DSS）。DSS使用了一个称做数字签名算法（Digital Signature Algorithm，DSA）的新散列算法。

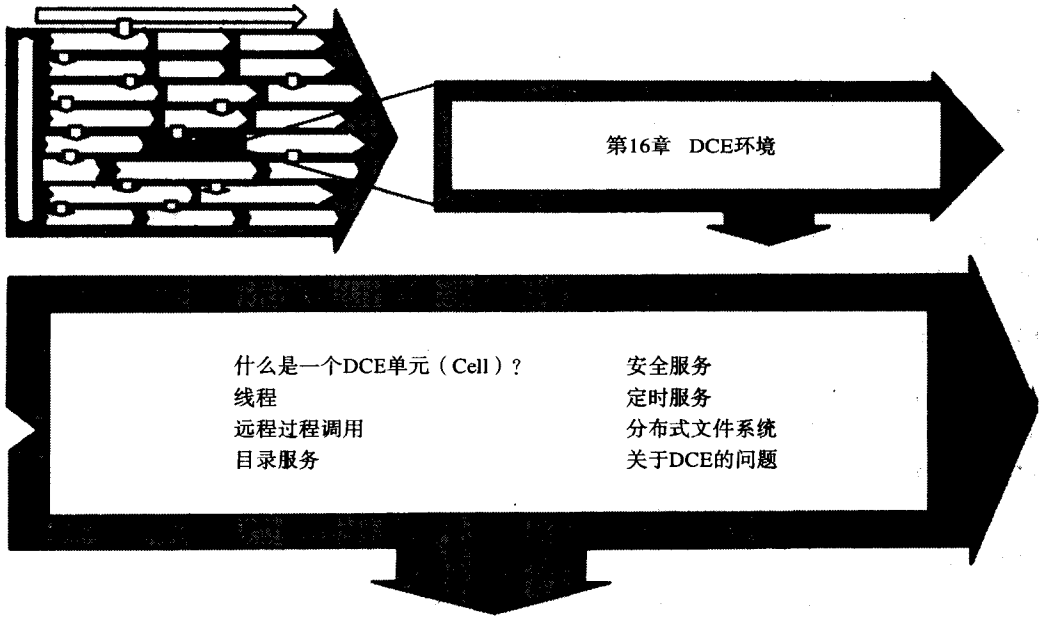
数字签名的主要应用领域是电子邮件、文档提交、公共网络上的电子商务以及电子数据交换（Electonic Data Interchange，EDI）。预计还会有许多其他的应用。可以预见，基本上任何需要不可改变的信息认证的应用都会使用该技术。只要你信任私有密钥的保密性，这就会发生。问题是怎么说服法院？

15.5 小结

加密是在计算系统中提供机密性的最佳方法之一。如果有人获得了对计算机系统和数据的未授权访问，那么加密可以起到第二道防线的作用。如果数据加密了，那么不解密数据就没什么用处了。在安全方案中，加密常常指定为一项需求，并且有可能甚至是在机密性需求弄明白之前。在选择特定的加密技术之前，一定要彻底弄明白自己的需求。在提供机密性这一方面，私有密钥方法和公开密钥方法各有所长。因为公开密钥技术提供了支持多对多关系的能力，但又不会明显地增加管理难度，所以近来这种技术引起了广泛的注意。请记住，机密是需求，加密是方案。

加密在操作系统和网络控制上进一步保护了公司的信息资产。随着商业和建议解决方案日益增长的可用性，特别是公开密钥技术的发展，使用加密来增强安全性会变得越来越普遍。如果密钥管理问题能够得到进一步解决的话，加密技术会成为计算安全世界中的一把利器。

第16章 DCE 环 境



在前面几章里，已经讨论围绕分布式环境中信任的关键问题。在LAN上以明文形式传送的密码是很容易截获和发现的。通过改变网络地址，基于网络标识的认证可以很容易伪造。客户机对应用服务器只进行很少的认证或者根本不进行任何认证，这使得攻击者可以假冒服务提供者。使用加密形式在LAN上传输数据并不是大多数操作系统所提供的一个标准功能。大多数的操作系统和应用程序都采用中心主机法进行认证，这使得用户必须要记住很多用户ID和密码。最后，把安全从系统管理员的责任中分离出来是非常困难的。

16.1 DCE的概念

前面那些问题已经伴随我们很多年了。开放软件基金会（Open Software Foundation，OSF）的分布式计算环境（Distributed Computing Environment，DCE）是解决这些问题（和其他问题）的一个办法。DCE为在多厂商环境中建立安全的、分布式的应用程序提供了工具和框架。当前，DCE在很多不同硬件平台上都受到支持，这包括Bull、SUN、DEC、HP和IBM。许多操作系统都支持DCE，其中包括DOS/Windows、Windows/NT、OS/2、UNIX、VMS以及MVS。

OSF/DCE环境是由许多相关的但却独立的组件或服务组成的。它的体系结构是非常灵活的，它使得应用开发人员可以在OSF/DCE的实现中非常方便。OSF/DCE环境包括安全服务、时间服务、目录服务和分布式文件服务。其他的服务在将来实现。图16-1说明了OSF/DCE环境的体系结构组件。

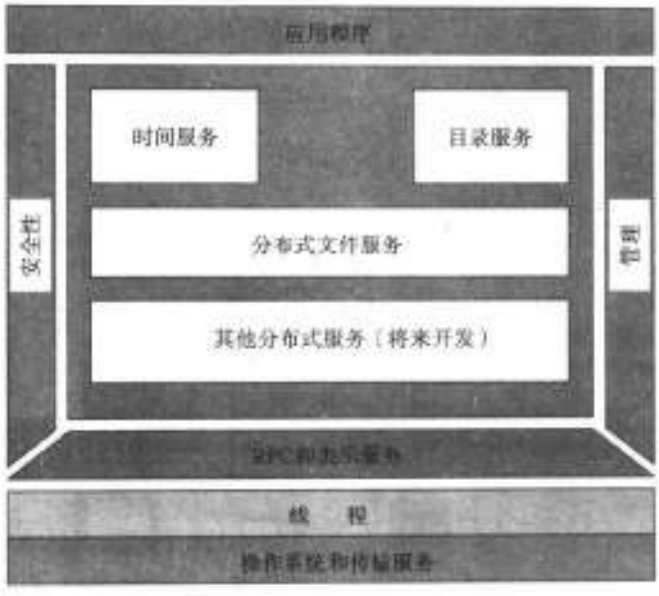


图16-1 OSF/DCE的体系结构视图

OSF DCE提供了一个重要的中间件服务框架（在网络、操作系统和应用程序之间），它以一个安全的方式把客户机和服务器结合起来。所有独立组件都是互相关联的。例如，时间服务是使用安全服务来认证客户机的。时间服务为安全服务提供了对DCE凭证到期时间的定义。

DCE服务是在一个称做DCE单元（cell）的管理域中提供的。在继续探讨之前，我们必须先要掌握DCE的一个要点。尽管DCE提供了一套相关服务，但是没有哪一个服务是必需的。例如，一个DCE单元可以选择不使用分布式文件服务。这完全由单元设计者来考虑，但是安全服务是必须使用的。每个独立服务也可以以不同的方式来实现。极大的灵活性导致了多种多样的客户DCE应用程序。至于一个特定组织应该如何实现DCE，这是没有任何限制规则的。在介绍每个DCE服务之前，我们先看一下DCE单元的组成。

16.1.1 DCE单元的概念

DCE单元的组成和结构曾经是最引人注目的客户讨论之一。什么是DCE单元？对于这个问题，随便的答案是“你想它是什么，那它就是什么”。更明确的答案是DCE单元是共享一个公共命名空间、安全策略以及安全域内其他方面的一组联网主机和服务。为了演示目的，功能完全的DCE单元可以在一个计算机系统上实现。DCE单元可以由一个到数百个计算机系统（或者节点）组成。图16-2说明了典型的具有几百个客户机并提供多种服务的DCE单元。

单元通常根据组织或者公司的结构来定义。对于DCE单元的大小的限制是DCE系统管理员管理单元的能力。另一个限制是，如果一个DCE单元要跨过WAN连接，那么为了性能原因应该分割该单元。一个单元内的节点数量通常是由上面两个因素中的一个或两个所限定的。如果能够提高WAN性能并改进DCE管理工具，那么DCE单元的大小限制就会减少。下面探讨一下DCE的各个独立组件。首先从提供多任务能力的组件——线程——开始。

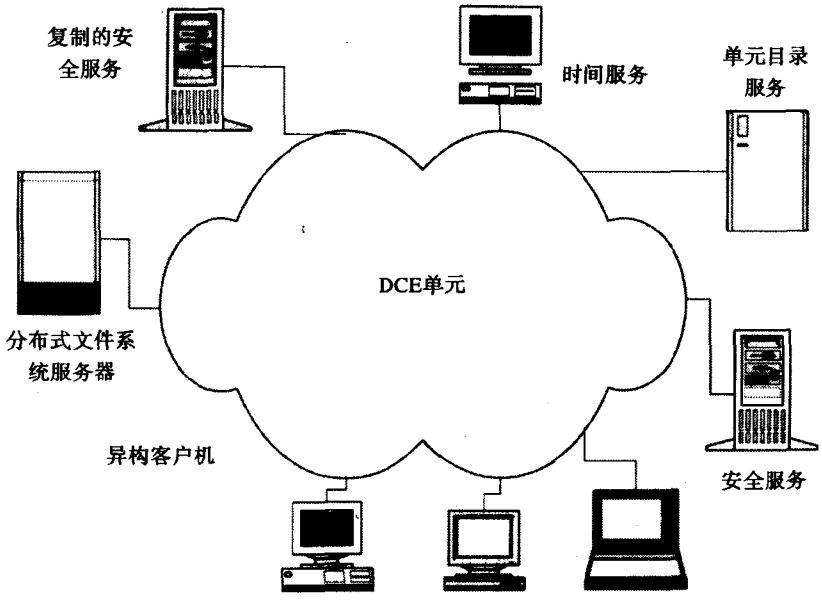


图16-2 典型的DCE单元

16.1.2 线程

线程允许DCE程序可以同时服务于多个客户机请求。来自客户机的一个服务请求可能正在等待一个资源的释放。在此期间，DCE线程可以对另外的请求提供服务或者临时进行服务器例行维护。

一个多线程程序支持几个程序执行上下文。多个线程存在于一个进程内，并一直执行到它们终止。多个线程不仅仅可以同时执行，而且还可以共享内存地址空间、外部数据以及程序变量。因为线程互相之间共享信息，所以每个线程必须要知道其他线程，并且绝对不能执行会为其他线程带来负面影响的代码。例如，一个线程的删除绝对不能破坏其他“活动的”线程所使用的信息。保护线程使用数据的完整性的一套库调用术语化为“线程安全”。

线程最常见的用途是使服务器可以同时响应来自不同客户机的多个接入请求。“线程化”方法使服务器可以通过创建线程来响应接入客户机请求。然后，客户机直接同新创建的线程进行通信。服务器进程继续执行监听新客户请求的任务。当客户请求处理完成以后，客户机对应的线程就删除。这同传统的UNIX守护进程（创建一个新进程来响应客户机请求）类似。然而，创建一个线程要比创建一个新进程快得多。

16.1.3 远程过程调用

在OSF/DCE中，远程过程调用（Remote Procedure Call——RPC）机制是把环境结合在一起的“粘结剂”。一个DCE RPC实际上是一个服务器代码，在一个同客户机的调用例程所分离的地址空间内执行的过程。尽管一个DCE RPC可以在单独一台机器上执行，但是RPC最常出现在驻留于通过网络连接的独立机器上的客户机和服务器中。在这种情况下，信息必须要在客户机和服务器间共享。公共数据必须是客户机和服务器都可用的，并且在一个多厂商环境中必须

要达成一个对数据的公共定义。还有，联网需求和不可靠的网络传输协议必须要加以处理。

DCE RPC向程序员隐藏了网络和多厂商环境的复杂性。DCE RPC运行时库替程序员完成了如下任务：

- 为客户机定位一个适当的服务器——使用绑定（binding）进程。
- 把数据移动成在网络上共享——使用编组（marshalling）进程。
- 卸载服务器上的数据——使用反编组（unmarshalling）进程。该过程也会把数据转化成合适的服务器机器格式。
- 执行相关的服务器代码——使用入口向量进程。
- 通过从服务器上编组请求信息在网络上来回答客户机请求。这些信息会接着在客户机上被反编组。

DCE RPC运行时库方便了这些复杂的操作。开发人员只需要把注意力放在客户机-服务器接口的定义上。RPC运行时会处理实际的通信。在DCE术语中，服务器所使用的数据和操作是由接口定义的。一般情况下，一个应用程序或者服务有许多接口，这取决于要回答的请求类型。

创建接口的第一步是创建一个全局惟一的标识符（Universal Unique Identifier，UUID）。该UUID是使用uuidgen特殊工具生成的一个32字符的16进制数。创建UUID会用到机器的LAN硬件地址和当前时间，这样能够为每个DCE接口生成惟一的UUID。所有的DCE接口，包括那些由时间和安全服务所使用的接口在内，都有自己惟一的UUID。

接口是通过使用接口定义语言（Interface Definition Language，IDL）特殊语言来创建的，这对客户机和服务器来说是一样的。IDL既包含了用于接口的UUID号，也定义了要编组的输入和输出数据。本地数据定义（用于具体的客户机或服务器配置）是由属性配置文件（attribute configuration file，ACL）特殊功能来提供的。

16.1.4 目录服务

目录服务的目的在于辅助DCE单元中的客户机查找和连接资源。资源可以包括应用程序服务器、打印服务或者数据文件。目录服务不是必须使用的。客户机可以不通过一个目录服务而直接连接到服务器上，但是使用目录服务可以带来好处，即服务可以改变它的位置而不会损害它的客户。

目录服务有三个主要的组件，全局目录服务（Global Directory Service，GDS）、单元目录服务（Cell Directory Service，CDS）以及全局目录代理（Global Directory Agent，GDA）。本地单元之外的名字解析是由GDS或者因特网域名系统（Domain Name Service，DNS）提供的。GDS是建立在X.500目录标准上的，它能够确定出当前单元之外的资源的位置。X.500标准是由CCITT X.500和ISO 9594标准规定的。DNS是一个广泛使用的全局名字服务，当前它是用于在因特网上定位计算机的事实标准。

CDS保存了位于DCE单元内的资源的名字和属性，它为这些资源提供了一个中心查询工具。客户机不仅仅使用CDS来查找本地应用程序服务器，而且还要定位DCE服务器（如时间服务器和安全服务器）。当一个服务需要有多个拷贝时，通过允许每个拷贝使用同一个名字注册在它的位置上，CDS方便了冗余服务的实现。客户机可以根据名字和其他属性来请求一项服务。GDS会将它们定向到一个合适的功能服务提供者。

GDA在CDS和两个名字解析服务（DNS和GDS）之间提供了一个接口。图16-3提供了一

个单元目录服务概图。

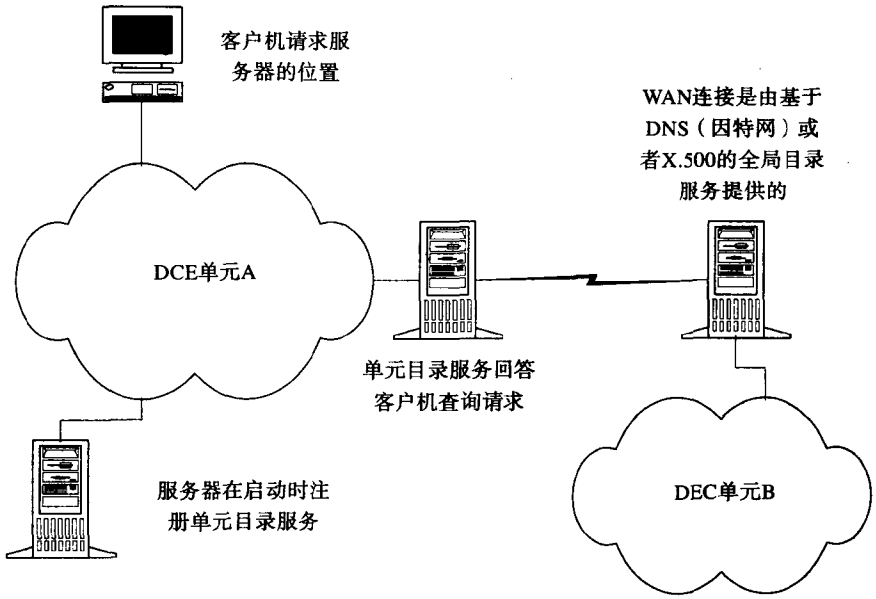


图16-3 DCE目录服务概图

客户机对目录服务的请求会转到称做CDS负责者的本地代理上，它运行在客户机计算机上。CDS负责者保持了一个查询请求缓存。如果缓存中包含了所请求的信息，那么就会立即满足请求。如果没有包含，那么CDS负责者就会联系CDS服务器。

16.1.5 安全服务

DCE安全服务为许多TCP/IP连网问题提供了答案，例如第7章中介绍的密码保护和欺骗问题。DCE假定网络是不安全的，其上的任何信息都可能会遭到未经授权的检查或修改。未经认证的客户机或者服务器标识是不可信任的。DCE假定客户机或者服务器可能是假冒的。

DCE安全一般提供：

- 每个单元内的客户机和服务器的注册（注册服务）。
- 客户机和服务器的安全认证——使用Kerberos认证模型（认证服务）。
- 消息保护选项——有助于保证在不可信网络上传输的数据的完整性和机密性。
- 访问控制列表——允许服务器授权对其资源的访问。
- 安全服务和管理是同其他单元管理功能隔离的。

将在下一章中详细探讨DCE安全服务。

16.1.6 定时服务

分布式计算中的许多关键功能都需要有一个精确的时间表示。计算机需要在很多关键功能上使用时间，这包括事务的协调和准确审计跟踪的维护。但是一个价值50 000美元的工作站和一个10块钱的手表比起来，哪个走时更准？答案是后者——10块钱的手表！计算机走时不准的原因在于它们是以无规律的速率来调整其时钟的，所以其时钟同真正时间比起来是忽

快忽慢。这种同真正时间的偏差术语化为时钟偏移。如果每个计算机都走时不准，那么试想一下一千台计算机连起网来会出现什么定时问题！

DCE分布时间服务（Distributed Time Service，DTS）在ISO 8601时间标准的基础上为该问题提供了一个解决方案。它使每台计算机可以从DTS服务器处得到一个统一的时间表示。通过使用多台DTS服务器，提供了冗余性，客户机通过查询这些服务器来找到一个公共时间间隔。每台DTS服务器不断地同其他服务器进行通信以同步它自己的时钟。

DTS并不是为每个客户机提供一个绝对的时间定义，而是保证所有客户机都把时钟设置到一个预定义的时间间隔内。如果认为系统太慢，那么它就会把时钟调快到标准时间；但是一个时间超前的时钟并不会调整回去，而是会降低速度直至其时间接近预定义的时间间隔。

时间定义可以来自一个通用协调时间（Coordinated Universal Time，UTC）提供者，比如原子时钟、政府时间或者无线电台。另外，也可以使用网络时间协议（Network Time Protocol，NTP），或者把一台计算机选做单元的真实时间。在这种情况下，所有的计算机都要以一个同样的速度同真实时间看齐！

DTS要让自己即通过安全服务又通过客户机系统的认证。因为DCE中的安全凭证对时间是敏感的。如果不要求DCE认证自己，那么一个假冒的时间提供者就可以把系统时钟调快或者调慢。这会使得冒牌者可以把过期凭证作为有效凭证来使用，或者把现有的所有安全凭证都做废。

16.1.7 分布式文件系统

分布式文件系统（Distributed File System，DFS）提供了对数据（如文件）进行分布式访问和管理的能力，如图16-4所示。DFS允许用户可以透明地访问分布式文件而无需考虑用户或者所请求资源的物理位置。DFS是建立在全局命名约定上的。使用这种约定，访问DFS文件是使用一个名字来进行的，它无须考虑所访问的分布式系统具体位于何处。例如，单元Polaris中的example.txt文件可以被Polaris.canada.hp.com单元之内或者之外的任何DCE客户引用做/usr/example.txt。

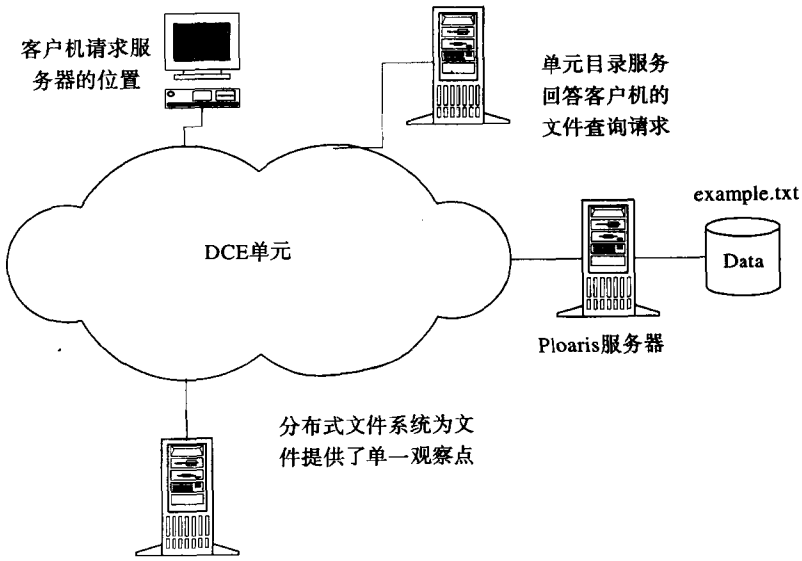


图16-4 DCE分布式文件服务

通过复制DFS服务器和资源，DFS使得其数据和服务具有高度可用性。复制使得一个文件可以有多个拷贝存在于分布式服务器上。如果主服务器出现故障，那么客户机可以透明地访问替代服务器以得到文件。这种类型的复制对于不断被改变的和由多个用户访问的文件来说是特别有用的。缓冲技术也用来提供冗余性。它允许文件的多个拷贝存放在DFS客户机上，这样即使客户机暂时同服务器断开连接，用户也可以访问文件的拷贝。对DFS文件进行备份和重定位都不会使文件变成用户不可用的。DFS也有能力同其他类型的网络文件系统进行交互。通过DFS到NFS安全网关，惠普公司提供了对安全使用网络文件系统（NFS）的支持。

16.2 关于DCE的问题

我们相信DCE为在部署分布式应用程序时所遇到的很多问题提供了解决方案。近几年来，对DCE环境的支持不断增加，其中包括主要数据厂商提供的一些最新产品。然而，在向DCE环境迁移这一方面，计算群体已经表示出了一些勉强之意。DCE所具有的一些缺点正在阻止它成为普遍接受的中间件技术。用户的这些不情愿是基于下面这些考虑的：

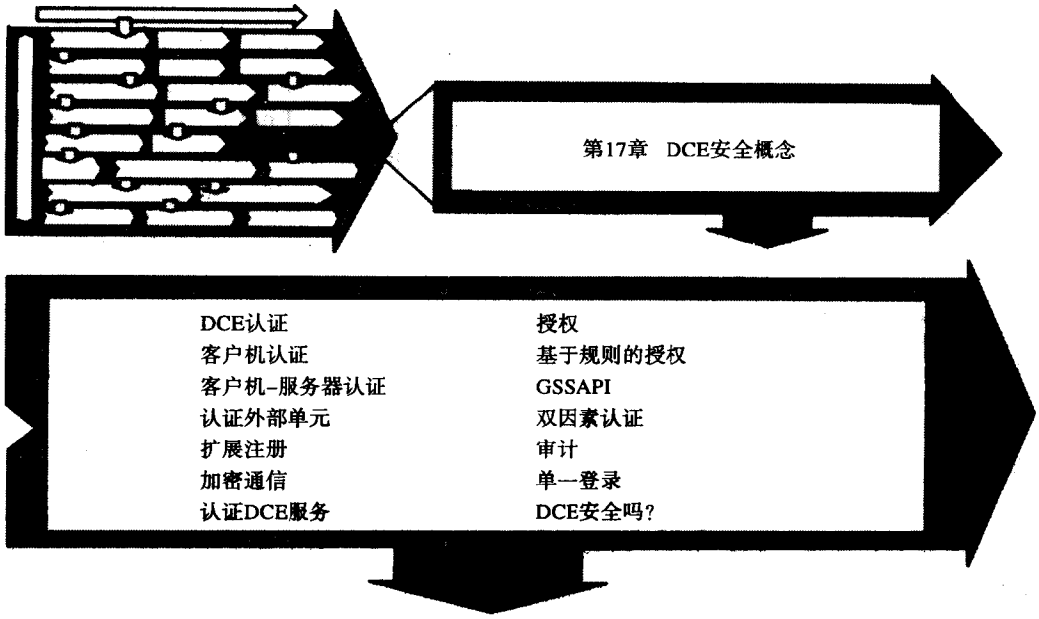
- 环境太复杂，难以编程。
- 因为会增大流量，所以有些网络可能会性能下降。
- 当前的管理工具是命令行式的，它们需要同基于GUI的网络管理工具结合起来。
- 主流应用程序厂商对OSF/DCE的支持仍很有限。
- 面向对象标准委员会对OSF/DCE的支持不够。
- 担心DCE不会获得足够的推动力从而成为一个普遍接受的技术。

尽管对OSF/DCE的将来发展存在着这些顾及，但也有一些积极的开发工作。世界上许多大的金融组织现在正在部署使用该技术的应用。OSF/DCE是否会得到足够的推动并成为一个普遍接受的技术，这还有待时间来证明。

16.3 结论

现在预测OSF/DCE的将来是不可能的。很明显，DCE需要更大范围的承诺，特别是来自DOS/Windows和应用程序解决方案领域的承诺。但是如果DCE不是能够解决所涉及到的问题的方案，那会如何呢？现在，DCE的替代物看起来要以粉碎性的手段来攻击这些问题，它似乎要以一个不同的、不相关的技术来解决每一个问题。在下一章中，将深入探讨DCE的安全机制。

第17章 DCE安全概念



分布式计算中的一个中心问题是如何使用一个“不可信的”网络？当密码以明文形式在LAN上传输时，我们如何能够信任安全机制？当分布式授权控制依赖于容易伪造的网络地址时，应该在这些控制上施以什么样的信任？这些问题的答案让人难以置信。

DCE的认证机制是基于Kerberos认证模型的，该模型是由麻省理工学院（MIT）在20世纪80年代后期开发的。Kerberos的名字非常古怪，它假定网络是不可信的。Kerberos使用多种密钥技术的组合来使客户机和服务器的相互认证。

> Kerberos又称Cerberus是长着三个脑袋的地狱守门狗。Cerberus使用它的尾巴和耳朵把人骗近地狱中。Cerberus是可以击败的。大力神在11岁时战胜了Kerberos，通过了最困难的考验。

在本章中，将探讨Kerberos模型的DCE实现（它是一个增强版），并介绍其他的DCE安全机制（例如访问控制列表）。为了让初学者明白DCE安全的复杂性，我们已经花了一些篇幅来讨论DCE安全性。我们不可能详细地介绍DCE所使用的全部机制。而是力图提供对DCE概念的一个好的概述，并帮助读者基本掌握DCE安全的工作机制。

17.1 DCE认证

DCE认证的设计宗旨是能够提供下面这些假设下的安全认证：

- 如果不加密，网络包会受到探查。会探查包含明文密码的包，会发现“保密的密码”。

- 可能会截获和分析包含认证信息的包，并在将来的攻击中重发。
- 网络地址不可信任（这包括IP地址和网卡上的硬件地址）。
- 不仅仅需要认证客户机，也需要认证服务器。

DCE认证处理把包加密、基于时间的认证凭证和一个“可信第三方”结合起来使用以提供安全认证。通过把授权和Kerberos中没有的其他安全控制结合进来，DCE增强了Kerberos认证模型。在DCE术语中，参与者指的是其标识可证明的任何东西或者任何人。参与者可以是个用户、应用程序、计算机、DCE服务或者DCE单元。一个认证凭证是一个电子证书——如果提供了的话，它可以证明参与者的标识。当发送到网络上时，密码和对话密钥是加密过的，但真正的包信息并不会加密。对话（或者会话）密钥是由系统生成的随机数，它用来保护参与双方的通信。通过认证以后，用户和服务器就会被给与时间有关的凭证，这些凭证可以抵抗重发攻击。DCE安全是基于Kerberos模型的，但是通过支持另外的安全属性而扩展了该模型。这允许服务提供者不仅可以检查客户的标识，而且还可以根据其他属性来授权访问。这些附加的属性包括了组标识和其他任意的属性。可信第三方——DCE安全服务——用来进行客户机和服务器的相互认证。DCE安全服务的每个用户，不管是人、计算机还是应用程序，都称做参与者。

> 可以把DCE使用的相互认证解释为“第一天见面”。假设有两个人都对对方有兴趣，但是这两个人都有点害羞，那么他们可以找一个共同的朋友来一起玩一下。虽然他们二人都对对方有点了解，但是让一个共同的朋友来开场介绍是最好不过了。看电影或者吃晚饭都是很好的选择。如果一切进展顺利，那么这个共同的朋友就可以走开了，然后这两个新朋友就可以私下里联系了。

DCE安全服务有三个组件服务，认证、特权和注册服务。认证服务的任务是在开始的时候验证参与者的标识。如果用户提供了一个有效密码，那么认证服务会为用户提供证明授予证明（Ticket Granting Ticket，TGT）证书。TGT使得用户可以向应用服务器的证明提出附加请求。特权服务为附加凭证中的参与者封装了附加的安全属性（特权），该附加证书称做扩展特权属性证书（Extended Privilege Attribute Certificate，EPAC）。应用程序可以检查客户机提供的EPAC，并且根据其内容来决定是否授予客户机访问权。注册服务是关于DCE单元中的参与者、组和账号的信息的一个存储仓库。它也包含了用于DCE单元的安全策略。图17-1说明了DCE安全处理的组织。

DCE安全服务为整个DCE单元维护了用户、服务器以及安全策略的数据库。安全数据及其相关接口的集合称做注册服务。一个从用户的保密密码中生成的密钥用来标识用户，并且由注册服务来维护。任何DCE应用程序，不管其在DCE单元中的位置如何，都可以对客户机的凭证进行认证。

17.1.1 客户机认证

客户机使用DCE工具（通常是dce_login）或者执行DCE登录的操作系统登录工具来获得DCE凭证。登录程序通过联系单元目录服务（Cell Diretory Service，CDS）来确定可用的安全服务器的位置。DCE的早期版本使用了一个（相对）很简单的密钥交换机制来认证用户。在OSF RFC 26.0中，惠普公司的Joe Pato讨论了很多关于使用密码猜测技术来攻击Kerberos模

型的问题。问题在于Kerberos和DCE凭证都可以在LAN上截获，或者被窃取以进行密码破解。Pato给出了一些建议技术，这些技术可以避免在LAN上发送使用用户的私有密钥加密的信息。一种称做第三方预认证的技术引入了一个存储在客户机上的新会话密钥。这种方法可以防止攻击者从加密过的凭证中猜测出用户密码。第三方预认证技术已经纳入OSF/DCE 1.1版中。

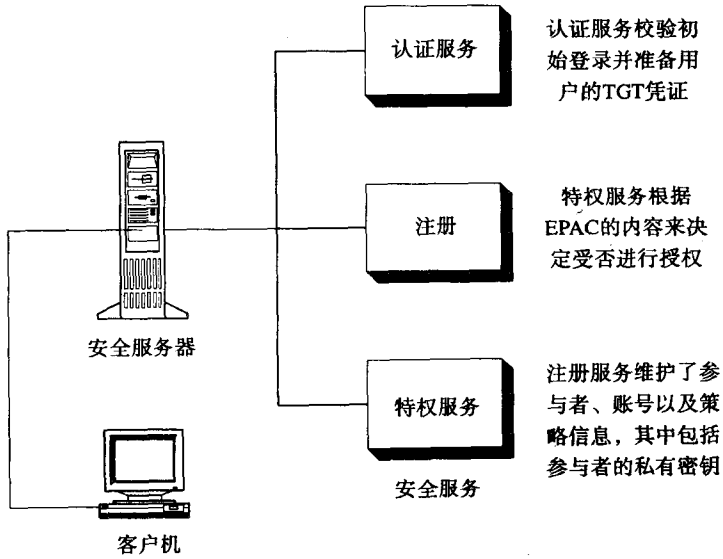


图17-1 OSF/DCE安全概览

客户机联系本地机器上的DCE安全服务（实际上它是本地DCE安全库）。本地机器有自己的TGT（称做机器TGT）和会话密钥，它们是在DCE安全服务初始化时获得的。用户并不会直接得到TGT，而是由本地安全服务代表用户获得TGT。因为本地安全服务和安全服务器共享公共密钥，所以用户的密码可以转化成一个私有密钥。用户密码不会传输到安全服务器，而从密码中创建的安全密钥会交换。本地安全服务也会生成对话密钥。这些密钥会随同TGT一起转发到DCE安全服务，并且会使用用户的私有密钥进行加密。整个包受到多级加密的保护。采用这种设计，流量受到了保护从而免遭密码探查或者重发攻击。

图17-2概括说明了DCE认证过程的初始阶段。

安全服务器使用机器保密密钥（它知道）来得到附加对话密钥并解密包。如果时间戳是在可接受的范围之内，安全服务器将为客户机创建TGT。该TGT包含了客户机的基本信息，并使用一个只有安全服务器才知道的密钥密封起来（例如加密）。安全服务器也会创建一个客户机对话密钥，客户机将使用该密钥同安全服务器进行后面的对话。密封的TGT和客户机对话密钥都置于一个包中，接着使用该包一个从本地安全服务处得到的对话密钥进行加密。然后该包（包括了密封的客户机TGT和对话密钥）返回到本地机器。本地机器接收加密过的包，使用它提供的对话密钥来解密该包，并存储客户机的TGT和对话密钥。这种方法有如下优点：

- 只有本地安全服务才能开启客户机的TGT和对话密钥，这是因为它们是使用最初由本地安全服务提供的一个对话密钥来加密的。
- 只有一个有效的安全服务才能提供TGT包，这是因为包是由本地机器的会话密钥（从安

全服务处得到）加密的。客户机和安全服务器因而互相保证了对方的标识。

- 用户的密码和密钥只需要在内存中保持很短的时间。这样能够防止攻击者使用内存探测技术来探查客户机系统上的密码或者密钥。

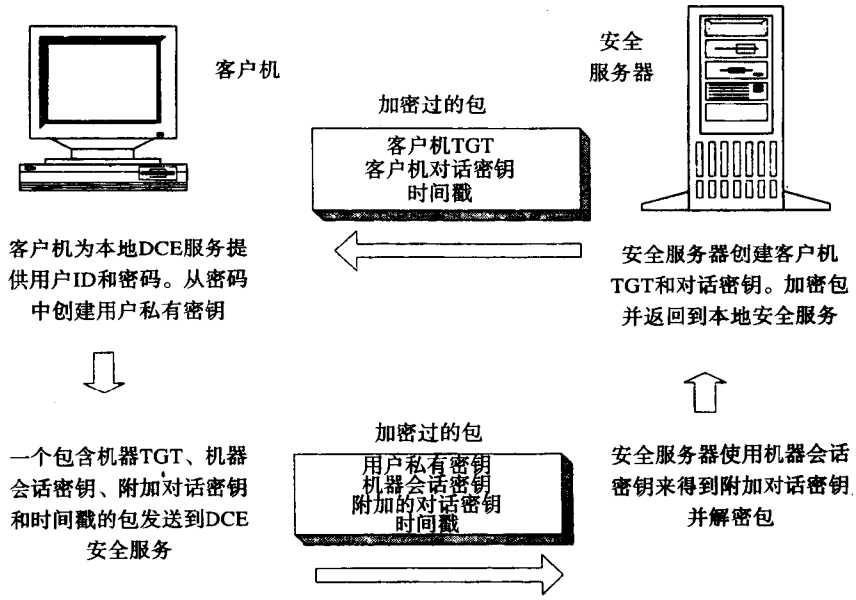


图17-2 对客户机的初始认证阶段

TGT的生命期是有限的，安全管理员可以对其进行配置，其默认值是10个小时。10个小时过后，客户机必须要通过重新启动登录程序来获得一套新的凭证。PTGT通常限制在2个小时之内。大多数客户机可以通过在10个小时以后重新登录来更新其基础凭证，但是还有一些其他方法可以让客户机自动更新其凭证。当客户机递交批任务的时候，这种技术是特别有用的。

应用程序和机器服务证明（同客户机的TGT相对）是自动更新的，这不需要任何人工干预。使用很多对话密钥可能看似过于复杂，但是这种做法确实有一些优点：

- 相对来说，对话密钥和凭证的存在时间是很短的。如果黑客要发现密钥，那么他必须要在很短的时间内完成。
- 攻击者必须要发现多个密钥。

应该指出，密钥和凭证认证的复杂性是由DCE运行时层而不是应用程序员来处理的。

客户机下一步要得到附加凭证——扩展特权属性证书（Extended Privilege Attribute Certificate，EPAC）。EPAC包含了更多的客户机信息，例如组成员和地点、客户机的特殊安全属性。EPAC是对原始的DCE特权属性证书（Privilege Attribute Certificate，PAC）的一个改进。在分布式环境中，一个服务器代表一个客户机向第二个服务器提出请求是很常见的。然而，如果要这么做，那么必须要解决一个基本问题，第二个服务器如何能够识别出来自第一个服务器的请求是代表一个客户机的？惠普Chelmsford实验室的Joe Pato为OSF RFC 3.0中的实用扩展委托提供了一个建议模型。该提议方案把服务器的特权属性证书扩展为结合客户机凭证。当前的ACL模型也重新设计成检查扩展的PAC。OSF/DCE在DCE 1.1版中引入了EPAC。

客户机首先把一个PTGT请求转发到特权服务。特权证明授予证明（Privileged Ticket Granting Ticket，PTGT）允许客户机从特权服务处请求特权服务证明。获得初始PTGT和EPAC的过程同初始认证请求类似，其原因在于对话是通过客户机TGT和会话密钥的一次交换来保护的。

客户机并不直接同服务通信，而是通过中间者来获得必要的凭证。关于整个过程的详细说明，读者可以参考发表在惠普期刊上的《DCE Seurity Service》一文（[Gittler & Hopins,1995]）。

17.1.2 客户机到服务器认证

当一个客户机希望联系一个服务器时，该客户机会在单元目录服务中查询该服务器。服务器和客户机之间的相互认证是使用一套新的对话密钥来完成的。在深入探讨之前，先介绍三个重要概念。

首先，应用程序服务器根本不需要执行认证过的RPC。确定合适级别的认证和所需安全是应用程序设计者的任务。第二，服务器实施一套最小标准，但是客户机可以请求使用更强大的安全机制。只有在客户机不符合最小标准的时候，服务器才拒绝客户机。本质上，客户机可以同服务器自由协商一个更高的标准。第三，不符合最小标准的客户机不会自动拒绝。应用程序服务器必须要明确检查客户机的请求，如果请求不符合最小标准，则拒绝该请求。客户机必须要预先知道服务器的最小需求，这是因为这些需求是不能直接从服务器上得到的。

客户机请求起始于对所需安全级别的声明。请求提交到认证服务处以获得一个对所需应用程序的证明。原始EPAC转发到安全服务处，并且使用应用程序的保密密钥来重新加密。这样能向应用程序服务器证明客户机的EPAC是有效的。该过程是通过交换对话密钥来保护的。设计者尽可能地把复杂性向应用程序员和用户隐藏起来。应用程序证明通常受到TGT和PTGT已有生命期的限制。

> OSF/DCE认证和国际旅行控制之间有很多相似之处。TGT和EPAC类似于护照。它们都有一个有效期限制，并且都使用了难以伪造的技术（纸张、邮票和封印）。服务器可以比做国家。有些国家允许无护照入境（例如加拿大人到美国旅行或者在EEC（欧洲经济共同体）成员国之间旅行）。大多数国家都要求入境旅游者持有效护照。另外，有的国家可能需要签证。签证的有效期要比护照的更短，通常护照过期它就会过期。可以同签证类比的是应用程序证明——应用程序服务器可能需要它作为另外的证明。

17.1.3 认证外部单元

如果用户从一个外部单元（如一个具有不同安全服务的单元）到达的话，那会如何呢？应该使用本地安全服务对用户进行认证吗？或者就因为坏人不能操作DCE单元而在表面价值上接受凭证吗？通过把用户指定为经过认证的而不是来自外部的，DCE允许进行调用。至于如何对待外部用户，这是由应用程序来决定的。

惠普的Joe Pato在OSF RFC 7.0中建议了一个交叉单元认证模型。该模型允许单元管理员来决定哪些外部单元是可信的哪些是不可信的。从组织内部的单元中生成的证明对于一个单元来说是可信的，但对其他单元来说不一定。这些建议已经结合到OSF/DCE 1.1版中。

17.1.4 扩展注册

DCE是偏离UNIX的，这是它的一个争议之处。在OSF RFC 6.0中，Joe Pato建议了一个更灵活的注册服务设计方案。该方案允许使用动态属性，这些属性对于很多的操作系统和应用程序来说都是合适的。该RFC中的思想已经结合到扩展注册属性（Extended Registry Attribute，ERA）功能中。ERA功能允许新属性的引入。这使得DCE对于非UNIX环境来说非常有吸引力，并且提高了DCE的可接受程度。

17.1.5 服务器认证

在对客户机认证机制的讨论中，我们隐含地假定应用服务器已经认证了。客户机和安全服务之间所进行的相互认证也要在应用服务器和安全服务之间进行。后面双方的认证同客户机到安全服务的认证方法相同，但是有几个重要的例外情况。应用服务器一般是长时间运行的，因此它们不能每隔八个小时就重新初始化登录过程。在到期时间之前，服务器使用替代方法来更新其凭证和对话密钥。应用程序不能等系统管理员在启动时把一个密码作为密钥。服务器的保密密钥密码可以保存在多个地方，但一般是以加密格式保存在keytab文件中。该keytab文件是由本地文件系统资源保护的。

17.1.6 加密客户机-服务器通信

DCE支持多种加密级别以保护客户机和服务器之间的通信。前面已经提到过，客户机可以同服务器协商一个需要的保护级别，只要该级别符合或者超过服务器实施的级别即可。表17-1列出了可用的保护级别。

表17-1 对客户机-服务器通信的保护

保护类型	说 明
默认	使用DCE单元的默认级别
无	不使用任何保护，从而也不需要Kerberos认证
连接	客户机和服务器的标识是在初始连接时认证的
调用	使用一个对话密钥来加密每一个RPC调用。独立数据包不加密
包	每个数据包都附加一个加密过的字段
包完整性	独立包有一个加密过的校验和以防篡改
完全加密	客户机和服务器之间的所有通信都完全加密

即使对于最不安全的应用，最小保护级别也是调用级保护。更低级的保护容易受到包篡改攻击。大多数在乎安全性的应用程序都是使用包完整性级或者完全加密级保护。完全加密受到了美国国防部的加密技术出口限制（关于这方面的详细信息，请参考第15章“密码学”）。应该指出，完全加密选项只加密LAN上包的数据部分。包括源地址和目的地址在内的TCP/IP地址和传送信息都不加密，可以使用包探测技术来查看这些内容。然而，真正的数据是加密过的，因而是不可理解的。一旦确定了通信保护级别以后，会话密钥的交换、数据的加密和解密就都由DCE来处理，这对于客户机和服务器来说是透明的。

17.1.7 认证DCE服务

DCE的一个重要特性是独立DCE服务，如定时服务和单元目录服务，也是由安全服务来

认证的。这使得攻击者很难假冒任何DCE服务提供者。

> 然而，向DCE单元提供服务的非DCE应用程序是被取笑的对象。我们曾经听说过这么一种情况，一个使用了因特网NTP时间服务的DCE单元遭到了攻击，攻击者把时间设置成下个世纪，从而使得当前所有凭证全都失效！

17.2 授权

DCE使用访问控制列表机制来对单元内资源的使用情况进行授权。ACL的使用是非常灵活的——事实上，它们是相对无组织的并且由独立应用程序来实现。ACL可以用来限制对文件、目录、硬件设备甚至应用程序服务器本身的访问。

在DCE的早期版本中，应用程序开发人员需要定义、管理和实施ACL。独立ACL一般保存在平面文件中，由服务器上的本地文件系统来保护。但实际上应该由独立应用程序来决定把ACL保存在什么地方以及如何使用它们。DCE 1.1版提供了管理ACL的编程库和工具。

ACL是同访问权限一起建立的。定义自己的访问权限是独立应用程序的任务，表17-2列出了一些访问权限示例。

表17-2 典型的ACL访问权限

访问权限	说 明
Read	客户机可以打开和检查资源的内容，但不允许修改内容
Write	客户机可以修改资源
Control	客户机可以控制资源
Insert	客户机可以创建新资源，例如新文件或目录
Delete	客户机可以删除已经存在的资源

可以通过把客户机的分类（例如“任何其他”）和一个适当的访问特权（例如“无”）结合起来以创建一个ACL。ACL可以应用到独立用户上，但是一般都是应用到一类或者一组用户上。定义分类是由应用程序所控制的，表17-3列出了一些典型的客户机分类。

表17-3 典型的ACL客户机分类

拥有者分类	说 明
拥有者	客户机当前是资源的拥有者。只有一个拥有者
特殊用户	客户机通过认证，并被特殊标识
外部用户	客户机被特殊标识，但它是外部DCE单元中的成员
组	客户机是资源所属组中的成员
特殊组	客户机是特殊组中的成员。可能有多个标识的组
其他	客户机是本地DCE单元中的认证用户，但不属于上面各种情况
外部其他	客户机已经在另一个单元中通过认证
任何其他	除上面各种情况之外的分类

17.2.1 基于规则的授权

当前ACL的管理是非常笨重的，这是它的一个问题。从前面可知，管理ACL是开发人员的职责。当你希望结合使用ACL的时候，管理问题会变成一个大麻烦。我们可能想建立一些

有关ACL如何工作的规则，而不仅仅是简单的文件访问列表。例如，如果你是一个信用部经理，那么也许能够批准最大500 000美元的客户贷款。希望的是，基于规则授权的管理工具会在不远的将来可用于DCE环境。

17.2.2 GSSAPI

如果我有一个客户，全套DCE都不适合于他，但我又希望该客户能够以一个安全的方式来使用DCE应用程序，那么在这种情况下应该怎么办呢？这种情况暗含着如下意思，这些客户机必须能够把它们自己认证到DCE安全服务上，并获得安全凭证。对于该问题，解决方案便是被提议的通用安全服务应用编程接口（Generic Security Service Application Programming Interface，GSSAPI）。GSSAPI不是一个绝对的安全解决方案，它允许客户机或者服务器使用一个公共的（或者通用的）方法来请求安全服务。GSSAPI位于客户机和安全机制之间，它允许应用程序员为认证和授权请求进行编程而无需考虑所使用的安全机制。GSSAPI所受到的支持正在逐渐增加，并且我们也已经看到了它在很多安全应用和产品中的使用。例如，GSSAPI对OSF/DCE和Kerberos V安全机制都支持。

对使用GSSAPI的讨论最初是由数字设备公司（DEC）的J.Linn在他于1992年提交到OSF DEC特殊因特网组织（Specical Internet Group，SIG）的OSF RFC 5.0中提出的。然后这项研究随着RFC1508和1509的提出而迅速扩大。RFC提议，非DCE客户应该被给于一套允许客户认证的接口——使用DCE共享的保密密钥（或者其他）机制。它也应该能够保护消息的机密性——使用机密和校验和。

从DCE的角度来看，GSSAPI的重要性在于它可以把DCE的强大安全能力扩展到非DCE世界。可以期待的是，GSSAPI的使用会极大的提高Kerberos和DCE的使用。OSF DCE 1.1发布版支持GSSAPI的使用。

17.2.3 双因素认证和智能卡

在RFC 59.0中，安全动态（Security Dynamics）的J.Kotanchik建议Kerberos V实现应该扩展成包括双因素认证。双因素认证是基于两套证明的，例如你所知道的某样东西（如一个密码）、你所拥有的某样东西（如一个认证设备）或者你是某样东西（生物统计学数据）。Kotanchik描述了一种可能的DCE安全实现，即把用户提供的密码和由手持设备生成的一个伪随机数（Pseudo Random Number，PRN）结合起来。

两个澳大利亚研究人员，DSTC的Gary Gaskell和Telstra的Michael Warner制定了OSF RFC 71.0，该RFC发布于1995年2月。该RFC说明了如何结合智能卡技术的使用来保护DCE安全和执行加密功能。

17.2.4 审计

早期版本的DCE没有在单元、应用程序或者服务器的基础上定义一个审计跟踪。审计标准和审计跟踪的组成留给了应用程序设计人员去解决。VDG公司的Shyh-Wei和IBM加拿大实验室的Robert Weisz已经提议了一个标准审计跟踪机制。在RFC 28.1和29.1中，他们建议应该定义一个符合C2安全需求的公共审计API和审计跟踪。提议的审计跟踪允许事件逻辑分组到事件分类中。它也允许DCE管理员有选择地增加对独立单一事件或者一类事件的审计。DCE

1.1版提供了对审计库和审计日志的支持，并增添了审计管理能力。

17.2.5 单一登录

单一登录概念是如果一个用户登录到一个网络上，那么所有其他连网系统和应用都接受这次登录。这样做的好处在于，用户只需要记住一个用户ID和密码以进行网络访问。因为OSF/DCE支持大量不同的硬件平台和操作系统，所以它是构建单一登录解决方案的合理候选者。扩展属性的使用，以及对扩展用户注册的支持，使得结合单一登录更为容易。

17.3 DCE安全吗

我们曾目睹过很多客户这么说，“如果实现了Kerberos和DCE，那我们就再也不必担心安全问题了”！这种看法是不正确的。首先，安全性不是光牵扯到技术这一方面。如果不分析整体情况——策略、程序、培训以及意识，那么不可能真正解决问题。在一个安全环境中，策略、程序、培训以及意识都是同等重要的因素。仅仅实现技术是不能解决计算机安全问题的。也就是说，DCE的安全措施需要同许多关键方面配合起来才能实现分布环境中的安全性。但DCE是一个安全解决方案吗？答案为不是！

DCE有很多脆弱点，它需要依靠客户机的操作系统来保护用户凭证。如果这些凭证被窃取，那么就有可能造出伪造的请求。同样，DCE信任服务器上的本地文件系统控制对服务器保密密码的保护。如果本地系统控制具有脆弱点，那么DCE认证程序就有可能被替换成特洛伊木马程序，从而造成用户密码的窃取。还有，除了操作系统使用的那些控制机制之外，DCE不提供任何附加的文件保护机制。它并不为系统管理员或者用户提供加密文件的能力。

只不过因为DCE有一些脆弱性，我们就什么也不用做了吗？绝对不可以！部署DCE绝对要比接受现状好！它可能不是一个银子弹，但它绝对要比军火库外的东西强。

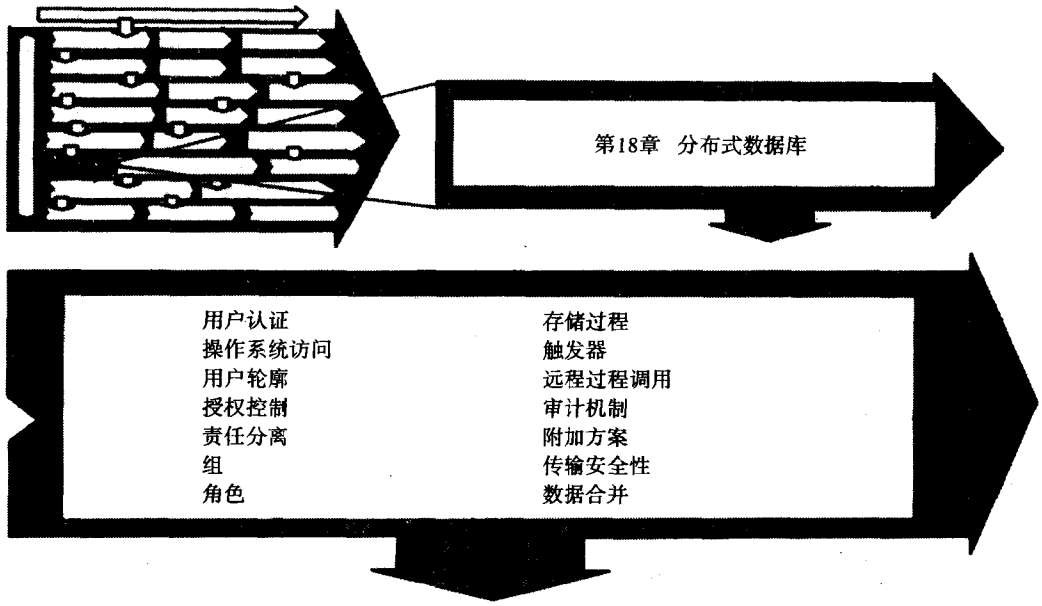
最近，很多有助于解决上面一些问题的产品已经出现了。惠普公司已经宣布了Odyssey产品，它能够安全地保存DCE凭证。Odyssey使用了GemPlus智能卡，它在释放凭证之前对用户进行认证。如果卡被窃取了，那么DCE凭证还会受到保护，这是因为用户必须知道卡上的密码才能开启它。

17.4 结论

很明显，通用计算群体不会接受现状。明文密码在LAN上的暴露、对安全认证客户机和服务器的需求以及通信的保护都会使情况发生变化。并且，对复杂网络环境中的冗余性、目录服务以及多厂商能力的需要还会继续增大。

DCE在今日市场上还没有任何明显的基于工业标准的继任者或者替代物。当前没有任何技术能够提供DCE所能提供的能力。尽管我们不能预测DCE的将来，但是对这种技术类型的需要会一直存在。我们不能再把信任置于当前的中间件技术上。

第18章 分布式数据库



本章将探讨分布式数据库解决方案的安全性问题。数据库是分布式计算的防护墙，它可以保护最有价值的个人资产——即使入侵者已经能够进入我们的房间。数据库在安全性方面具有几个重要的优点，其中包括强大的数据访问控制和审计跟踪。

同前面几章中讨论的技术一样，本章将着眼于影响分布式数据库的安全问题、分布式数据库的工作机制以及其中的一些关键技术。本章并不把重点放在任何特定的数据库实现上。相反，我们将对三大分布式数据库技术的一些共性进行探讨：Infomix、Oracle和Sybase。在掌握了数据库环境以后，将介绍所涉及到的问题的解决方案。

18.1 RDBMS的概念

关系数据库管理系统（Relational Data Base Management System，RDBMS）是基于关系型数据库模型的数据存储和检索系统。这种模型使用表作为其基本结构。表是查看相关信息的一种逻辑方式。例如，一个员工表可能包含员工号码、姓名、部门、位置以及聘用日期之类的信息。关系表由多个行和多个列组成。在一个表中，数据被组织到垂直列和相交行中，其中每个交点处只有一个数据项。行可以看成是数据记录，列是记录中的数据项。例如在一个员工表中，可能每个员工对应一行，而员工号码、姓名及其其他信息各对应一列。关于数据库用户的信息，包括密码和用户轮廓，也保存在表中。

RDBMS是由很多表组成的，它们之间互相关联。使用称做键的公共数据元素可以把这些表链接起来。为了快速查询键，也可以创建索引。视图是一个并没有物理保存的逻辑表，它

允许对一个表或者多个表中的信息的子集进行访问。例如，经理可能需要限制对工资表的访问。部门经理可以完全访问其手下员工的工资信息，但不能查看更高级管理人员的薪水。经理可以查看的信息称做一个“视图”，这些信息可能分布在同一行或者不同行中的列的许多子集上。

大多数关系型数据库都支持一个用于执行特别查询、编程和管理的共同语言。美国国家标准委员会（ANSI）和国际标准化组织（ISO）制定的结构化查询语言（Structured Query Language，SQL）提供了一套可供所有标准RDBMS实现使用的标准数据定义和数据操作语句。一个SQL数据库环境实际上是由很多标准组成的，其中包括在编程语言中使用SQL语句的嵌入式SQL（ESQL）。应该指出，用于从客户机程序中调用服务器的应用编程接口（API）的标准化工作也已开始。例如，SQL访问组（SQL access group，SAG）提供了一个公共API定义，大多数RDBMS厂商都已经采纳该定义。

交互式结构化查询语言（Interactive Structured Query Language，ISQL）是用于数据库管理和特别查询的一个工具。它允许用户直接访问数据。使用标准SQL命令，用户可以创建定制的报告并执行对数据库的一次查询。大多数RDBMS厂商都支持标准SQL的使用，但是他们为SQL和数据库环境添加了一些扩展以提供附加功能。不幸的是，各个厂商的扩展为安全问题带来了不同的解决方法。并不是每个数据库实现都提供同一级别的整体安全性。

18.2 启用应用的不同模型

有三种不同的体系结构可用于启用RDBMS应用，它们是中心服务器、受限客户机-服务器和完全客户机-服务器。在传统的中心服务器模型中，数据和应用程序逻辑都存在于数据库服务器中。本地工作站用做访问应用的智能终端。所有的程序逻辑都在服务器上的应用程序中预定义和执行。应用数据保存在服务器上的数据库中。在受限客户机-服务器模型中，客户机工作站起到访问RDBMS的智能接口的作用。它为用户提供了一个复杂的图形用户界面，但是几乎全部的应用数据和逻辑都存在于服务器上。在完全客户机-服务器模型中，数据和应用逻辑存在于客户机和服务器上。客户机维护了数据的本地表，并且有能力向数据库服务器提出附加数据或处理请求。数据库服务器也可以反过来成为一个客户机，在这种情况下，它代表原始客户机向其他数据库提出请求。

客户机通信在网络上进行，它可以使用很多不同的通信方法。不同厂商提供的产品各有不同的方法。例如，Oracle使用SQL*NET做为网络通信基础，而Sybase则使用远程过程调用机制。

安全控制是各个RDBMS都具有的功能。图18-1提供了一个RDBMS安全功能概图。

RDBMS解决方案（如Oracle、Sybase或者Informix）提供了很多安全功能，但各个厂商所提供的数据库解决方案却各有其独自的功能。这包括了认证用户和把用户分派到组结构中的能力。在其用户ID或组成员关系的基础上，可以允许或者拒绝用户访问表、视图或者独立数据元素。用户动作的审计跟踪是各个RDBMS解决方案的共同特征。最后，存储过程和触发器之类的高级功能也是可用的。我们将在后面探讨每个安全功能。但是，首先来看一下操作系统访问和数据库之间的关系。

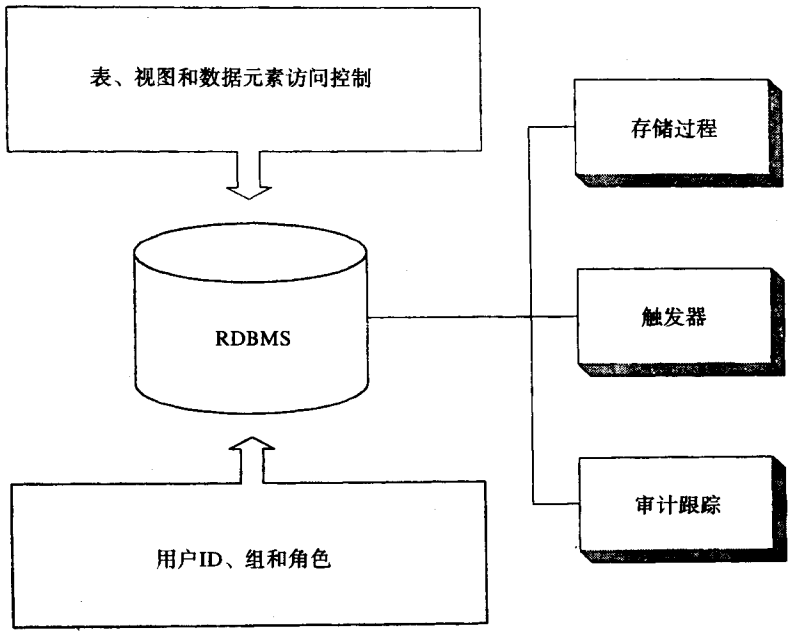


图18-1 典型的RDBMS安全功能概图

18.2.1 用户认证

同操作系统一样，数据库访问的认证也是使用用户ID和密码来完成的。尽管数据库可以使用自己惟一的用户ID和密码，但大多数RDBMS实现可以使用同操作系统一致的用户认证信息。例如在Oracle 7.0中，数据库可以配置成禁止那些使用非标准操作系统用户ID的用户访问数据库。

尽管在LAN上以明文形式传输密码一直是标准做法，但是RDBMS厂商都在着手解决该问题。例如，Sybase和Oracle都有能力实现一次密码（例如挑战响应）认证技术（可选项）。这些技术也可以包括整个认证过程的加密。

18.2.2 操作系统访问

对于数据库应用程序设计者来说，使用中心服务器模型来禁止用户直接访问操作系统是很常见的做法。用户通过操作系统登录，但是会立即置于一个数据库应用程序菜单中。这种方法的优点是用户不需要了解操作系统及其内部命令。它也允许应用程序设计者把重点放在数据库上。

另一种经常使用的技术是强制所有数据库访问都在一个用户ID（例如应用程序用户ID）下进行。这种技术是当前对一些事务监视器的需求。这种方法的优点和缺点如下：

- 相对来说易于实现和管理。
- 所有数据访问都强制在应用程序中进行（只要没人能伪造SQL请求）。
- 强大的数据库访问控制，如表、视图和数据元素安全，会被消除。
- 因为所有的活动都看起来是来自一个用户ID（如应用程序），所以审计跟踪的作用就降低了。

18.2.3 用户配置文件

用户配置文件允许数据库管理员定义有关用户会话的特性。用户配置文件主要用来提供限制用户访问数据库资源的配置设定，但是它也可用来在一段无交互时间后自动断开用户。

18.2.4 授权控制

RDBMS使用特权概念进行授权。特权是执行特定动作的权限。如果用户具有足够的特权，那么他就可以对数据库进行一个给定动作。最初创建数据库的用户称做DB拥有者，他对该数据库拥有全部的特权和权限。DB拥有者向其他用户授予特权（例如，在数据库上执行选定动作的权限），这也包括用户向其他用户授权特权的能力。在大多数RDBMS环境中，数据库拥有者也称做数据库管理员（DBA），他拥有完整的特权集合。

可供授予的特权分为两种，系统级特权提供了用于整个数据库的特权；表访问特权用在一个选定的表或者视图上。

表18-1列出了一般的系统级特权。

表18-1 一般的系统级特权

特权名称	说 明
CONNECT	可以访问数据库，提交命令以操作数据（包括导入和导出），创建视图
RESOURCE	可以改变数据库的结构，创建表和索引，创建和指派用户到组和角色中
DBA	完全访问每个表和所有数据，可以赋予和撤销访问权限，创建和管理用户账号，控制审计功能

具有CONNECT和RESOURCE特权的用户可以访问数据库以及创建新表和索引。只具有CONNECT特权的用户可以访问表但不能创建新表。已经创建了表的用户被给与一个特殊的权利。这些用户称做是对象的拥有者，并且对他们所创建的对象拥有全表访问特权。对象的所有者或者DBA可以把表访问特权指派给其他用户。

表18-2列出了一般的表访问特权。

表18-2 一般的表访问特权

表访问特权	说 明
SELECT	可以选择对象并查询该对象
INSERT	可以在表或视图中创建新纪录
UPDATE	可以改变表或视图中的数据项（记录）
DELETE	可以删除列或行
ALTER	可以改变数据类型，可以添加或删除表列
INDEX	可以创建索引以方便表查询
REFERENCES	可以创建表的特殊索引——外键

DB拥有者、DBA或者拥有资源的用户，可以向独立用户或者用户组授予访问特权。特权可以给与表、视图或者列（数据项）上。用户也可以向其他用户授予特权，然后再继续向其他用户传递特权。只有授予特权的用户才能撤销特权。这里有一个有趣的副作用，原始用户可能不能直接撤销他们表上的独立特权。然而，数据库拥有者可以撤销所有表特权并重新赋予。

18.2.5 责任分离

传统的RDBMS功能，如备份和恢复，需要操作者有DBA权力才可以进行。许多数据库厂商已经认识到这一问题，并在其产品中设计了解决方案。例如在Sybase 10.0中，已经定义一套新系统特权。这些系统特权包括了系统安全负责者和操作员。系统安全负责者允许用户来维护系统ID、密码、审计信息以及特权的管理。操作员特权允许备份和恢复功能的委托。

18.2.6 批处理SQL语句

批SQL任务是包含一串SQL语句的文件。这些语句是通过提供用户ID和密码来进行认证的。批任务可以在没有实际可用的人的时候启动。如果SQL语句中嵌入了一个密码，那么任何能够读取这些语句的人都可以很容易地读取该密码。另一个问题是包含密码的命令行能够被性能监视工具所查看到。解决该问题的一种方法是强制用户在提交批任务的时候输入密码。

18.2.7 用户组

大多数RDBMS系统允许创建用户组。这样，访问特权可以赋予到组而不是独立用户上。一般情况下，组的创建是基于组织结构、任务功能或者二者的结合体的。与为每个独立用户单独标识访问特权比起来，使用组通常能减少和简化数据库的管理工作。一个称做PUBLIC的特殊组允许特权指派到数据库的每个用户上。

当对数据的访问可以同部门或者任务功能行组织在一起并且访问特权静态不变的时候，组结构工作良好。当根据用户所执行的功能类型而对用户授予不同权限的时候，组结构就不再工作。在这种情况下，组结构趋向于太固定。

18.2.8 角色

某些RDBMS实现提供了一个新功能，基于角色的访问特权。角色允许基于任务功能的特权指派。角色不同于组，其原因在于角色要映射到任务的实际职责上，而不是部门或者组织结构的职责。例如，可以为个人经理定义一个角色，该角色允许他们访问所有的个人数据。

角色没有组结构严格，一个角色可以同其他角色结合起来以执行更复杂的功能。角色也可以授予用户，并且同其他角色结合起来以形成新角色。角色不是具体于数据的，它可以同数据库系统特权结合起来。例如，角色可以允许特权用户创建表的新视图——即使该用户没有RESOURCE特权。

在Oracle 7.0中，使用角色可以为备份和恢复创建一个OPERATOR角色，而无需授予用户完全DBA特权。角色可以定制以允许DBA责任的分离。另外，具体角色的使用可以有选择地需要密码。

角色是灵活的，有必要为其制定相应的标准和指导原则。否则，应用角色很可能导致安全权限的重叠和冲突，从而最终面对的是一个混乱的Web环境。

18.2.9 存储过程

存储过程是预定义的SQL语句集合，它可以处理、编译以及优化。一个存储过程调用可以执行多个SQL声明。存储过程对于客户机-服务器模型来说非常具有吸引力，在这种模型下，

存储过程通常要比其他方法更擅于提交SQL语句。存储过程可以被多个程序共享，从而提高编程效率。存储过程也可以被普通用户通过提交ISQL请求来调用。尽管存储过程的标准化工作已经开展，但是当前各个厂商都以不同的方法来实现存储过程。

如果用户有权利执行该存储过程，存储过程可以用来允许存储过程创建者的特权被其他用户临时使用。这些特权是临时的，并且只在存储过程执行时才有效。如果一个存储过程是DBA创建的，那么该存储过程可以允许一个普通用户访问DBA特权。该特点最常见的用途是允许责任分离——允许用户为某些经过选择的功能而受限访问DBA特权。

> 数据库设计者可以使用存储过程来极大地提高安全性。通过撤销普通用户对选定表或视图的访问权并将其赋予到一个存储过程，数据库设计者可以强制所有的访问都通过该存储过程。这不仅降低了管理难度，而且能够防止用户、程序员以及应用程序访问敏感表。存储过程也可以拒绝对SQL报告书写器的访问，并保证审计跟踪的完整性。

可用密码来授权存储过程的使用。同批SQL语句一样，问题在于应该把密码放在哪里。经常使用的一个方法是，如果存储过程是从PC客户机上执行的，那么应该把密码隐藏在该PC机上的一个库中。

18.2.10 触发器

触发器是在一个预定义的任务进行以后自动执行的一个存储过程，其目的在于维护数据库的参考完整性。正常情况下，触发器是由同数据相关的任务启动的，如表插入、更新和删除。从安全角度上看，触发器对于施加额外安全控制来说是非常有用的。

在RDBMS环境中，为了提高安全性和审计，触发器的使用是很常见的。有一种数据库实现使用触发器来有选择地监视对敏感记录的访问。在这种情况下，所有对高级管理人员工资记录的访问都会被审计。触发器的一个更精妙的用途是用做数据库安全的一个备份。创建和维护一个单独的用户访问特权表，该表同正常数据库访问特权是分离的。这种方法意在检查出任何破坏安全性的用户，并给与他们另外的特权。同存储过程一样，各个厂商对触发器的实现各不相同。

18.2.11 远程过程调用

远程过程调用（RPC）是一种特殊的存储过程，它设计为在客户机-服务器环境中运行。正常的存储过程可以在服务器上（例如，在数据库引擎上）独占运行，但RPC是由客户机启动的，然后引起服务器上某种形式的程序执行。通常这会导致一个响应发回到客户机上。RPC的不同类型包括OSF DCE RPC，Sun的RPC以及Sybase RPC机制。

同所有的客户机-服务器通信一样，使用RPC也有几个需要考虑的安全问题：

- 如果RPC进行认证需要依赖客户机发送的数据（例如用户ID），那么如何才能保证客户机没有发送伪造的信息呢？
- 如果客户机熟悉RPC调用的机制，那么如何才能防止客户机完全绕过认证方法呢？

答案取决于远程过程调用机制的强度。

18.2.12 审计机制

RDBMS解决方案中有两种可用的审计机制，事务日志和审计跟踪。事务日志是在系统故

障后恢复数据库的方法。这些日志记录了对数据库的所有改变情况，如果出现问题，那么可以使用这些日志来重新建立数据库。在数据库术语中，事务定义为执行逻辑条目所需的一系列动作。通常情况下，事务明确定义在一个BEGIN WORK语句之后，并且只能执行到一个COMMIT或者ROLLBACK语句为止。一些数据库实现也支持数据库级的事务记录，如最近的Informix版。这种类型的日志记录了所有的动作，不管一个独立的事务是否已经提交了一个BEGIN WORK语句，数据库都可以恢复。事务日志通常保存在数据库外部。

事务日志的中心功能是提供了在数据库或者事务失败的时候回滚不完整事务的能力。回滚的经典用途是银行应用中的资金转移。当一个储蓄账号和一个提款账号之间要进行资金转移时，如果系统出现故障，那么账号资金不能失去平衡。如果在储蓄账号中已经提出资金但是提款账号尚未得到资金之间出现系统故障，那么整个事务都必须要回滚，从而使得两个账号都回到一个一致的、平衡的状态下。事务恢复机制保证了事务会按照正确的顺序回滚，从而保持数据库的一致性状态。

RDBMS环境也提供了审计跟踪功能。RDBMS审计跟踪提供了数据库变化记录。在目的和结构方面审计跟踪不同于事务日志。事务日志的用途是在数据库出现故障的时候进行数据库恢复，而审计跟踪则着眼于跟踪事件。尽管审计跟踪也跟踪数据库变化，但是它所跟踪的变化是经过选择的，并且不一定是完全的。它们也是不同步的。如果从审计跟踪中重新构建一个数据库，那么该数据库可以肯定是不可靠的和不一致的。

作为一个安全工具和恢复机制，RDBMS审计是非常有用的。如果启用的话，审计可以用来跟踪安全事故中的用户动作。如果监测到了特殊安全事件，审计也可用来向触发报警器提供数据，或者提供警报。这也包括用户失败访问表或视图，这种情况有可能意味着用户正试图破坏正常的安全机制。

18.3 有关RDBMS的问题

数据库环境是一个非常健壮的安全环境，对其操作可以予以适度信任。不幸的是，要进一步信任RDBMS环境，首先要解决很多问题，其中既有数据库内部问题也有数据库外部问题。

尽管RDBMS授权机制是非常健壮的，但是基本的认证机制并不足够强大。问题如下：

- 如果客户机或者服务器上的操作系统安全性破坏，那么数据库安全性也可能会破坏。
- 如果用户不能直接访问操作系统或数据库，那么他们如何才能定期改变密码呢？
- 批处理应用程序授权问题：如果批SQL语句中含有明文密码的话，应用程序就很容易破坏。
- 如果厂商为认证过程提供加密处理，那么解决方案可能是厂商独有的，因而不能被其他RDBMS所使用。
- 一个RDBMS可以代表用户来调用其他RDBMS，支持这种代理行为的认证和授权机制具有很多问题。本质上，客户机的标识是以明文形式（或者根本就没有）传递到第二个RDBMS上的。
- RDBMS需要有能力在用户和访问类型的基础上限制ISQL访问。
- 大量远程RDBMS系统的管理需要自动工具来同步用户ID、密码和用户访问权限。

授权机制的强度是由伴随的认证机制来决定的。在大多数数据库环境中，密码在网络上

是以明文形式传输的，因此它们很容易被LAN流量监视工具截获。操作系统认证同样难逃厄运。

如果操作系统控制是脆弱的，那么不管数据库的内部控制如何强大，破坏者都可以很容易地破坏它。客户机认证程序（例如logon登录程序）可能会被特洛伊木马所取代，木马程序伪装成合法程序，但实际上会捕获用户的密码。如果一个UNIX系统上的安全性很松散，那么攻击者可以使用一个调试工具来检查驻留数据库的设备文件，并得到DBA密码。

> 数据库管理员应该掌握对UNIX原始磁盘设备的攻击。使用很多常见的调试工具（如UNIX八进制转储工具），攻击者可以很容易地发现许多RDBMS版本的DBA密码。不管RDBMS控制有多强大，真正起作用的是操作系统的安全性！

如果用户访问数据库应用程序，那么后者要代表用户去访问第二个数据库，这会出现一些问题。所有的请求看起来都好像来自同一个地方（初始数据库），如何才能在第二个数据库上施以访问控制呢？

如果许可客户机可以通过一个应用程序在数据库表中插入行或列，那么需要注意什么问题呢？应用程序可以保证数据库会以一个一致的方式修改，但是如果实现了ISQL的话，它们的访问就是不受控制的了。

对多个数据库的数据库用户和特权进行集中式管理也是一个问题。RDBMS安全管理一般是由数据库厂商在一个中心数据库的基础上提供的，它不允许集中式管理。

18.3.1 附加的解决方案

RDBMS环境有很多可用的第三方解决方案。这些解决方案提供了大量的安全增强功能，其中包括：

- 用于认证的一次挑战-响应密码解决方案——防止在LAN上发现密码。
- 密码策略增强，例如最小密码长度限制和过期时间限制。
- 用户、角色和特权的集中式管理。
- SQL访问强制通过SQL访问控制机制，每次查询都需要控制机制进行授权。
- 提供日历访问控制。
- 用户动作和命令的审计跟踪。
- 基于存储过程的用户访问控制。

一般来说，数据库厂商非常乐意为其环境提供第三方解决方案分类目录。

18.3.2 传输安全性

许多数据库厂商都已经认识到，在客户机-服务器世界中，必须要有在不可信网络上提供通信完整性的机制。这些机制为客户机和服务器之间的认证处理和后续通信提供加密。这也可以包括对二者之间所传输的数据的加密。

例如，Oracle公司提供的安全网络服务（Secure Network Service，SNS）产品就是这种类型的机制。SNS在离开客户机之前对密码进行加密。通过支持DES40算法和RSA数据安全公司的RC4算法，SNS提供了数据完整性。

Sybase 在其10.0版中也提供了这种功能，它可以对用户密码的存储进行加密。另外，当

密码从客户机发送到服务器上的时候，它还可以对密码进行加密（可选项）。

包括Informix和Oracle在内的大多数RDBMS厂商都宣布了结合OSF/DCE环境的产品。从信任角度上看，可以使用DCE支持的Kerberos认证模型来认证数据库用户。从DCE安全服务处得到的证明可以由数据库服务器校验。另外，也支持DCE定义的6个加密级别。这使得可以使用多级安全措施（例如加密）来保护从客户机到数据库服务器的调用。DCE安全的使用也提供了单一登录功能，这可以为其他非RDBMS应用程序所用。有些厂商也允许DCE安全服务来管理和控制数据库安全特权，如用户到角色的指派。

不幸的是，并不是所有厂商都为全部OSF/DCE环境提供全面的支持。例如，只有一些厂商支持命名服务，并不是全部的厂商解决方案都提供支持。

18.3.3 数据合并

在使用数据库时，一个普遍让人头疼的问题是不能访问和掌握已经累积的大量数据。做出更好的决定需要足够的信息。但是如果不能定位数据，那么如何从信息中受益呢？另一个问题是性能问题：如果提供了产品数据库的在线查询能力，那么如何保证查询性能？建议建立数据仓库来对付这些挑战。

18.4 数据仓库的概念

数据仓库是由公司生成和保持的数据组成的有组织的数据集合。更为正式的定义是，数据仓库是一个面向目标的、集成的、随时间变化的、永久的数据集合，它主要用来支持组织决策。数据仓库用做收集、标准化和总结操作系统中进行的事务累积的仓库。数据仓库中的信息实际上是静态的，其主要用于决策支持和管理报告。数据仓库更多的是一个组织概念而非一个技术概念。如果实现正确的话，数据仓库为组织进行更好的决策提供了一种方法。数据仓库化用于以下方面：

- 历史和总结数据的一个中心分发点。
- 根据目标而非应用组织数据。
- 集成数据以满足整个组织的需要。
- 为建模和趋势分析提供历史数据。
- 为用于组织的信息提供标准定义和表示。

若要真正起作用，数据仓库必须要包括数据仓库用户轮廓、把数据放置于仓库中的用户列表、仓库中所包含的数据分类。为了使用户可以访问并有意义地报告数据，必须要有可用的工具。如果要求重复报告，还可能会包括预约服务。

通常情况下，数据仓库用做分析和决策支持工具，它一般不支持业务事务处理。数据仓库中所存储的数据是基于目标、产品或者员工的，而不是基于操作数据库的事务的。在数据仓库中保存数据应该同时保存对数据何时收集的指示说明。我们需要知道数据的时间关系以便掌握生成的报告并以历史观点来分析。数据仓库中的数据通常是只读的。这些数据既不是数据仓库生成的，一般也不会在数据仓库中更新或者删除。

数据仓库是不容易定义和实现的。标识置放于仓库中的数据一定要小心谨慎。数据从哪里来，如何总结或者如何进行过滤以决定是否进行存储，对这些问题一定要仔细解答。如果不进行仔细的分析和计划，那么所实现的数据仓库必定只不过是提供了另一个数据存储地点，

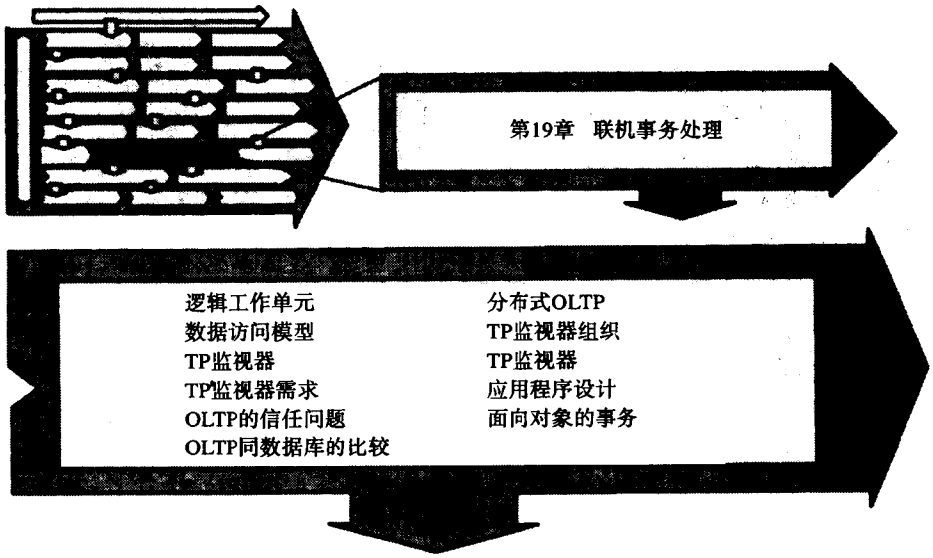
数据很难访问和使用。从源系统中提取、变换以及装载处理信息时一定要小心谨慎。

从安全和信任角度看，数据仓库提供了集中管理访问公司数据的功能，而不管数据位于何处。一个RDBMS所面对的基本安全问题在数据仓库中仍然存在。为一个集中式数据库解决这些问题要比为多个分布式数据库解决问题容易得多。分布式数据仓库要解决分布式数据库所面临的所有安全问题。

18.5 结论

数据库可以起到保存重要信息的防护墙作用。数据库可以保护应用程序数据（至少在某种程度上）——即使在系统已经破坏的情况下。数据库提供了强大的数据访问控制，这包括根据特定标准来限制数据的用户视图。数据库也提供了相对健壮的审计跟踪，并把它们向本地系统管理员保护起来（但并非全部）。这些优点使得数据库成为在分布式环境中存储和保护应用程序数据的合理地方。

第19章 联机事务处理



自动柜员机（ATM）的出现使得很多银行事务不再通过银行营业员来办理了。在进行事务时，用户不会关心他所使用的是什么类型的ATM或者这些机器由谁控制。当用户通过ATM来存钱或者取钱的时候，就可能执行一次联机事务。事务的概念实际上是批处理的扩展，其中执行一个计算机程序、访问和操作多个文件，并且根据所输入数据而传送产生的结果。在该过程中，计算机终端用来输入文件和显示结果，从而代替了输入文件和打印报告。

银行和保险机构是联机事务处理（On-Line Transaction Processing，OLTP）的早期用户。当你站在银行柜台前时，银行营业员使用OLTP系统能够访问准确的信息和进行资金事务。在你离开柜台之前，这些事务就已完成，并且你的账号得到了更新。在本章中，我们将探讨一下事务的组件以及它与其他处理形式的区别。当事务成为分布式系统的一部分时，它们的管理和安全性就变得非常重要了。

飞机订票系统是OLTP的最早应用，它能很好地说明OLTP的能力和价值。使用OLTP系统，航空公司能够合理地安排飞机航位，即使不能准确的知道具体座位上的乘客或者座位具体由谁来出售。旅游或者航空代理可以代表航班来销售机票，通过访问一个数据库，他们可以查到是否有空位，如果有便向乘客销售。OLTP系统需要在很短的时间段内处理大量的事务，并且要向机票代理机构提供及时的响应。试想一下，如果航空公司没有使用联机事务处理系统来管理航班的能力，那么空中旅行会多么的可怕！

银行和飞机订票系统在事务出现时进行事务处理。例如，ATM遵循一个严格的协议来进行事务执行和控制。银行需要保证当你在你的账号上提钱时，总是能够得到正确数额的现金；如果机器因为某种原因而出现故障，那么你的账号不会减少实际上并没有被提取的资金

数额。银行必须保证不能出现如下情形，一个用户从一台机器上提取200块钱，并在另外一台机器上或者从银行的另一个分部中也提取相同的200块钱。

在信任方面，事务系统与基于普通用户的系统有所不同。安全性通常分布到操作系统、网络系统以及事务系统本身上。在一个事务系统中，事务的实际用户以及事务启动的位置可能是瞬时的。执行一个事务所需要的授权是基于用户的已认证过的标识的。接受事务所需要的授权是基于事务启动位置或者网络访问的已认证过的标识的。例如，对一个ATM上的银行事务的授权要依赖该ATM的物理位置和标识，以及事务用户的标识——用户需要提供一张卡和一个个人标识号（PIN）。在分布式事务处理系统中，解决这种授权问题是非常困难的。

19.1 事务的概念

在过去，区别事务驱动的系统和非事务驱动的系统是很容易的。如果使用一个终端并按回车键，那么它就是一个联机事务系统。那么到底如何定义事务呢？事务是一个用户和一个应用程序之间、或者两个或多个应用程序之间的一个数据驱动的关系，它需要在所提供数据的基础上实时执行一个程序并产生可证明的结果。事务有一个明确定义的开始和结果，并在二者之间进行某种形式的程序执行。联机工作单元看做是一个事务，前提是它具有所谓的ACID属性。ACID属性如下：

- **原子性（Atomicity）**——事务是完整的或者是不完整的。事务不能部分完成，并处于一个不确定的状态下。
- **一致性（Consistency）**——更新数据需要保持数据的完整性，即从一个稳定（一致）状态变换到另一个稳定状态。
- **隔离性（Isolation）**——对于共享数据上的事务来说，其作用只应该在该事务提交后才能被其他事务所看到。进行中的事务所需要的任何信息都应该加锁以防止其他事务修改它们。
- **持久性（Durability）**——事务的作用是持久的，并且对数据的关键修改不会被后面的系统故障破坏掉。

> 事务是进行联机访问的一个完整的工作单元以及在保证完整性的情况下对共享数据的更新。工作单元的执行必须要遵守ACID属性，否则便不能看做是一个事务。

19.1.1 分布式逻辑工作单元

在第9章中，我们已经介绍了逻辑工作单元（LUW）的概念。现在，要把这个概念扩展成分布式逻辑工作单元。逻辑工作单元的概念依旧不变，只不过扩大到具有多个处理器的分布式环境中。在图19-1中，一个独立的逻辑工作单元分布到两个服务器上。读写文件X、读文件Z、然后请求在服务器B上执行一个过程——读并更新文件Y，这些动作可以看做是一个逻辑工作单元。逻辑工作单元包括了服务器A上的过程A和服务器B上的过程B。如果从过程A的开始到文件Z的更新这一范围内的任何操作出现失败，那么文件X、Y、Z都要退回到过程A启动的时刻之前。每个逻辑工作单元代表一次任务的完成。

在前面一章中，我们已经探讨了回滚的机制，回滚的目的是为了保持数据库始终处于一个一致的状态。在事务处理中，提交和回滚概念是非常重要的。提交是事务周期中的一个时

刻，到该时刻完成的所有工作都会永久使用。如果事务过程中出现任何错误，那么就会调用一次回滚，从而把所有一切都重新设回到前一次提交时刻前，或者事务开始执行前。提交通常是在一个逻辑工作单元的完成之前调用的。如果需要的话，就会进行回滚，从而把事务设置回到逻辑工作单元开始之前。

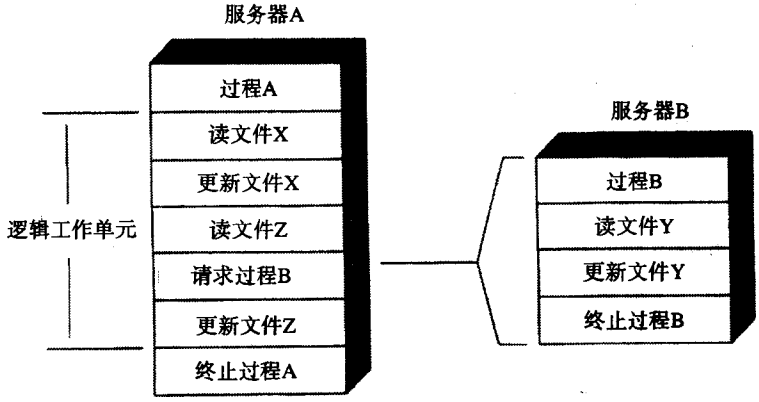


图19-1 分布式逻辑工作单元示例

19.1.2 分布式数据访问模型

OLTP系统收集和处理关于业务事务的信息，并把改变施加到组织共享的数据库和文件上。对于访问共享文件或数据库来说，一个长期存在的问题是谁管理这种访问。用户必须管理这种访问吗？或者，如果进行自动处理，那么需要注意些什么？图19-2说明了两种不同的程序执行和共享数据访问模型，它们描述了不同的客户机-服务器需求。

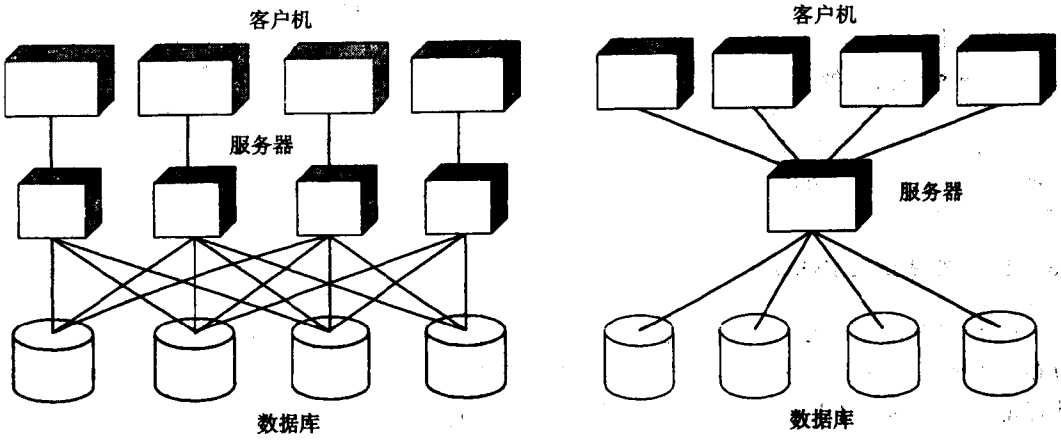


图19-2 事务处理模型

在一个过程中访问多个文件或数据库的一种方法是让客户机来维护服务器过程的管理和信息。在这种情况下，客户机需要知道数据库服务器所在的位置，从而加重了客户机的负担。所以，客户机管理模型并不广泛使用。解决此问题的另一个方法是让服务器来维护文件或者数据库的管理和信息。同样，协调和更新问题仍旧存在。管理数据库访问的一个更好的方法

是把程序的执行和对文件或数据库的访问都集中起来。这样，访问和管理文件或数据库的任务是由软件而不是用户来承担的。因而，管理难度大大降低了。

19.2 事务处理系统的组件

如果要集中管理事务的执行和对不同数据库的访问，那么必须要有一个正确的机制。事务处理（Transaction Processing，TP）系统利用TP监视器主程序来管理事务的执行。TP监视器起到一个微型操作系统的作用，它具有对事务执行和数据访问进行调度的能力。TP监视器也管理ACID属性，并提供维护这些属性所需的服务。当事务执行成功或者完成时，TP监视器管理提交请求；如果执行不成功，那么TP管理器调用回滚。TP监视器能够在过程失败的情况下维护数据的完整性。

图19-3说明了OLTP系统的各个组成部件。在这些组成部件之中，有一些已经移到了OLTP系统之外，从而利用数据库系统的优点和桌面系统的处理能力。屏幕管理器提供了用户界面机制，现在，它一般由一台本地PC来提供。OLTP系统所需的资源管理是依赖于实现系统的操作系统的。事务处理服务负责分派要进行的工作。资源管理器提供了对处理器、内存以及其他资源的有组织的访问。通常，数据的管理被委托给独立的数据库系统（OLTP系统需要与其通信）进行。分布服务组件管理数据的共享和对数据的访问。另外，有很多工具可用于帮助构建基于应用的事务。TP监视器本身就是一个应用程序，它需要使用操作系统和网络传输的处理和通信服务。

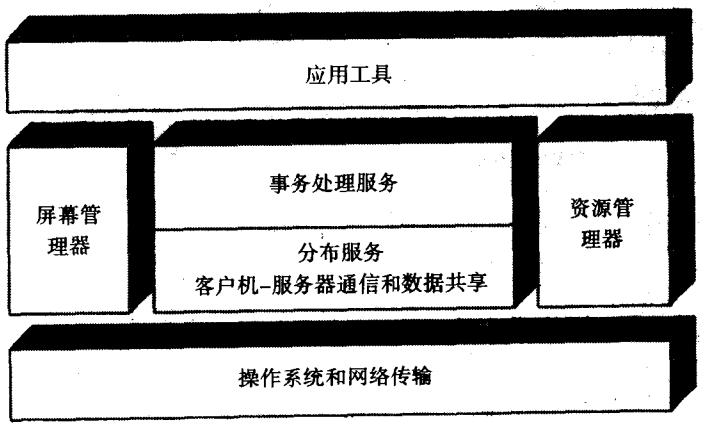


图19-3 OLTP系统组件

19.2.1 TP监视器

由于本身性质的原因，事务的起源和其在一个给定的时间段内需要处理的次数都是不可预测的。我们可以猜测有多少人可能要从银行中取钱，不管是用ATM还是经过银行营业员。但是，我们不可能知道准确的数字。我们需要有一个方法来管理这些事务的处理，从而做到能够以这样一种方式来分配所需的资源，即保证事务准确和尽可能快地被处理。若要进行预测，那么最容易的方法是假设所有的ATM和银行终端都具有相同的使用率。但这是不切实际的，不可能所有的ATM或者银行终端都在全天时间内完全利用。需要一个机制来帮助我们管理事务发生的高峰期和低谷期。这种事务管理机制通常称做事务处理监视器。

使用联机事务处理监视器的主要目的是管理事务的调度和资源的访问。资源经常用来描述可以消耗或者需要管理的任何东西，如处理器内存、数据文件和数据库、处理器周期以及通信系统。事务监视器必须管理事务，即分配资源以保证事务能够尽可能快地完成。OLTP监视器必须能够动态的管理可变数目的事务，因为我们对事务的数目和发生时间根本无法预测。

联机事务处理监视器经常用于分布公司内部以及公司之间的事务。OLTP监视器必须能够提供可以跨越独立OLTP和数据库系统的事务服务。这包括了完整事务或者事务处理组件的路由。如果你使用一个ATM，而控制该该机器的银行并不是你的账号所属的银行，那么在这种情况下事务必须要在至少两个系统中处理，一个系统用来从你的账号中扣除提走的钱数并把钱从你的银行传送到拥有该ATM的银行，而另一个系统则用来分发钱、减少该ATM上可用的现金量并从你的银行中接受钱以补偿你的提款。

19.2.2 TP监视器需求

前面我们已经介绍了ACID属性，并把它们作为判断一个工作单元是否能称做一个事务的标准。为了提供事务系统所需的服务，TP监视器必须也提供附加需求的解决方案。事务所需的资源是由TP监视器管理的。因此，TP监视器必须包含通常情况下可以在操作系统或者数据库系统中找到的属性和机制。表19-1概括了一个OLTP系统必须提供的需求和性质，比较各种商业TP监视器应该参照此表。

表19-1 联机事务处理监视器需求

需 求	TP监视器
优先级管理	根据事务预先指定的优先级或者到达时间对事务的执行进行优先级排序
恢复	在出现系统或者应用程序故障的情况下保存数据更新
应用程序接口	通过标准API同带有事务监视器的应用程序进行交互
数据完整性	当管理并发更新和系统故障时保持数据完整性
数据访问	提供对大型数据库的可靠的实时访问
性能	在不牺牲用户性能的情况下同时处理许多事务
安全性	只对授权用户提供数据访问权，并且只允许通过授权途径访问数据

19.2.3 OLTP是可信任的

OLTP系统提供了信任的可用性和性能组件。TP监视器技术已经出现20多年了，并且已经开发和调整成可以提供高事务处理性能。并且在分布式环境中它也可以提供这种性能。ACID属性成为OLTP环境的一个特征已经有了很长的一段时间。这些属性也已经扩展到一个异构环境中的多个平台上。尽管ACID属性在本地数据库中是可用的，但是一般它们不扩展到一个分布式的、异构的环境中。几乎所有的TP监视器都包含能够管理应用程序访问的安全机制。许多OLTP实现还支持DCE和DCE安全服务。这种支持可以在连网事务中提供强大的授权、认证以及机密性控制。

19.2.4 OLTP与数据库

数据库产品和联机事务监视器都试图解决同样的问题，但二者采取了不同的方法。如果

要在OLTP监视器和数据库解决方案之间做出选择，那么这要取决于具体问题所定义的具体需求。看起来，这种选择也是由这两种方法后面的宗教信仰所驱动的。对于每个事务问题来说，可能没有完美的选择，但是有一些考虑事项是需要注意。表19-2给出了TP监视器和分布式数据库系统在解决基于事务的问题方面的对比情况。

表19-2 事务处理监视器同分布式数据库的比较

性 能	TP监视器	DDBMS
TP环境	只允许事务和批任务，不能进行特别查询	事务、批任务以及特别查询
客户机-服务器支持	调用服务、对话	远程存储过程调用
分布化	使用应用服务或者对等连接	使用数据或者存储过程访问
异构访问	X/Open DTP接口	专有网关
资源管理范围	可以跨越多个资源管理器提交资源	不能使用远程资源管理器来提交
可伸缩性	支持同一工作负载几乎不需要资源	需要另外的线性资源
性能	能够操作更快、处理更多的卷	在重负荷下性能很差
事务管理	X/OPEN DTP或者专有的	专有的
用户数量	大数量（>100）	小数量（<100）
数据位置	保存在异构数据库或文件中	保存在同构数据库中
分布范围	任意多个服务器	三个服务器或者更少
安全性	需要保护业务功能	足够的数据安全性

所有的数据库系统厂商都对其产品的事务处理速度（每秒或者每分）大做宣传。尽管这些产品在一个本地环境中可以得出令人惊奇的成绩，但是如果把事务工作单元分布到网络上，它们就没有那么好的表现了。存储过程是主要的数据库事务机制，它用来在数据库中启动事务，但是它一般不能同分布式环境中的其他其他事务单元一起使用。数据库只能提交由该厂商数据库所管理的事务资源。数据库不能同步或者提交由一个外部数据库（其他资源管理器）所管理的工作。

数据库解决方案非常适合于具有大量用户的本地环境。可以快速开发和实现应用程序，并且它们很容易进行管理和设置。有关数据库技术应用的开发软件数不胜数，并且关于数据库技术的即时可用的打包应用程序也不胜枚举。TP监视器通常可以管理任何网络地点或者其他应用中所调用的更高的事务卷。然而，OLTP应用程序通常更加复杂并难以开发。

19.2.5 分布式OLTP

TP监视器跟踪在单一服务器内执行的事务。在分布式环境中，我们也需要跟踪在多个服务器上执行的事务。为达到此目的，TP监视器需要具有另外的机制以管理在多个服务器上执行的逻辑工作单元。这包括了提交和回滚需求。在第9章中，我们介绍了几个用于在客户机和服务器之间共享工作的模型。在TP系统之间共享工作使用的也是这些基本模型。我们需要定义一个可用于分布式事务系统的标准结构，目前，这方面的一些工作已完成。

1. OSI事务处理标准

国际标准化组织（ISO）已经定义了开放式系统互联（Open System Interconnection，OSI）分层网络结构标准。该体系结构定义了一些协议和服务，它们分布到七个层次中。OSI模型的层次划分用于许多其他的网络协议和服务定义，但是它没有定义具体的协议。该体系结构模型要比包含在其内的OSI协议更常使用。OSI体系结构也包含了两个用于事务处理系统互操作

的标准。OSI-TP（事务处理）标准说明了应该如何定义和管理事务标识符。另外，它也包含了一套用于协调逻辑工作单元提交和回滚（在事务失败的情况下）的机制。OSI-CCR（提交、并发控制和恢复）为使用在单一会话上的工作单元提交定义了协议。

2. X/OPEN分布式事务处理

X/OPEN是一个由多个致力于推进开放式系统发展的公司所组成的非盈利的国际联盟。在X/OPEN术语中，开放式系统的定义是由通常可用的产品所组成的一个独立于厂商的计算机环境，它是通过使用已接受的官方标准和事实标准来实现的。X/Open的分布式事务处理（Distributed Transaction Processing，DTP）模型定义了一套应用编程接口（API）和系统级接口，使用这些接口，应用程序可以同跨越多个平台的不同事务管理器进行互操作。虽然事务需要跨越多个平台，但它仍然是一个逻辑事务。

图19-4描述了X/OPEN分布式事务处理模型以及资源管理器之间的API。DTP模型是基于四个软件组件的：

- 应用程序（AP）定义了事务边界和构成事务的过程。
- 通信资源管理器负责事务在多个应用程序间的协调。
- 资源管理器（RM）提供了对共享资源的访问。
- 事务管理器（TM）协调和管理事务，并在出现失败的情况下进行恢复。DTP标准使用了两阶段提交——有关详情，请参考第9章。

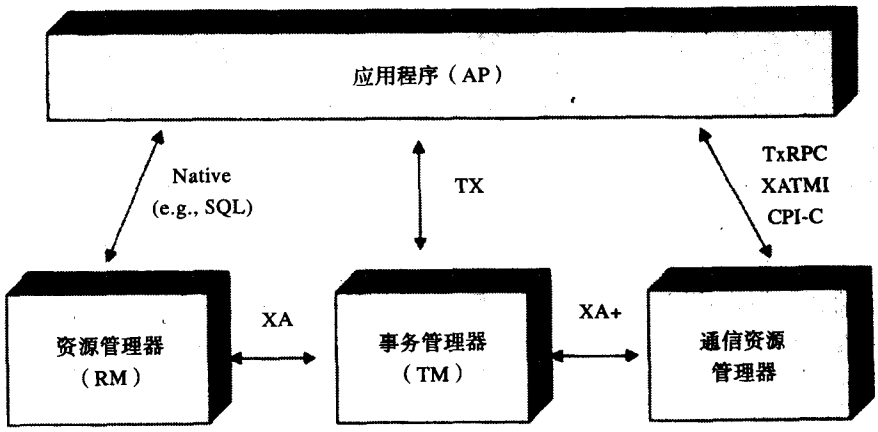


图19-4 X/OPEN分布式事务处理

X/OPEN分布式事务处理模型为管理器间的服务通信定义了API。通信资源管理器包括了三个接口，这三个接口是用来在应用程序之间进行通信的。公共编程接口通信（Common Programming Interface Communication，CPI-C）使用了用于对等通信的SNA规范LU 6.2。事务RPC（transactional RPC，TxRPC）是由Encina TP监视器使用的机制。XATMI规范是由Tuxedo OLTP系统使用的。通信管理器同使用XA+ API的事务管理器进行通信。该API的作用是把分布式工作单元的状态通知到本地事务管理器。XA接口是资源管理器和事务管理器之间的接口，其作用是同步资源变化。TX接口是事务管理器和应用程序之间的接口，其作用是定义工作单元的开始和结束。如果遵循这些API规范，事务和TP监视器就可以相互操作。

19.2.6 TP监视器组织

前面我们讨论了X/Open的DTP模型。该模型定义了资源管理器和应用程序之间的责任和接口，其目的是支持分布式事务。我们将以DTP模型为基础来描述一些服务，这些服务是由一般OLTP系统的组件管理器所提供的。图19-5中所说明的模型包含了支持分布式事务处理所需的一些组件。这是一个逻辑模型，其中一些服务可以由位于具体的OLTP系统之外的组件来提供。

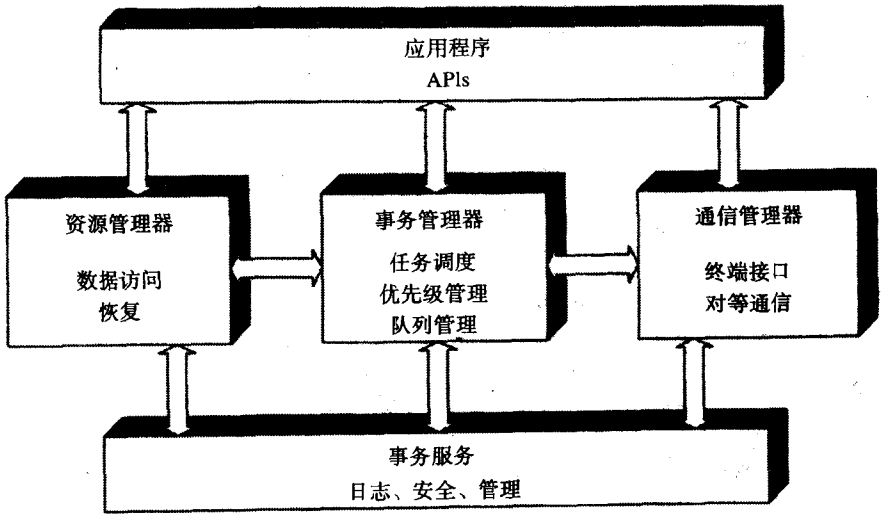


图19-5 OLTP模型

通信管理器通常是从用户或者从另一个互连系统处接收事务的第一个管理器。事务处理系统的核心是事务管理器。该管理器的任务是接收事务请求和监视事务执行。事务可以根据其类型和一个预定义的类别来进行优先级安排，另外事务到达时间也是一个考虑因素。高优先级的事务要比低优先级的事务先分派。资源管理器同文件或者数据库进行交互，从而访问和处理事务所要求的数据。应用程序通过一套标准API来同管理器进行交互。为支持事务处理系统，模型还提供了一组功能来提供事务日志、管理以及安全服务。

1. 资源和通信管理器

资源管理器在事务的执行中起到了非常重要的作用。通信管理器规定了事务处理系统和用户或者另一个互连系统之间的互操作。使用正确的协议来汇集消息，同所需的用户或系统建立通信。资源管理器处理数据请求，因而需要同文件系统或者数据库进行交互。两阶段提交所需要的功能也在这里管理。如果事务失败或者系统出现故障，那么文件和数据库必须要保持一致状态。这必须扩展到互连的系统上。

2. 事务服务

OLTP模型需要具有事务管理功能来配置和管理事务以及资源管理器。事务必须要标识成执行的多个过程所组成的一个集合，并且为了执行，事务必须要分配适当的资源。其他事务服务还包括跟踪工具——其作用是跟踪事务的执行并帮助进行问题确定。转储工具能够提供应用程序状态和相关资源的快照，这些信息可用于问题分析。另外，模型还需要安全服务来认证用户并确定用户是否可以授权执行事务。

19.2.7 TP监视器

事务系统中的主要开发工作是在分布式处理这一方面。目前，大型机系统和高度专有系统提供了很好的处理性能。为了在替代平台上提供事务处理能力，以及在各种平台间提供分布式处理能力，已经开展大量的工作。在各种类型以及各个档次的平台上进行分布式事务处理是健壮事务处理系统的一个目标。目前，有几个商业事务监视器产品可用来进行分布式事务处理。下面，将简要介绍几个流行的具有分布式事务处理功能的产品。

1. Encina

Encina产品系列是由Transarc开发的，它定位于解决开放式、分布式OLTP需求的特殊需要。在这里，“开放式”的定义主要指的是UNIX市场。Encina利用了OSF/DCE所提供的一些功能。Encina提供了一个全功能的TP监视器，并为在IBM和HP的UNIX平台上实现CICS提供了许多基础服务。

Encina的体系结构是建立在体系结构的两级技术基础上的。Encina利用OSF/DCE的很多组件来进行一些分布式服务。第一级包含了把DCE环境扩展成支持分布式事务支持的服务，第二级包括了汇集和操作分布式事务的服务和资源管理器。

Encina监视器是支持分布式事务的事务处理引擎，它提供了负载平衡、单元管理以及分布式事务调度功能。访问该事务监视器的客户机可以是终端、PC或者UNIX工作站。该监视器使用DCE的安全服务来进行认证和授权，并使用了访问控制列表（ACL）。为配置和支持分布式事务，Encina监视器也提供了所需的管理功能。

2. CICS

IBM的客户信息控制系统（Customer Information Control System，CICS）是用于事务处理应用的事实标准，它在全世界有超过36 000个的实现。财富百强公司中的大部分都使用CICS环境来支持某种类型的事务处理系统。CICS系统是存在时间最久的事务系统之一，它是在20世纪60年代后期设计出来的。CICS的基础结构开发成运行在IBM大型机操作系统上。

CICS系统管理内存分配、处理器上的工作（事务）调度、文件访问，并把结果传送给用户而不是实际的操作系统。IBM提供了一些管理工具来辅助CICS系统的配置和操作。执行在CICS上的事务受到了内部CICS安全机制和外部安全监视器（如果可用的话）的保护。CICS的内部安全机制使用了一个事务级和资源级安全。用户登录可以用来建立用户ID和事务和资源安全级别，但是用户登录并不是必须要求的。事务安全级（Transaction Security Level，TSL）和资源安全级（Resource Security Level，RSL）密钥列表是同一个用户定义相关的。这些列表的作用是控制执行事务和访问资源的能力。如果事务或者资源的密钥同用户的RSL或者TSL中的一个密钥相匹配，那么操作就可以继续进行。

IBM和HP都提供了一个运行在UNIX操作系统上的全功能CICS系统操作。UNIX上的CICS实现实际上是几个分布式计算功能的组合，这些功能提供了互相补充的服务。CICS监视器运行在Encina提供的服务之上，并取代了Encina监视器。在CICS范围之外的是DCE提供的分布式元素。CICS设计所基于的大型机原则并不适合于UNIX环境。大型机环境适合执行批处理程序，而UNIX环境则定位于执行登录用户的命令。

3. Tuxedo

Tuxedo产品是一个更为成熟的UNIX平台OLTP产品。它是由AT&T在1983年引入的，曾经

有一段时间归Novell所拥有，现在是BEA系统公司的产品。Tuxedo是最早为分布式系统提供TP监视器的产品之一。该产品分为三个级别，PC或者工作站、UNIX系统以及大型机。如果同CICS系统一道安装了用于Tuxedo的附加大型机软件，Tuxedo能够同CICS进行互连。Tuxedo下进程间的通信是使用软件桥来实现的，软件桥在不同机器上的进程间转发消息。Tuxedo支持依靠数据的路由，即根据消息的内容把消息正确地路由到合适的服务器节点上，同时不需要应用程序编码。这种路由标准是在配置表中定义和管理的。

4. Top End

Top End是由NCR开发的一个事务处理产品，它是在1990年进入市场的。Top End支持多个API，其中包括一套用于CICS的和一套用于Tuxedo的。Top End已经移植到了IBM、HP、Sun的平台上，并且也可用于AT&T支持的平台上。Top End是一个运行在UNIX平台上的消息传递产品，DOS、Windows、OS/2或者Windows NT都可以用做客户机。该产品也提供了工作负载平衡能力。消息的路由也可以建立在消息内容的基础上。Top End支持两种应用风格，标准应用程序模型和管理服务器模型。在标准应用程序模型中，应用程序完全由其自己的事件来管理，并且应用程序管理着整个事务处理。管理服务器模型同Encina或者CICS产品类似，在这种模型下，Top End系统代表应用程序来维护控制。Top End产品的主要用途是支持分布式决策支持系统，这同一般的分布式事务处理系统完全一样。

19.2.8 应用程序设计

设计基于事务的应用程序与设计一般的批处理或者命令驱动的程序有一些不同之处。在这种应用程序中，多个事务可能会同时执行共享资源。设计事务应用程序的原则是，持有资源的时间应该尽可能地短。前面我们已经讨论了逻辑工作单元。事务应用程序应该设计成把所需的功能都组织到工作单元上。文件或者数据更新需要持有一段时间的互斥锁。其时间越短，其他事务的访问就越顺利。例如，一个事务应用程序要读一个数据记录以进行更新，如果在事务启动时锁住该记录并直到事务结束时才释放该记录，那么这种做法不是一个好的设计形式。在这种情况下，所有其他的事务都必须要等待直至第一个事务完成为止。事务应用程序必须是一个能够尽可能快、并使用最少的资源来完成的一个程序。

> 事务处理同高速公路交通非常相似。假设在一个繁忙的高速公路上，有一边出现了一个交通事故，那么所有的车辆都会受到影响。当另一边的一辆汽车减速看热闹的时候，其后面的车辆都必须要刹车。很快，事故发生点后面的车辆会排起一条长龙。

19.2.9 面向对象的事务

面向对象技术已经进入了联机事务处理领域。事务处理正在同分布式对象进行融合。对象管理组织（Object Management Group，OMG）已经在CORBA 2.0规范中采用了对象事务服务（Object Transaction Service，OTS）接口来处理分布式对象。OTS允许跨越多个对象请求代理的分布式对象参与一个ACID事务。这种服务为对象事务和过程事务提供了互操作能力——如果它们遵循X/OPEN DLP标准的话。OTS为组成事务单元的对象定义了接口定义语言（Interface Definition Language，IDL）接口。对象和分布式事务的联姻是对面向对象处理的一

个自然扩展。

19.3 五大列表

分布式事务可以有许多不同的方法，并且使用了许多不同的技术来解决问题。当评价可能的解决方案的时候，总是需要考虑很多方面进行很多权衡。下面这个列表是组织在考虑分布式事务处理时应该考虑的前5个方面：

认证和授权。在分布式事务处理环境中，主要的考虑之一是确定需要什么样的认证和授权机制以及需要在什么地方调用它们。如果使用了一个公共的安全机制，例如DCE提供的机制，那么问题就基本上解决了。如果没有使用公共机制，那么你必须要考虑需要认证和授权的所有地方。接下来，你必须确定需要什么类型的认证以及用于授权的标识符。在执行事务的第一个位置，需要进行认证以保证事务的可信性吗？执行认证的每个分布位置都需要进行认证吗？标识会从事务处理系统的一个部分传递到另一个部分吗？

资源。事务系统所消耗的资源（内存和处理能力）是难以预算的，它取决于同时到达的事务数量以及完成一个事务所需要的时间。如果你准确地知道在给定的时间段内要处理多少个事务以及在任意时刻需要处理的事务量，那么制定准确的资源计划就是有可能的。当计划一个事务处理系统需要多少资源时，可以使用如下三条原则：

- 原则1——永远也没有足够的资源。
- 原则2——你的资源预算太低。
- 原则3——参考原则1。

回滚/恢复/重启动。只要事务分布到多个系统上，事务处理系统就必须实现这样的机制，在事务失败的时候回滚处理；在系统出现故障的时候恢复和重启动处理。这些功能应该构建于事务监视器中。事务处理的设计会受到恢复和回滚需求的影响吗？也许你可以把事务分割到多个逻辑工作单元中，这样可以更容易进行恢复。

事务调整。除了早期事务处理应用编程所使用的技巧以外，事务应用程序调整也是必不可少的环节。事务信息的性能和利用必须要得到重点关注。调整事务系统的一个简单办法是花钱购买更多的资源。事务调整就如同走迷宫一样，你必须要找到正确的出口。你需要有好的检查和分析工具，它们可以确定在基于事务的系统中会发生什么。当事务分布在多个系统上时，问题更加复杂。你必须要知道该怎么走！

管理。分布式事务的管理是系统管理领域中的一个挑战点。为对付该挑战，需要有特殊的工具来控制和管理环境，并且需要有覆盖分布式环境中所有系统的工具。当系统拥塞时，或者当一个不可用的资源正妨碍系统完成时，必须要采取某些动作来做出响应。监视器或者平台应该能够自动平衡多个事务监视器执行实例上的负载平衡。分布式处理系统的管理需要有可靠的决策。一个实时监视工具能够帮助确定系统中正发生的情况以及在发现问题后应该采取的措施。

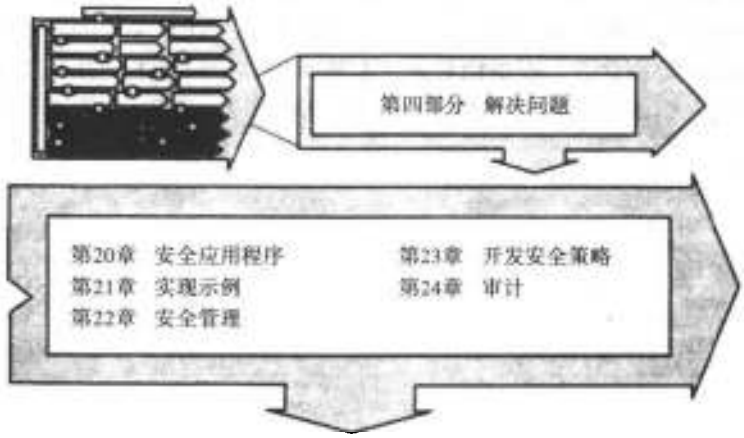
19.4 小结

OLTP机制的使用可以在分布式环境中提供一个强大的、健壮的事务环境。这可能是高卷

环境或者异构环境中的惟一选择。对象技术的发展为事务和事务处理带来了新变化。大型机系统上的联机事务处理利用了成熟的、可靠的产品，而用于分布式环境的事务处理系统还是市场上的新生物。

随着分布式环境的成熟，事务处理能力也会提高。关键是要使用一个基于标准的分布式处理基础结构。具有事务能力的数据库系统是OLTP系统的竞争对手，它的使用要多于OLTP系统的使用。从这一点上我们可以看出建立在工业标准上的解决方案要比基于专有系统的产品更具有用处。连接和管理这些系统的能力也在逐渐成熟。而分布式的面向对象事务也表现出了惊人的发展前景。

第四部分 解决问题



在前面的章节中，我们已经讨论了分布式计算环境中的安全性挑战。已经看到问题的规模和复杂程度。我们也已经探讨了依照策略、指导原则以及体系结构的形式来构建正确基础的必要性，并且还探讨了分布式计算中的许多关键技术。可以看到，技术不仅仅为计算安全性问题提供了解决方案，而且还加大了问题的难度。在后面的章节中，我们将要讨论用来解决这些问题的措施。

在第20章中，我们将点明这条根本原则，必须要一开始就将安全性设计到应用程序中。如果要为已经设计并实现了的客户机-服务器应用程序重新添加安全性，那么即使不是不可能的，也是非常困难的。我们还要探讨面向对象技术的应用以及该技术所带来的许多安全性问题。在第21章中，我们要探讨一下如何把安全性部署到基础结构应用程序中。特别要看一下安全性在电子邮件和工作组计算中的应用。

安全管理是第22章的重点所在。在这一章中，我们要探讨一下集中式安全技术的作用，并对安全管理的各个方面展开分析。很多人都使用电子设备来加强自己家的安全保护。电子防护设备可以为我们提供安全消息，比如电子报警系统，使用它我们信任自己的家是安全的。其后面的一些原因是：

- 当被触发时，它们能够自动发出警报声音，而不管我们是否在家。
- 非法入侵警报可以从许多入口处探查到。
- 它们可以召集其他人进行援助。
- 同窃贼带来的潜在损失比起来，它们便宜得多。
- 相对来说，它们的使用和实现是很容易的。

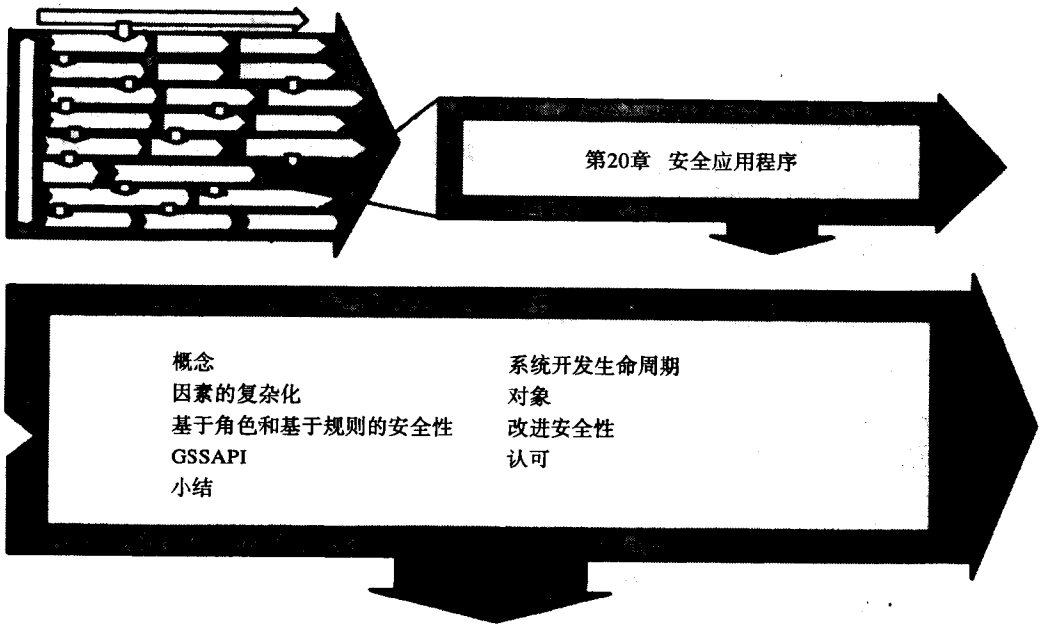
同在家中使用自动报警器一样，分布式计算环境也需要有类似的东西。当有人非法入侵我们的网络或者计算机系统时，如果有一个自动报警器来通知我们，那么这会极大地提高计算环境的整体安全性。一旦收到了安全警报，安全人员就可以来处理情况。

第23章将为读者提供一个方法学，我们要用其中的方法来解决全部计算安全问题。安全策略提供了一个分析安全措施的方法——这些措施会移动到（或者在某些情况下仅仅维护）组织的适当安全级别。

在第24章中，将探讨审计的重要性。审计可以保证组织对标准和指导原则的遵循和实施情况。不幸的是，在许多组织中，内部审计部门和信息科技部门之间往往存在着一种对抗情绪。我们会分析该问题的原因，并对二者之间应该具有的良好关系给出建议。

最后一章的内容是对前景的一些对立看法。我们知道，计算安全问题会在将来解决，有许多原因可以使我们相信此点。改进的技术、管理承诺、加密技术的使用以及其他许多因素都有助于整体解决方案的出现。另一方面，我们也相信问题本身也会有变化，并且会变得更加复杂和棘手。

第20章 安全应用程序



在设计和开发分布式计算应用时，必须要对安全性进行详细地考虑。大型机环境具有自己的集中式安全机制，它允许应用程序设计人员把重点放在应用程序本身的安全需求上。而在分布式系统中，应用程序的安全问题就复杂多了。分布式应用程序的开发人员不仅仅要为应用程序设计具体的安全机制，而且还要解决分布式环境的安全问题。客户机-服务器计算的出现已经促进了计算资源在物理上的分离，但同时也带来了另外的安全风险。

网络也为安全应用程序的实现带来了一些特殊挑战。如果在分布式应用程序的设计完成以后再为其添加安全性，那么这不但很困难而且成本也不菲。在本章中，我们会看到安全性在应用程序开发的每个阶段都是非常重要的考虑事项——从开始需求定义一直到开发和部署阶段。我们对设计安全应用程序的讨论将首先从几个关键概念开始。

20.1 概念

术语“可信计算基础”（Trusted Computing Base，TCB）是用来描述一个计算系统内的总体安全保护机制的。这个概念对于分布式应用程序的设计来说是非常重要的，其原因在于它让我们把注意力放在评估计算环境的安全性上，而不仅仅是应用程序本身。TCB要求设计人员必须要注意应用程序的非传统方面的安全性，如操作系统的安全性。在某种意义上，它把应用程序设计的范围扩展成包括操作系统和网络组件。为了掌握操作系统的整体安全性和确定应用程序功能的可靠性，TCB的概念是非常重要的。例如，对于一个UNIX应用程序来说，设计人员可能需要考虑很多附加的安全机制，以便对付特洛伊木马以及第11章中所介绍的其

他攻击方式。

分布式应用程序中的另一个重要概念是安全域。安全域定义为人或者处理程序可以访问的信息或应用程序集合，其中这些访问受到公共安全策略的控制。在一个分布式处理系统中可以有一个或几个已定义的域。在系统的设计阶段，确定或者明确这些安全域是非常重要的。安全域之间的互操作性是由覆盖整个域的安全技术、策略以及标准确定的。我们需要知道是否存在独立的安全域以及操作和安全化域间的访问需要做些什么。

20.2 系统开发生命周期

许多组织已经采用了一个正式的方法或者途径来开发计算应用程序，这种方法称做是系统开发生命周期（System Development Life Cycle，SDLC）。SDLC定义了管理计算系统开发项目所需的一些过程和转折点，从项目开发的第一个概念到实现再到应用程序的操作都受其控制。这种形式化的过程有时候是根据经验形成的，从出版的资料中吸取而来，或者在某种情况下是取决于开发技术的。在任何情况下，使用生命周期过程都必须要考虑安全性问题。

实现计算机应用程序可以有许多不同的途径。应用开发的经典方法有6个主要部分：初始需求定义、分析和设计、应用软件的开发、测试、实现以及应用程序的操作和维护。

图20-1概括说明了系统开发周期过程。

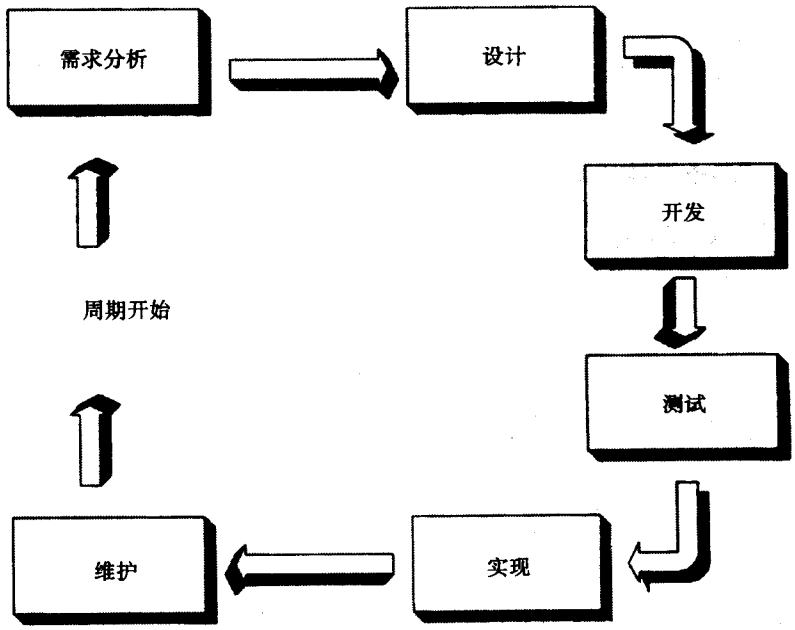


图20-1 经典的软件开发生命周期

另外还有其他一些开发方法，如快速应用开发（Rapid Application Development，RAD）试图通过使用高级软件开发和原型工具来减少配置时间。RAD主要着眼于缩短开发阶段，它允许用户积极参与项目开发的所有阶段。然而，不管使用什么方法或技术，SDLC的基本概念仍然是有效的。

20.2.1 需求阶段

需求阶段的目标是收集和文档化应用程序所涉及到的业务需求。这包括调整应用需求使其符合组织的业务目标，并明确用户部门的需要。本阶段也要明确应用程序的整体性能需求和成本目标。我们认为应用程序的安全性目标也是必须概括出来。很多安全目标可以从公司的安全原则和策略中获得，详细情况参见第4和第6章。

安全和审计部门的积极参与是非常重要的，我们强烈推荐这么做，特别是在大型的或者关键任务的应用程序开发中更应如此。安全和审计部门的作用是保证应用程序会满足公司的安全和审计目标。在应用程序开发过程中，审计和安全部门应该尽可能早地涉入，这样做有很多好处。第一，应用程序是否满足公司的策略和安全原则，这是由审计和安全部门来最终判定的。所以开发人员应该尽可能早地征询他们的意见。第二，保证应用程序的安全可能会影响应用程序的开发进度和成本。最后，如果在项目早期没有解决好安全需求，那么问题就会一直拖延下去。许多组织都在应用程序开发完毕，甚至是在部署了应用程序以后，才发现了安全需求根本没有被满足。在这种情况下，重新为应用程序添加安全性不仅非常困难，而且成本不菲。惟一的工作就是战战兢兢地向管理人员要钱，并且一次又一次。

整体安全需求可能包括如下需求：确定用户的标识（认证）、保护公司或者个人信息免遭未经授权的泄漏（机密性）以及记录用户动作（审计跟踪）。如果存在的话，公司计算策略和原则将是整个安全需求的最好来源。

20.2.2 设计和分析

在设计和分析阶段，开发人员要根据成本和时间约束来分析应用程序的需求，即在应用程序的总体需求和所花费的最小成本之间取得一个平衡。在系统设计阶段解决应用程序的安全需求，效率更高并且成本更低。如果要在这步以后才添加安全性，那么通常不但难以实现和维护，而且成本也很大。这一阶段是安全原则、策略以及体系结构的切入点。我们可以把这些元素用做核对安全需求的参考蓝图。它们可以帮助我们决定需要包括的内容和函数的位置。设计阶段也需要考虑提供审计信息。最后，考虑也必须经过选择的安全机制的性能。

风险分析是设计和分析阶段的一个预备行动。该过程着眼于在安全暴露的预期后果和其发生的可能性之间取得一个平衡。这使得应用程序设计团队必须要将注意力放在那些具有高风险的安全问题上，这些问题可能会为组织造成重大损失和危害。例如，如果可以拨号访问一个应用程序，那么攻击者通过窃听线路来探寻密码的可能性有多大呢？如果攻击者窃取了密码，他就可以非法访问该应用程序，那么后果会怎样？

风险分析完成以后，应用程序设计团队就可以开始解决前一阶段中定义的安全需求了。同样，这一过程也是一个需要找到某种平衡的过程。例如，如果机密员工信息保存在一台膝上机上，那么应该采取什么控制措施来保护这些数据？应该使用软件来提供访问控制（如用户ID和密码）以保护这台膝上机吗？数据需要加密吗？如果需要，应该选用什么类型加密算法？什么强度的密钥？应该使用硬件设备来保护加密算法及其密钥吗？进行风险分析有助于回答这些问题。如性能和成本之类的约束肯定会极大地影响设计选择。

在客户机-服务器应用程序中，设计人员必须要解决如下问题：安全环境应该如何综合？个人计算机上需要施加控制来预防对该机器的未经授权的访问吗？应该使用加密来保护个人

机器上保存的数据吗？如果个人计算机上使用了加密，那么需要使用附加的硬件（如智能卡）来保护加密算法及其密钥吗？客户机和服务器之间的网络有多可信？网络流量需要加密吗？是加密全部流量还是只加密选择过的数据字段？服务器上需要有另外的操作系统或者数据库控制吗？需要什么样的审计跟踪，应该保存在什么地方？公司计算策略和原则可以提供这些问题的答案。审计和安全部门的积极参与也是值得推荐的。我们知道，回答每一个可能的安全问题和暴露是不可能的。最好的方法就是使用可用的资源来补偿暴露带来的损失，并让管理人员来决定是否要批准另外的资源。

20.2.3 应用开发和测试

在部署一个应用程序之前，一定要严格测试其安全控制。测试目标包括中间件、操作系统、数据库以及网络控制。测试的目标称做风险评估。对应用程序、数据库或者操作系统的未经授权的访问是否被允许，是由风险评估来决定。如果可能的话，测试人员不应该是开发团队中的成员。如果测试人员同开发人员毫无联系的话，他就不会对需要测试的环境做什么假设，因而更有可能发现开发团队没有注意到的安全问题。敏感的应用程序还需要有关人员对实际应用代码进行结构化遍历。代码遍历可以用来检查应用程序远和设计人员所忽视的问题。

也应该评审测试数据的使用。测试环境和生产环境必须要分离开。测试数据绝对不能进入生产环境。应用程序员和开发人员也绝对不可以通过测试行为来访问正常情况下他们不能访问的敏感信息。

20.2.4 实现

应用程序开发的安全考虑可以分为几类。第一个是应用程序本身的保护，第二个是安全功能在应用程序中的实现，第三个涉及到了测试和生产环境，最后一个同应用程序的维护相关。应用程序的保护可能是最容易解决的。那些具有已定义需要的人可以访问应用程序。

一般来说，把具体的安全功能结合到应用程序中不是一个好方法，除非没有其他可用的替代方案。具体的密码或者其他认证服务不应该编码到应用程序中。应用程序不允许在一个特权状态下运行。如果出现了错误或者bug，那么用户不应该被留在一个特权状态下，否则他就可以对应用程序或工具进行未经授权的访问。应用程序的运行应该遵循最小特权原则，这样，才会给用户只有那些绝对需要的特权级。

应用程序测试环境和产品环境应该是分离的。测试应该在不会改变或者影响产品系统的前提下进行。无论什么情况，实际的产品数据绝对不可用于应用程序测试。否则，产品数据的完整性可能非常危险。一旦系统通过了测试，开发人员应该调用一个形式化的改变过程，从而把该系统实现到产品中。应用程序开发、测试、实现以及操作中所牵扯到的责任需要分离开，这是一个重要原则，但它不会正常应用。从根本上来讲，这一原则说明了谁也不可以改变应用程序并把变化实现到产品中。这些责任应该明确定义，从而限制任何人未经授权的访问或者修改产品系统。

应用程序中的某些错误可能会为系统带来安全影响，对这些错误的测试也是非常重要的。黑客利用的大量安全漏洞都是一些软件错误或者bug所造成的结果，这些错误可以打开进入系统其他部分的大门，甚至利用操作系统的命令。用不常见的情况来测试应用程序可以降低这

些安全暴露问题所造成的风险。

20.2.5 维护

对于一个分布式应用来说，一个颇为头疼的问题是对分布化和实现改变的需求。该问题让人头疼的程度取决于必须要改变的应用程序的数量和地理位置。如果要保证所需的全部改变都会正确且及时实现，那么必须要有一个形式化的改变管理过程。通常这意味着所有的改变都必须同时实现，或者至少是所有位置都以一个一致的方式来实现。该变化管理过程也必须要保证只有经过授权的改变才能实现。在这里，改变过程的自动化是非常有帮助的。

改变管理应该由一个策略来控制，该策略需要规定实现改变和过程所需要的责任。改变管理过程的安全性应该在系统的设计阶段进行考虑。攻击者应该不能使用改变管理系统来分布未经授权的软件，例如特洛伊木马。只有对处理逻辑的变更才能需要进行应用程序改变。当授权表更新或者安全原则改变的时候，应用程序本身不需要进行改变。系统中应该选择使用可以同其他安全机制进行互操作或者进行结合的安全标准，这是一个非常好的做法。

20.2.6 认可

系统是否满足一个定义良好的、通过正式过程测量的安全标准，这是由系统认可来正式认可的。认可程序的结果是对风险的一个正式认可。认可程序同用来验证安全标准是否满足的质量控制过程非常类似。在移到产品中之前，每个系统和应用程序都应该交给一个认可过程来评估。认可过程将检查系统的安全组件以判断是否支持设计需求和标准，以及是否是解决问题的正确方案。认可过程应该定期进行，特别是在系统中进行了大量的变化以后。

一个正式认可过程的组件包括了质量保证测试和证明。质量保证测试运用控制来测试它们在技术上是否正确和有效。证明过程是测试所实现的安全机制是否满足已制定标准的正式方法。系统风险越危险或者越大，证明过程就应该越正式。该过程应该由一个同设计或者构建系统没有关系的授权体来执行。系统开发生命周期中应该包括质量保证计划。

20.2.7 GSSAPI

我们需要把安全服务集成到应用程序中，这种需要是可用的。现在，已经提议了用于在应用程序和所需安全功能之间提供链接的一套标准，即因特网工程工作小组（Internet Engineering Task Force）的网络工作组（Network Working Group）所提议的通用安全服务应用程序编程接口（Generic Security Service Application Programming Interface，GSSAPI）。该标准是在RFC 1508中描述的。GSSAPI为客户机-服务器应用程序提供了一套同安全功能进行交互的通用接口。从应用程序中使用很多API调用，很多安全机制都可以同GSSAPI一起使用。管理指定安全机制的责任可以从应用程序中删除。GSSAPI是独立于安全机制和协议的。它使得应用程序可以在不同环境之间保持源代码级的可移植性。

使用GSSAPI实现的应用程序可以同具有DCE（Distributed Computing Environment，分布式计算环境）提供的安全功能的应用程序进行交互。使用GSSAPI，应用程序可以同DCE安全服务器进行通信，从而允许应用程序不必非得成为一个DCE节点。客户机平台上的GSSAPI用来请求和管理DCE安全服务器和目标应用之间所需的安全凭证。GSSAPI可以用来为分布式系统中的通信、数据来源认证以及数据完整性提供保密性。也可以扩展GSSAPI的使用为WWW

应用程序提供安全，这是一项正在开展的工作。

20.2.8 对象

在第9章中，我们已经探讨了一些分布式对象的知识，并介绍了一些安全性问题。在分布式面向对象应用程序中，最大的挑战之一在于保证对象能够得到所需的保护机制。面向对象系统的主要特性之一是它具有把客户机对象的功能委托给目标对象的能力。执行路径可能并不总是可预测的，目标对象本身可以成为一个客户机对象。正常情况下，可以分布到多个平台上的对象并不带有用户概念。为了保护对象免遭未经授权的访问，我们需要有一个方法来提供对象间的信任。

请求应用服务的人或者程序需要指定一个惟一的标识符，通常情况下，这种标识符所使用的是一个用户ID或者一些其他方式。另外，它还需要一个与其相关的特权列表。在第9章中，我们看到在分布式对象应用程序中，需要有一个对象请求代理来代理对分布式对象的访问。管理对象间的关系也是ORB的责任，这样做的目的是保持所需的保护级别。我们需要保证，不管如何调用对象，对对象的访问都必须是经过授权的。当认证用户或者系统时，建立的特权就需要委托到可能执行的所有对象上。

处理分布式对象安全的标准和技术还没有达到其他分布式技术的成熟程度。当考虑分布式对象处理时，我们需要考虑对象框架，它提供了最丰富的安全功能级别和最大的灵活性。公开密钥技术和证书及认证中心的使用非常有助于定义对象框架所需的功能和性质。ORB应该能够利用不同的安全技术，同时要把这些技术的差别向应用程序隐藏起来。

对象的引入为标识、认证和授权过程都带来了新挑战，这些过程通常都是同用户相关的。对象一般没有用户概念。如果对象分布到多个位置或者多种平台之上，那么情况会更加复杂。数据库系统通常具有自己的安全系统，它们可以很好地保护数据库。当数据库被事务或者面向对象系统访问时，这会变得不明显。当使用对象技术时，由于安全问题的复杂性，审计和安全部门最好参与整个应用程序的设计过程。

20.2.9 基于角色和基于规则的安全性

授权通常是使用个体或者请求授权的过程的标识来进行的。这意味着用于授权的机制必须要访问一个定义这些标识的列表以及与它们相关的特权。维护标识和特权列表会极大的增加管理难度和工作量。由于员工或者责任的变化而导致对这些列表的添加或者修改是必须要注意的。

允许授权访问的另一种方法是使用已定义的角色来代替个体标识。角色是由工作责任而不是由用户定义的，并且是面向组的而不是面向用户的。授权访问是根据一个角色组的成员关系而不是根据独立个体标识来授予的。这种方法可以减少整体管理需求，并且可以把定义所需安全性的责任从系统管理员移到已定义的策略和指导原则上。特权不再是根据个体来维护的，而是由角色来定义。个人责任的改变只会简单地反映到不同的角色上，而不是改变可以维护授权列表的所有地方。

在业务规则的基础上提供授权是对基于角色的安全性的一个扩展，它也代替了独立个体的使用。普通的授权决策是建立在对个体是否能够访问受保护资源的判断之基础上的。是否允许对受保护资源的访问，可以根据请求的类型或性质来决定，而不是仅仅根据请求来决定。

例如，一个用户可以处理1000元以下的事务，而另一个用户可以进行10000元以下的事务。通过使用特殊的程序代码来处理所需的访问规则，许多应用程序已经提供了这种功能。如果所有需要授权处理的地方都必须使用和维护规则，那么这可能会变得非常笨重。

分布式系统中最合适的响应是，使用一个集中式授权服务来授予访问权，授权所根据的是应用到请求个体或角色上的业务规则。这些规则可以结合到授权处理中。例如，惠普提供了一种集中式授权服务，它可以在已定义的业务规则的基础上对分布式请求进行授权。在这种情况下，就没有必须要使用应用程序代码来应用规则。这些规则可以在不修改应用程序代码的情况下改变。临时改变授权可以不进行费事的额外授权工作。这种做法可以在无需额外努力的情况下提高安全性级别。

20.2.10 重新添加安全性

通过本章的介绍，我们已经知道了安全性必须要一开始就设计到应用程序中。如果没有这么做，那么后面就有可能再对产品应用环境重新添加安全性。在这种情况下，产品环境可能会变得支离破碎。例如，解决网络安全问题的一种途径是在客户机和服务器之间引入硬件加密设备。这些设备（黑箱）可以对客户机和服务器之间的数据流量进行加密。同时，客户机或者服务器上代码不需要改变。对基于LAN的网络中的网络流量进行安全性保护的另一种途径是，截获LAN卡和应用程序之间的流量。在这种情况下，应用程序和网络之间需要有一套软件解决方案，该软件可以无缝地截获和解密客户机和服务器之间的应用流量。

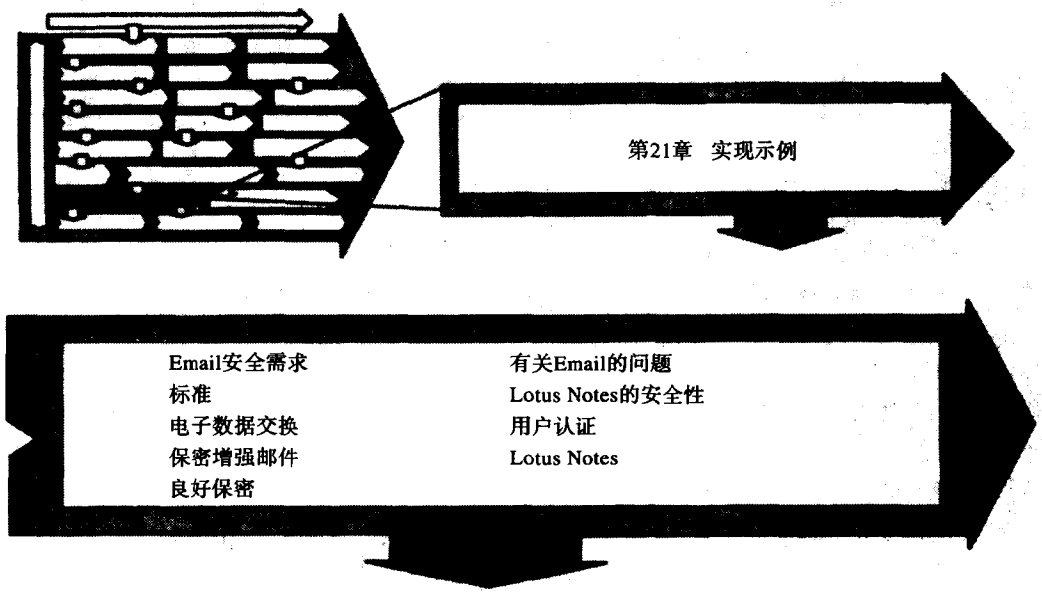
对应用系统重新添加安全性实在是下下之策。尽管确实有相应的技术解决方案，但是对安全性问题还是要尽可能早地解决。如果在早期没有制定良好的计划并及时行动，那么解决问题的整体成本就会变得很大，并且解决问题的思路也会被限制得很窄。

20.3 小结

在今天的客户机-服务器环境中，对系统重新添加安全性不是一个简单任务而且成本巨大。同大型机环境不同，分布式环境中的应用程序通常不能进行独立的全面控制。在应用程序设计中，安全性必须要尽早的包括进来，它是应用程序开发过程的一个核心组件。

计算机应用的安全性考虑包括了应用程序的安全开发和管理，以及应用程序所需的安全工具。系统开发完毕并不意味着安全考虑就结束了。对任何具有具体需求的计算机系统来说，安全性必须要在设计阶段就考虑，并一直持续到开发阶段。由于安全问题有很多不同的解决途径和技术，所以挑战就在于如何把正确的技术和方案集成到具体的应用中。在下一章中，我们将看一下如何把安全性带到大量的分布式应用程序中。

第21章 实 现 示 例



前面我们已经花了大量的篇幅来讲述分布式安全环境中的问题以及复杂性，现在我们来看一下如何实现这种分布式安全环境。电子邮件系统是一种很常见的应用，它必须要在一个安全性、分布式的工作方式下操作。群件系统是对邮件系统的扩展，它起到了分布式知识支持系统的作用。它的出现为计算机的使用带来了巨大的影响。在本章中，我们要探讨一下分布式电子邮件系统中的安全性问题，并介绍一些正在出现的技术和标准。我们还要看一下信任问题在一种主要的群件产品——Lotus Notes——中是如何实现的。

下面我们定义一个新概念——基础结构应用程序——在分布式处理环境中使用一个技术和标准框架来执行其功能的应用程序。换句话说，这些应用程序在系统之间进行通信和协调以完成其任务。为达到此目的，我们需要一个公共接口，或者一个能够同所有不同的平台和邮件应用程序进行通信的接口。

21.1 电子邮件

电子邮件或者消息系统已经面世很长一段时间了。它们从简单的消息存储和转发系统一直发展成熟到全功能的分布式应用。早期的邮件实现提供了一个集中邮箱仓库来存储消息，并且提供了一个存储和检索消息的方法。每个用户都指定有自己的邮箱。消息的邮递非常简单，就是把消息保存在另一个邮箱中。用户通过中心邮件应用程序来格式化、投递以及检索邮件消息。只要每个人都使用同一个邮件系统，这种方法就能正常工作。

简单的消息应用程序也迁移到了PC/LAN环境中。这些应用程序服务于本地用户，它们被用做部门邮箱系统。随着需求的增长，人们把这些部门系统同其他邮件系统连接起来，进而

同其他组织的系统进行互连。如果所使用的邮件产品来自于同一厂商并支持同一标准的话，那么问题就相对简单了。如果涉及到了不同的产品，那么解决方案就很困难了。一些流行产品提供了转换网关来在其产品之间移动邮件，这样就解决了一部分问题。

当处理邮件系统互连时，一个主要挑战是为电子邮件系统提供安全性。如果要在网络上发送机密消息，那么必须要保证传输的安全性以免消息被窃听。应该对消息发送者进行认证，从而证明消息的来源，并防止发送者标识被伪造欺骗。消息的发送也必须要保证消息的完整性，并且能够检查出消息的任何改变。为保证消息传输的准确性（由预期的接收者收到和读取），消息系统应该提供传输证明。

有两个标准电子邮件（e-mail）体系结构可用于直接向用户提供电子邮件服务并互连专有电子邮件系统。第一个是著名的X.400标准，它定义了一个独立于厂商的邮件标准。该标准包括了一套完整的安全机制规范。第二个称做因特网邮件体系结构。该体系结构的安全功能没有X.400中的安全功能成熟。因特网邮件体系结构的保密增强邮件（Privacy Enhanced Mail，PEM）选项在1993年成为可选的，它也提供了安全功能。良好保密（Pretty Good Privacy，PGP）是一个公开密钥加密工具，它在因特网邮件应用中具有非常高的流行度。后面我们将详细介绍这些安全组件。

21.1.1 电子邮件安全需求

今天的电子邮件系统被用来在组织内和组织间进行通信。为了支持这种外部的通信需求，邮件系统之间需要使用一个安全方法来互相发送消息。表21-1概括出了支持这种安全消息交换所需要的一些需求。只有解决了所有这些需求，邮件系统才能是可信的。

表21-1 电子邮件安全需求

需 求	说 明
提交证明	验证消息已被邮件系统所接受
传输证明	验证消息已被传送给邮件接受者
机密性	验证消息内容的私密性，被发送的消息可以保持内容私密性
来源证明	能够证明消息发送者标识的能力
匿名	能够发送一条消息同时不让接受者发现发送者的标识的能力
审计	系统能够记录同安全相关的事件，从而在将来进行分析
自销毁	消息发送者可以指定，接收者收到并读取消息以后，消息应该销毁
认可	防止发送者或接收者对消息的发送或者接收进行否认
消息完整性	保证一条消息或者一组消息会以发送的顺序到达目的地，并且不会损失任何数据
密钥管理	在多个邮件系统参与方之间协调私有密钥，同时不会危害密钥的安全

21.1.2 标准

同分布式计算中的其他领域一样，也有几个有关分布式电子邮件服务的标准。在这些标准中，有些是互相补充的，有些是互相竞争的。随着分布式技术的成熟和我们对电子邮件技术的关注点的变化，这些标准也在不断变化。下面我们将简要介绍当前分布式邮件服务所涉及到的一些标准。

1. X.400邮件服务

X.400标准是国际标准化组织（ISO）和国际电信联盟（International Telecommunications

Union，ITU）联合定义的，它定义了一个独立于厂商的邮件标准。网络应用程序使用该标准来在应用程序（例如email）之间存储和转发消息。对于大多数公共服务或者军事消息应用程序来说，同X.400规范兼容现在已是一个标准要求。大多数邮件厂商都支持X.400标准，或者推出了X.400兼容产品。该标准提供了对大型二进制对象（Binary Large Object，BLOB）的支持，从而允许应用程序交换图像、传真或者其他可以附加到邮件消息上的二进制文件。X.4090规范包括了X.500标准所定义的目录服务。X.400的安全性是在X.509标准的一部分中制定的。前面第15章中已经提到过了X.509标准，该标准制定了用于密码标识、加密、数字签名以及审计的机制。

2. X.400安全性

X.400标准的安全功能是非常全面的，它被其他开放式系统应用程序用做一个参考模型。X.400体系结构包括了一个称做用户代理（User Agent，UA）的逻辑级别和一个更低的称做消息传输系统（Message Transfer System，MTS）的级别。MTS是由多个消息传输代理（Message Transfer Agent，MTA）组成的。UA执行所有的同用户联系的消息服务。MTA提供消息存储和转发服务，它可以把消息从一个UA传送到正确的目的UA。X.400标准已经进行了扩充，以便支持电子数据交换（Electronic Data Interchange，EDI）事务（X.435标准定义）的传输。

X.400安全消息功能包括了19个服务元素，这些元素可以分为5组：基本的端到端服务、消息路径服务、MTS协作服务、认可服务以及安全管理服务。当前该标准没有指定任何加密算法，公开密钥或者私有密钥加密机制都可以使用。X.400标准为实现提供了大量可用的安全选项。这些选项包括了消息来源认证、传输证明、内容完整性、内容机密性、对等实体认证以及几个认可服务。

3. X.500目录服务

最基本的问题之一是找出东西的位置。X.500目录可用来连接和组织来自不同地点的信息，这对用户和应用程序来说都是很有用的。这种能力是由国家标准化组织（ISO）和国际电话与电报顾问委员会（Consultative Committee for International Telephony and Telegraphy，CCITT）联合定义的一套标准定义的。X.500目录是由一个同面向对象模式非常类似的模式所定义的，它提供了整体逻辑结构的一个映射。X.500目录的基础结构是一个树型结构，靠近树的顶端位置保持了全局或者公共信息，树的底端则保存了对象的更详细信息。沿着通往根的路径而下，下面对象继承了上面对象的属性。实现了以后，X.500目录看起来像一个目录，而实际上它可能使用了分布在多个平台和位置上的多个目录。

4. 售主无关性的消息接发

售主无关性的消息接发（Vendor-Independent Messaging，VIM）是一个由多家软件厂商所支持的消息接口，其中包括Lotus、Apple、IBM、Borland、Oracle、WordPerfect以及Novell。VIM设计用来提供一个跨平台的消息接口。它是一个用于从各种客户机环境直接连接邮件引擎的规范，使用VIM可以代替网关。消息层实际上是各种邮件引擎的一个部分。独立厂商在其传统平台上提供了对VIM的支持。VIM由55个支持简单邮件、消息存储和目录服务的API调用组成。尽管VIM的名字表明它是一个售主无关性的标准，但实际上它只代表非常少的消息系统厂商。

5. 消息应用编程接口

消息应用编程接口（Messaging Application Programming Interface，MAPI）是微软公司

对电子邮件标准的响应。它是一个基于Windows的数据链接库（Data Link Library，DLL），该链接库执行客户机服务，并通过服务提供者接口（Service Provider Interface，SPI）同其他邮件处理引擎进行通信。MAPI的基本功能，称做简单MAPI，包括了一系列的12 个API调用，这些API调用可以激活邮寄、存储和提供目录服务的公共服务。MAPI的高级功能称做扩展MAPI，它提供了另外一套具有附加功能的API调用。这100多个API调用提供了对更复杂的消息和功能以及附加邮件服务的支持。操作系统（例如Windows for Workgroup）所带有的MAPI客户功能提供了自由的邮件访问，但是它可能限制分布式环境中的跨平台选择，其原因在于MAPI是Windows环境所特有的。

6. 公共邮件调用

公共邮件调用（Common Mail Call，CMC）是由X.400 API协会（XAPIA）定义的，它是为VIM和MAPI提供支持的一个方法。CMC接口在消息服务和启用邮件的应用程序之间定义了公共消息调用。Microsoft和Lotus都为应用程序开发人员提供了现成库，通过这些库，开发人员可以避免为VIM或者MAPI维护多个独立的邮件产品版本。CMC规范为简单邮件功能提供了10个API调用。使用一个类似于CMC的接口标准可以把可用的功能缩减到一个公共子集，从而不必支持VIM和MAPI所提供的许多扩展功能。

7. 简单邮件传输协议

简单邮件传输协议（Simple Mail Transport Protocol，SMTP）主要是在UNIX域中所持的。现在，它是因特网上的主要邮件传输规范。该协议已经成为黑客的几个最大攻击点之一。使用SMTP的邮件程序都已经成为几种攻击的受害者，这些攻击所利用的安全漏洞是由邮件程序的调试功能所提供的。这些漏洞包括：使用UNIX邮件程序转发密码文件的拷贝、使用调试过程中的缺陷来获得超级用户特权以及使用高级特权来保存文件。

8. 多用途邮件扩展

Multipurpose Mail Extension（Multipurpose Mail Extension，MIME）是一个因特网标准，它定义了消息类型（不是ASCII文本）如何使用一个因特网邮件消息来传递。内容定义的类型包括文本、多部分、消息、应用程序、图像、音频和视频。其中，因特网RFC 1521定义了文本部分（第1部分），RFC 1522定义了非ASCII文本部分（第2部分）。超文本传输协议（Hypertext Transfer Protocol）使用MIME来描述文档的类型。基本的MIME规范没有包括具体的安全保护。

因特网RFC 1847，即MIME安全部分，定义了可以应用到MIME部分消息上的安全服务接口。RFC定义了两种新的MIME消息元素类型，签名多部分和加密多部分。前者指定了如何通过使用数字签名来支持认证和完整性，后者指定了如何通过使用加密来保持消息部分的机密性。RFC并没有指定具体的安全标准，而是描述了MIME如何同其他已有安全协议进行对接的一个框架。

另一个因特网RFC（1848），MIME对象安全服务（MIME Object Security Service，MOSS），定义了一些安全协议，它们可以应用到RFC 1847定义的多部分框架上。实际上，MOSS是主要建立在RFC 1421所定义的保密增强邮件标准上的。MOSS服务是基于公开加密技术的。消息的生成者使用一个私有密钥来为消息创建数字签名，并使用接受者的公开密钥来加密消息以保证消息的机密性。消息的接受者使用其私有密钥来解密消息，并使用发送者的公开密钥来检验消息的数字签名。MIME安全性的使用使得不同的消息内容类型或者主体部分

都得以保护，并且允许使用不同的安全服务来保护消息组件。

21.1.3 电子数据交换

电子数据交换（Electronic Data Interchange，EDI）的出现为电子信息应用带来了革新。EDI可以描述成电子形式商务文档的移动，它可以支持业务关系。许多生产商改变即时清单系统的目的就在于试图支持订单和确认信息的电子交换。例如在某些情况下，汽车制造对零件或者材料的需求是以小时为单位测量的。EDI是网络消息系统的特殊业务实现，它设计来支持这种快速定购和发货需求。例如，汽车制造商向零件提供商发送一个订单消息，然后零件提供商返回一个发货通知来确认请求者。发票可能也顺道一起发回来。

当前有很多用于EDI的标准，它们通常都是基于具体工业实现的。一般来说，这些标准可以分为三组。联合国/技术文档交换（United Nations/Technical Document Interchange）主要用来支持欧洲EDI。EDI标准中的ANSI X.12标准是由美国国家标准委员会（ANSI）认可的。用于管理、贸易以及商务的电子数据交换（Electronic Data Interchange for Administration、Trade and Commerce，EDIFACT）是正在兴起的国际标准。X.435标准是对X.400的EDI消息服务增强，它允许邮件和EDI事务共享同一个邮件主干。X.435同信封类似，它封装了EDI事务消息。X.435提供了扩展的安全功能，这包括事务通知的证明和认可、接收内容的证明以及内容来源或接收的认可。当你使用电子系统进行购物的时候，需要确认订单及其内容的真实性。

21.1.4 保密增强邮件

保密增强邮件（Privacy Enhanced Mail，PEM）是一个因特网标准草案，它定义了消息加密和认证过程，从而在因特网上提供安全邮件服务。该标准规范包含在因特网RFC 1421文档的4个部分中。第1部分——消息加密和认证过程，第2部分——基于证书的密钥管理，第3部分——算法、模式和标识符，第4部分——密钥证书及其相关服务。PEM设计来同已有的邮件系统一起工作，但主要是使用SMTP定向到邮件系统。因特网邮件是建立在底层的文本-字符串传输系统之上的，而不是建立在透明的二进制传输系统之上的（于X.400不同）。这使得PEM有必要在常规的加密功能之外再包括一些特殊的处理步骤。

PEM为三种类型的消息提供了安全，它主要设计来同已有的电子邮件系统一起工作。一个PEM消息基本是一个由PEM保护的正常邮件消息。简单消息完整性检查（Messge Integrity Check，MIC）消息提供了完整性和认证检查，但没有提供任何机密性检查。另一种类型的PEM MIC消息同简单MIC消息提供了一样的服务，但是它包括了一个附加的编码步骤，从而允许消息可以通过多种邮件网关。这类消息可以保证完整性和签名信息不会被篡改。最后一种消息类型为完整性组件和认证组件添加了机密性。多种私有和公开密钥加密算法都可用于完整性和认证保护。当前，使用加密的消息类型只使用DES加密算法。

21.1.5 良好保密

这是一项名字奇怪来历更奇怪的技术。一个名为Phil Zimmerman的计算机程序员开发了一个加密程序，他有意把该程序用做共享软件。Zimmerman写这个程序的意图在于为计算公众提供强大的加密能力。不幸的是，PGP（Pretty Good Privacy，良好保密）的早期版本使用了RSA加密算法，这侵犯了RSA的专利权。该程序的一份拷贝被传到了因特网上。因为该程

序是即时可用的，因此它迅速成为加密安全的非正式标准——尽管这在技术上是违法的。该程序的合法和受支持版本现在也已经有了，但是它们仍然受到美国出口限制的约束。

良好保密是一个通用的加密程序，它使用公开密钥加密来提供保密和认证。由于它的灵活性和相对较低的成本，PGP正逐渐变得很流行。它为电子邮件提供了机密性、数据源认证、消息完整性以及源服务认可。它设计来同现有的电子邮件系统一起工作，主要是那些在因特网使用的系统。国际数据加密算法（International Data Encryption Algorithm，IDEA）用于数据加密，RSA算法用于加密密钥管理，MD5（消息摘要5）或者RSA用于消息完整性检查和数字签名。相对来说，IDEA算法是比较新的，它是由瑞士开发的一个128位密钥算法。

PGP一直处于侵权和出口限制的争论之中。PGP的早期版本有潜在的专利冲突危险。PGP的2.6.1版使用了RSAFEF工具箱进行公开密钥加密，对于非商业应用来说它是合法的，但仍然受到出口限制。一个称做2.6ui（非官方国际）的PGP版本，存在于美国境外，它违反了RSA专利。PGP 2.4版是一个商业版本，完全许可它用于商业目的。尽管PGP的起步有点蹒跚，但是现在它已经广泛使用以保证因特网邮件的安全性。

21.1.6 同电子邮件有关的问题

为解决分布式邮件系统所带来的挑战，业界已经进行了大量的努力。但不幸的是，许多问题依旧存在。下面所列的是在调查分布式邮件安全的时候应该考虑的一些问题。

- PEM同X.400安全消息传递不兼容。这两套端到端安全功能不能互连网。因特网邮件协议不同于X.400邮件协议，但是二者共享同一个底层体系结构模型。一个用于在X.400和因特网邮件环境之间进行互连的规范已经开发出来了。
- PEM是基于层次化组织概念的，而PGP是基于独立分布式网络的。PEM更适合于公司和组织使用，而PGP则对使用因特网邮件的个体用户更为有用。
- 对于许多用来提供安全的加密机制来说，加密密钥证明和分发是一个问题所在。应该如何提供这种服务呢？由谁来提供呢？
- 如果使用不同的邮件文本应用程序，那么一定要小心谨慎。例如，空格的对齐或者插入可能会影响消息完整性检查。
- 数据加密机制同数据压缩算法不可能兼容。
- 加密算法是否受到出口限制？是否有可能侵犯其他产品的专利？这一点务必要注意。早期版本的PGP就是一个好例子。
- 使用DES算法的私有密钥加密要比公开密钥加密算法（例如RSA或者PGP）使用更少的处理资源。

21.2 Lotus Notes

在第9章中，我们把群件引入到分布式系统的讨论中。Lotus Notes是一个非常复杂的群件产品，它更像一个计算机会议产品而不是一个邮件系统或者分布式事务处理器。该产品是由Iris Associates同Lotus在80年代末期合作开发的。Lotus Notes由以下组件所组成，一个文档数据库服务器、一个邮件服务器、一个用来路由邮件和服务的服务器基础结构、一个客户机GUI环境、安全服务、分布式服务以及一个应用程序开发环境。Lotus Notes的基础概念是在一个已定义的用户组中，以一个良好的组织方式、协调地共享文档。

在Lotus Notes中，文档和数据库的定义同其常规定义有所不同。文档概念是Notes的基础，它指的是把多个多媒体元素（数据字段、文本、图形、音频和视频）组织在一起的一个定义良好的单元。这些相关文档的集合存储在数据库中。检索文档可以使用任何标识的文档属性或者文档内容。文档数据库可以使用一个辨识图标和标题提供给用户，并包括帮助面板和说明数据在内。一个由数据库组成的集合构成了Notes服务器上的文档存储器。

Notes服务器和Notes客户机通过复制来共享Notes数据库，并定期同步化数据库以保持一致。Notes中没有主服务器概念。复制机制在数据库复制品之间以双向模式运行，从而保证所有的复制数据库都是同步的。复制动作可以以一个经过选择的频率来进行，并在一个后台过程中无人值守操作。复制的目标可以是全部也可以是部分文档。同步化复制的数据库有两种方法。基于服务器的复制——通常是无人值守特征的调度事件，基于客户机的复制——通常由用户请求发起。移动用户可以把复制的数据库存储在他的笔记本电脑上，当用户登录时，文档就会被同步。文档的不同更新版本标记成对主文档的响应，从而避开了并发更新问题。实际上，最近的响应可能不是最新的。

21.2.1 Lotus Notes的安全性

Lotus Notes的安全性是以三种方式来提供的。Notes提供了安全机制来保护对Notes文档的访问、对Notes文档用户的认证和授权以及Notes文档内容的机密性。这些数据安全机制包括文档读访问列表、数据字段加密以及访问控制列表（ACL）的使用。

图21-1说明了Notes安全机制的层次结构。对于用户或者其他Notes服务器来说，第一入口点是在服务器级上。如果要访问一个服务器，那么访问者必须要有该服务器所接受的一个证书。如果服务器或者用户授权使用数据库ACL，那么它（他）就被授予了数据库级访问权。

表21-2概括了不同的ACL级别所授予的访问权限。只有在通过Lotus Notes来访问数据库的时候，数据库ACL才有用。数据库可以拷贝到Notes环境之外。数据库ACL级别用于管理对层次结构图中的更低级组件的访问。如果数据字段加密了，那么用户必须要有正确的加密密钥。

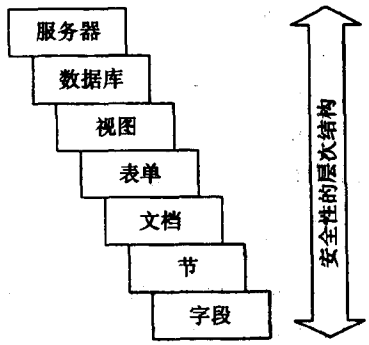


图21-1 Notes安全性的层次结构

表21-2 访问控制列表级别

无访问权	用户不能访问该数据库，服务器不能复制它
委托人	用户可以向数据库提交文档但是不能读取数据库，服务器不能复制数据库
读者	用户只能读文档，服务器只能接收对文档的改变
作者	用户可以读取所有的文档并编辑或删除他们创建的文档，服务器可以接收文档并复制改变或者删除——如果服务器拥有该文档的话
编辑者	用户可以读取和编辑数据库中的所有文档，服务器可以复制新文档和改变并执行删除功能——如果允许的话
设计者	用户可以改变数据库结构并编辑所有文档，服务器可以执行除了复制访问控制改变以外的所有功能
管理员	除了拥有设计者的所有访问权以外，该级别还可以控制访问控制列表、复制优先级以及删除数据库，服务器可以复制所有改变，包括访问控制列表在内

访问控制列表用于判断用户或者服务器是否可以访问数据库，如果可以访问的话，该用户或服务器可以对数据库进行什么操作，用户具有什么数据库角色或者安全级别。对于每个数据库来说，ACL是惟一的。一个复制的数据库可以具有一个同源ACL不同的访问控制列表。ACL中的项可以是用户名、服务器名或者由用户名和（或）服务器名所组成的集合——以组名的形式。

21.2.2 用户认证

从前面可知，访问控制是依赖于用户标识的。为提供认证和机密性，Notes同时使用了公开密钥加密和私有密钥加密。公开密钥加密用于用户认证，而私有密钥加密用于机密性。Lotus Notes是RSA公开密钥加密技术的早期用户之一。然而，由于美国出口的限制，Lotus Notes具有两种版本——分别针对与北美境内和境外。Notes的国际版使用了一个不同的加密算法，该算法使用了更短的密钥。由于这两种版本的存在，每个用户具有两个公开密钥对，一个大约512位的长密钥和一个大约400位的短密钥。

Notes认证过程在用户和服务器之间或者两个服务器之间使用了一个询问-响应协议。当用户试图与Notes服务器建立一个会话时，他们首先向服务器发送他们的标识符（ID），其中包括有他们的公开密钥和一个证书列表。证书的作用是使用用户的公开密钥来验证用户的名字——假设服务器信任颁发证书的认证中心。然后，服务器向用户发送一个询问数字，接着用户使用自己的私有密钥来加密该数字并把它发回给服务器。然后服务器使用用户的公开密钥来解密该数字，如果前后两个数字相匹配，那么用户就通过了认证。然后，用户也使用同样的过程来认证服务器。

Notes文档的机密性可以在文档级上提供，也可以通过对具体字段进行加密来提供。用于文档加密的密钥可以由用户来生成，并添加到他们的ID上。设计者级和管理员级控制都必须要启用文档字段加密。如果其他用户要访问文档，那么他们必须要有合适的加密密钥。这可以通过使用email工具或者使用一个特殊的对话框来实现。另外，数字签名也可以附加到文档中的字段上。

21.2.3 Lotus Notes邮件

Lotus Notes提供了电子邮件工具，不过它们只是被看做包含邮件文档的另一个数据库。所有的Notes工具都可用来管理邮件文档。使用一个基于VIM的前端，Notes可以用做后端消息存储。Notes也提供了同许多其他的邮件系统进行互连的网关。Notes数据库中的用户可以使用同X.500兼容的命名，从而可以同其他基于X.500的系统进行互操作。为了安全邮寄和存储，所有的邮件都可以使用RSA公开密钥机制来进行加密。发送者使用接收者的公开密钥来加密消息，而接收者则使用他的私有密钥来解密消息。在这种方式下，只有接收者才能够解密消息。另外也可以使用电子签名来签名文档。当用户打开文档时，Notes使用电子签名来验证发送者的标识。

21.3 下一步的展望

因特网文档工具和WWW服务正浮之欲出。超文本传输协议（Hypertext Transport Protocol，HTTP）服务器和超文本标记语言（Hypertext Markup Language，HTML）浏览器

提供了类似于Notes观感的文档服务，但是它还没有达到Notes的规模和能力。HTML文档具有文档导航能力，这为处理广泛分布的文档提供了方便和高效率。在许多平台上，访问HTML文档是非常容易的。然而，HTML文档的开发和安全性还不能和Notes同日而语。

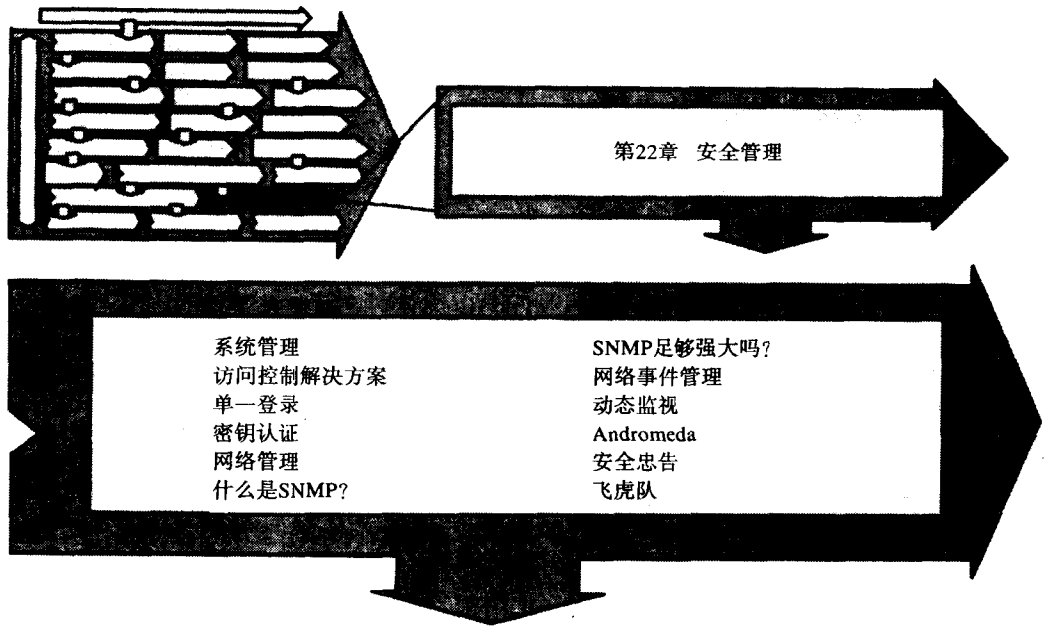
Notes的下一版本将会提供从HTML到Notes的转换功能、HTTP导航功能，以及对Java的支持。这意味着可以通过Web浏览器和通常的Web客户软件来访问Notes文档。这还意味着Notes客户端也可以用做一个Web浏览器。如果Notes文档中包含一个Web文档引用（通过一个URL链接），那么Notes可以获取该网页。

日历和工作流技术也是依赖于一个定义良好的基础结构的分布式应用程序。排时间表和日历工具早已成为群件技术的一部分。业界也提出了几个用于日历功能的标准。厂商独立日历（Vendor Independent Calendaring，VIC）API是Lotus和IBM支持的。Microsoft也在其扩展MAPI中包括了日历和排时间表工具，而XAPIA则把重点放在日历交换上。当把排时间表和日历技术同工作流工具结合在一起的时候，就可以更好地实现办公自动化。究竟这是会提高企业办公效率，还是会让计算机更多地控制我们的生活，还有待时间来说明。

21.4 小结

电子邮件系统是分布式系统的最早实现之一。许多针对与邮件的技术和标准使得其他分布式系统走向成熟。Lotus Notes是一个非常成熟的群件系统，它着眼于文档的分布化——不管用户位于何处，他们都可以访问和使用文档。通过使用Notes，一个高效率的工作组可以分布在一幢大楼内，也分散在世界各地！因特网的发展将继续影响技术和标准的开发和成熟。在前面两章中，我们已经探讨了如何让分布式环境工作起来。在下面几章中，我们来看一下如何真正实施分布式环境。

第22章 安全管理



在安全解决方案的部署中，管理是一个重要的考虑事项。一个不可管理的解决方案是不能真正发挥作用的，并且非常容易受到破坏。在分布式环境中，特权是在系统到系统的基础上进行扩展的，因而访问特权的及时删除是一个很常见的问题。如果没有集中式的管理控制，那么员工在离开公司以后仍然可以保留其访问权限。除非本地管理员撤销它们，否则它们将一直存在。

另一个问题是，本地系统管理职责中有太多不一致的地方。系统管理员具有不同的教育程度、工作经验以及工作量。有些可能是全职管理员，有些可能是兼职工作的，并只在需要的时候才工作。尽管每个系统都有自己的系统管理员来维护，但是许多同系统管理员相关的因素导致了不同的系统具有完全不同的安全性级别。

因为入侵者可以寻找到系统中的漏洞并进而用做其入侵的基础，所以提高整体计算群体的一般安全等级是非常重要的。如果不使用一个集中式的管理工具来实施一套最低安全标准，那么如何解决系统管理不一致的问题？依靠本地管理员的专家经验来捕捉入侵者吗？或者，这是一种能够自己成为集中式方法的行为吗？

尽管分布式系统和安全管理的成本通常是隐藏的，但对于大多数组织来说，这是一个非常值得注意的问题。组织应该把安全问题交给一组经过良好培训的人员去解决，而不是把它们委托给本地管理员去做。大多数组织都没有精力和财力对每个系统管理员进行所有计算安全方面的培训和教育。尽管需要向管理员培训一些计算安全知识，但是更实际可行的做法是让少数几个人成为安全专家。

集中式安全功能也已经出现在用户群体中了。谁也不愿意非得记住多个用户ID和密码。集中式解决方案可以利用单一登录技术，这种技术的认证机制可以被很多系统和应用程序所使用。通过使用集中式系统管理解决方案，本地管理工作的总体需求也降低了。这种做法具有另外一种优点，即它可以减少需要具有本地系统高级特权（例如UNIX根访问）的用户的数量。

审计功能的集中化也是一个发展所趋，它可以跟踪许多系统上的用户动作，而不是依赖于不同系统上的独立审计跟踪。集中式的报警机制可以提供快速探查非法行为的能力，这要比依靠单独的系统管理员来检查入侵者的做法快得多。

> 安全功能的公共机制和集中管理是一件好事吗？一种观点认为分散式的、非集中式的安全控制是理想的。其根据是单一控制不管有多强大，总是可以突破的。但是要突破多级控制则困难得多。这种观点认为，如果不使用多样化的安全控制，那么攻击者只要突破一点就可以破坏整个系统。因而，安全的集中是一种糟糕的做法，就是因为它会成为单一突破口。这种观点是值得注意的，它并不是一无是处。补偿式控制当然要比单一控制更强大。作为对集中式观点和分散式观点之争的结果，我们可以把本地控制和集中控制结合起来使用，从而更好地保证系统安全。

有很多原因促使我们为安全问题找到一个公共的、集中式的解决方案。首先，一个集中的解决方案可以为整个计算群体实施标准，因而能够提高整体安全级别。并且，通过把功能集中到少数经过良好培训的、可信任的人员，集中式解决方案还可以降低安全和系统管理的总体成本。对于个体用户来说，集中式解决方案可以为其提供单一登录能力，因而消除了对多个用户ID和密码的需要。集中式安全管理也限制了需要高级访问特权的用户数量。集中式安全管理具有调查用户的责任，当非法行为可以定向到个体用户上时，对这种行为的警报和检查可以变得更有效率。最后，集中式审计可以对跨越多个平台的用户行为进行统一的跟踪。

在本章中，我们将探讨一下分布式环境中的集中管理安全解决方案。这些解决方案大致分为如下几组：

- 系统管理。
- 访问控制。
- 单一登录。
- 密钥管理。
- 网络管理。
- 事件管理。

同前面的章节一样，我们的目的并不是评论某个产品的功能和优点，而是对如何实现这种解决方案给出一个总体指示。我们将根据每种解决方案的类型来看一下所涉及到的问题，并给出通用的解决思路。

22.1 系统管理

集中式系统管理产品所执行的任务面很宽，其中包括用户账号的维护、数据备份以及打印假脱机服务。这类产品有Computer Associates公司的CA UNICENTER、惠普公司的OpenView系统管理系列产品以及Tivoli Systems公司提供的管理工具。这些产品提供了集中管

理大量系统的能力。一般情况下，它们也提供了对从个人工作站到大型机环境在内的不同平台的支持。这类产品一般也支持包括UNIX和Novell Netware在内的不同操作环境。通过把管理任务集中起来，这些产品至少是降低了对本地专家和支持的需要（如果没有消除的话）。

> 今天，很多系统都是由位于很远距离的管理员管理和支持的。我们的本地UNIX服务器是由一个2500英里外的小组支持的。惟一的本地管理工作就是每天傍晚更换备份磁带。在我们本地办公室里，没人知道系统的超级用户密码。

很多产品都具有一个在所有平台上都容易使用的界面，这使得管理人员在执行公共安全任务时无需具有太多的专家经验。因此，很多普通的安全管理任务，例如用户维护等，都可以委托给一个中心授权体去完成。

系统管理产品提供了很多安全功能，其中包括：

- 认证的集中管理，其中包括用户ID的维护和密码策略的实施。
- 安全控制机制的应用，例如日历访问限制。
- 用户访问权限到资源的映射。
- 对控制应用中的脆弱点或者漏洞进行监视。
- 集中化的审计跟踪，包括对失败访问尝试的监视。

对用户ID以及相关的用户ID号进行集中式控制可以解决很多问题。首先，资源访问权限是在用户ID的基础上授予的。如果两个用户被给与同样的用户ID，那么其中每个都可以访问另一个的资源。集中式管理解决方案支持用户ID和用户ID号的一致指定。

普通用户对安全的最常见的抱怨是他们必须要记住多个用户ID和密码。大多数系统管理解决方案都提供了单一登录能力。例如，Tivoli为Kerberos认证模型提供了管理支持。这允许用户只需要被解决方案提供的授权中心认证一次。不幸的是，单一登录工具一般很少支持每个平台、操作系统或者应用程序。

系统管理解决方案解决的另一个问题是用户访问权限的控制问题。在分布式计算环境中，单一用户可能可以访问许多系统和资源。如果不使用一个公共的用户ID，那么这些资源和访问特权在不同系统上的标识可能会非常麻烦。如果使用集中式系统管理，那么在员工离开组织以后，管理员可以很容易标识和撤消该员工曾经访问的资源。

因为认证是集中管理的，所以组织不仅可以规定而且还可以实施安全策略。例如，解决方案要求用户每隔30天改变其密码。如果没有一个中心系统，那么策略就会留给独立系统管理员去实施。系统管理解决方案所涉及到（至少是部分涉及到）的另一个问题是审计跟踪的集中化。因为认证是集中管理的，并且可以实施公共用户ID，所以集中跟踪用户是有可能的。中心审计跟踪通常不是全面的，它的大部分数据都依靠本地审计跟踪为其提供。这意味着大部分的用户行为审计都是本地审计跟踪而不是中心审计跟踪。

集中式系统管理解决方案对于每种情况来说都是正确的吗？绝对不是！首先，有些解决方案允许集中管理安全功能，但实施功能却是本地式的。在这种情况下，我们需要使用本地访问控制来控制安全。集中管理下的系统仍然具有前面讨论过的所有安全问题和限制。明文密码在LAN上的暴露问题，在集中式解决方案中仍然存在，而且危险更大。入侵者只要在LAN上发现一个密码就可以访问许多系统和应用程序。

更重要的是，安全管理只是这类工具所涉及到的一个领域。系统管理解决方案中所包括

的其他功能，如集中式数据备份和打印假脱机，可能是不需要的。既然厂商不愿意把安全组件从全部管理解决方案产品中单独出来，那么你只好多花冤枉钱去买一些用不着的功能。最后，对于一个组织来说，并不是它所使用的所有计算平台都受到某个系统管理解决方案的支持。

对于那些希望把安全问题作为整体系统管理问题中的一部分来解决的组织来说，系统管理工具可以为其提供一个解决方案。在采用一个系统管理解决方案之前，组织最好把自己的安全需求同工具的能力进行详细的分析比较。

22.1.1 访问控制解决方案

访问控制解决方案为操作系统提供了附加控制。这些类型的解决方案存在于多种计算环境上，这包括DOS、Windows、OS/2和UNIX。访问控制解决方案允许密码策略的实施，提供了对系统资源访问的控制，并且提供了用户行为的审计跟踪。认证过程的保护也是它的一个标准功能。许多解决方案可以防止密码在网络上被探查到，还有许多产品提供了对替代密码技术的支持。例如，有些访问控制厂商提供了一次密码集成技术和数字密码生成器。使用这些技术，密码不能捕获并重发。

这些解决方案提供的控制能够基于多种因素来指定访问权，这些因素包括主机到主机、日历以及访问方法（登录、telnet、FTP以及其他方法等）。这些产品还支持用户轮廓，其中包括用于系统管理员的最少特权策略的实现。用户动作的集中式审计和日志机制是一个很常见的功能，它可以报告控制中的脆弱点。

对于所有的集中式安全机制来说，一个重要的考虑是对其内部通信的保护问题。许多集中式解决方案使用不可信的机制进行通信，如使用不加密的远程过程调用通信。

22.1.2 单一登录

单一登录指的是向每个计算机和应用程序验证用户只需要一个用户ID和密码。通过使用单一登录技术，用户就不必为LAN服务记住一个用户ID和密码，为UNIX记住第二个密码，再为大型机访问记住第三个密码。当系统中添加了密码时效需求，但没有在用户需要访问的多台服务器之间进行同步时，许多用户对此十分恐惧。如果用户必须要记住很多会改变的密码，那么很自然，他们会记下这些需要的密码。他们经常会把这些密码记在便签条上，并把它粘在很明显的地方，因此其他人就可以很容易地读取这些密码。

> 我们曾经研究过，通过检查工作区域来发现用户的密码到底有多容易。我们发现，终端、键盘以及桌子抽屉上贴有很多便签条。在50%以上的工作区域里都可以发现密码。

普通用户的恐惧跟计算机支持人员的比起来可以说是小巫见大巫，后者一般要访问大量的系统。单一登录解决方案通过把认证功能集中起来，解决了多用户ID和密码的问题。使用单一登录解决方案需要注意的是，它们必须要安全地进行。如果入侵者能够发现一个用户的密码，或者破坏了安全性，那么他们就可以访问该用户可以访问的每个系统上的每个账号。

当前解决单一登录问题主要有三种方法。第一个是使用前面已经讨论过的系统管理工具和访问控制解决方案。一般情况下，这些工具可以把用户管理扩展到它们管理的所有平台。这也为系统管理工具提供了一个扩展单一登录能力的基础。不幸的是，很多这样的解决方案

具有非常脆弱的认证功能。它们一般会以明文形式在LAN上发送密码。但是如果要把这些产品用做单一登录解决方案，那么一个最大的限制是，它们不能照顾到一个组织所需的全部计算环境。

解决单一登录问题的第二个途径是使用过程编制方法。过程编制（有时也称做屏幕碎片化）能够捕捉用户的击键动作，并在以后进行重放。使用这种方法，用户ID和密码就会被用户可以访问的所有不同系统和应用程序捕捉。过程编制应用程序保存了这些命令，并起到了一个智能前端的作用。当提交一个命令或者点击一个图标时，被捕捉的命令和击键就会重放。这种方法的优点是它可以被很多操作系统或应用程序采用。它的一个主要缺点是过程编制应用程序必须要安全地保存捕捉的命令。如果这些命令能被入侵者发现，那么它们就会泄漏用户ID和密码。

当把过程编制方法用在客户机-服务器应用程序中时，这种方法就会暴露出它的第二个缺点。在这种应用程序中，可能不会给与用户主机系统上的一个真正会话。他们使用远程过程调用之类的方法同服务器进行通信。在这种情况下，登录过程需要为用户提供证明他们已经进行了登录的证据。证书或者访问令牌经常用于此目的。OSF/DCE使用的Kerberos模型就是这种方法的一个好例子。然而，对于大多数组织来说，对每个应用程序和计算系统都使用Kerberos或者类似的技术是非常不现实的，原因就在于它们太复杂。

最近几个月来，加拿大和美国的三个独立的、不相关的客户在分析单一登录技术。他们得出结论，没有一家提供商能够满足他们对单一登录的需求。好消息是，现在的提供商仍然要继续改进其产品，并为之提供更好的服务。并且，还有许多新的厂商要进入市场。成功的解决方案将会是那些解决了安全问题、容易使用、并且支持多种平台和应用程序的产品。

22.1.3 密钥认证

集中式安全管理解决方案需要考虑的另一个方面是加密密钥的控制问题。我们在第15章中已经讨论过，加密密钥的保护和分发最好由一个集中式的函数来进行。对于大多数组织来说，加密密钥的分发和维护应该交给一个中心安全组去做，并且该安全组不能直接访问计算系统或应用程序。这种方法不但把安全功能同应用程序支持人员分离开，而且组织可以使用一个非常安全的中心站点。

22.2 网络管理

网络管理工具为网络管理员提供了管理复杂的企业级网络的能力。HP的OpenView网络节点管理器就是一个很好的例子。这种软件能够为多种网络组件产生一个单一的、简化的视图，网络组件包括路由器、网桥、集线器、打印机和计算机。这种软件可以：

- 提供网络及其设备的一个中心视图。
- 允许对网络设备（例如集线器、网桥和路由器）进行集中管理。
- 监视网络流量。
- 自动探查添加到网络上的新设备。

这些工具所提供的流量过滤功能也可以应用到安全控制上。增强网络安全性的一个最常用的方法是把网络分成多个可信子网。用来实现可信子网的方法包括分离网络、限制对网络的访问以及根据原地址、目标地址或者流量类型（例如TCP/IP端口号）来过滤流量。

1. 分离网络

路由器和网桥可用来把本地子网从公司内部网络中分离出来。这样可以创建可信子网，即只可被本地工作组中的成员和可信外部用户访问的子网。除了被特殊指定去往本地子网的那些流量，其他流量都会被一个网络设备（例如一个路由器）禁止进入本地子网。与此类似，内部流量也被限制在本地子网的可信边缘内部。在这种情况下，本地流量不会被本地子网外部的用户监视到。

2. 过滤流量

很明显，有些流量必须可以在可信子网和内部网络之间传输。否则，把子网连接到互联网络上还有什么意义呢？路由器上的过滤表可以用来明确定义进入连接。例如，假设只有来自外部IP段15.37.xxx.xxx的流量才是有效的通信流量，那么我们可以构造过滤表来拒绝该IP段之外的所有连接。

网桥和路由器也可以限制从子网内部去往互联网络的流量。例如，一个应用程序开发组可能希望把他对互联网络的访问限制到一个软件发布站上。只有来自这个站点的流量才允许进入互联网。网桥或路由器也可用来限制去往指定地址的流量。不仅仅可以根据流量的源地址和目标地址来限制流量，而且还可以根据流量的类型来过滤流量。例如，进入开发子网的流量可以限制为FTP服务。

3. 访问网络

根据网络的类型，把本地设备连接到LAN主干上需要使用称做集线器的LAN设备。访问本地网络以及互联网络，可能会受到集线器的限制。大多数集线器都有能力禁用未使用的端口。这可以防止有人把一台PC机插到未使用的端口上并监视LAN流量。另一个可选择的办法是，当一个未使用的端口启用时，只发出一个警报但并不限制该端口的使用。

当某个端口上使用了一个新LAN卡的时候，集线器可以检查网络地址的变化，并发出警报。当入侵者把现有机器的网线从LAN上拔下来并使用他自己的设备的时候，一个警报就会转发给集中式网络管理系统。

网络管理工具提供了管理集中网络分离的能力。可以不断检查网络设备的配置以保证控制已经实现并且仍旧在发挥作用。这类工具还提供了管理集线器和网桥的功能。

22.2.1 什么是SNMP

简单网络管理协议（Simple Network Management Protocol，SNMP）是用于管理网络上的网络设备和计算机的事实工业标准。称做代理的本地程序用来同网络管理软件进行通信。这种通信可以建立在预定义的标准上，或者可以定制它们从而满足一个特定的需求。可以定制的代理称做“可扩展性”。

管理信息基础（Management Information Base，MIB）指的是由各种类型的网络对象上的信息所组成的一个数据库。MIB可用做多种目的。例如，一个MIB项可以提供有关指定厂商路由器上使用的协议的信息。MIB对于管理许多网络设备来说是非常理想的，其原因在于实现它们只需要相对很少的开销。

SNMP可用来管理连网设备，这种使用可以扩展为提供安全解决方案。扩展SNMP代理能够检查出控制或者破坏中的脆弱性，并且可以把警报转发到网络管理系统。例如，可以通过询问MIB来监视路由过滤功能的持续作用。

22.2.2 SNMP足够强大吗

当前版本的SNMP没有涉及到认证问题。任何能够公式化SNMP请求的客户机都会被给于一个响应。SNMPv1没有提供任何有效的办法来防止第三方查看管理器和代理之间的流量。更糟的是，它也没有提供任何有效的办法来防止攻击者对SNMP管理器进行未经授权的访问。为了克服这些缺陷，已经提议了一个新的版本——SNMPv2。

SNMPv2请求中添加了两个安全功能——一个用来标识消息源发者的认证器和一个用来保护SNMPv2请求免遭泄漏的保密协议。通过使用消息摘要（请见第15章），数据完整性得以保护。请求的接收者使用消息摘要技术来验证请求发送者。有关SNMPv2的详细信息可以从因特网RFC1445、1446和1447中获得。不幸的是，在本书写作的时候，SNMPv2的最初发布版还没有涉及到安全性。

22.2.3 网络事件管理

网络事件管理系统，如HP的IT OpenView Operation，可以从整个公司互联网络中收集信息。这些由消息和警报组成的信息可以在一个中心事件管理站中准备、合并以及提供。根据警报或者消息的类型，事件管理系统可以手动启动或者自动响应。警报及其后动作的日志通常是为审计目的而维护的。

尽管事件管理解决方案主要用于网络和系统管理，但是它们也可以扩展包括安全性。各种类型的警报可以转发到事件管理系统。例如，我们可以开发一个扩展代理来查找没有密码的账号。如果发现一个，那么可以向事件管理系统转发一条消息。可被监视的消息和事件的类型几乎是无限的，但是恒定的监视和消息发送行为会为网络和系统性能带来影响。

消息可以以多种方式来生成：

- 本地系统上的代理可以监视本地系统，并直接同事件管理系统进行通信。
- 消息可以写到一个日志文件中，然后该文件可以被事件管理系统访问。
- 连网设备所生成的SNMP警报可以被捕获和处理。

前面已经提到过，基于SNMP的代理可以根据多种情况来提供警报，其中包括文件和目录许可以及已有的本地安全程序。如果本地代理探查到一个问题，那么一个警报可能会转发到中心事件管理系统上。可监视的同安全相关的行为包括了重复登录系统的尝试。其他同安全相关的事件还包括审计程序的停止或者对登录程序的修改。消息的分离和分组可以基于多种因素，其中包括：

- 类型（例如安全性）。
- 严重性（例如危险的）。
- 源节点。

可以给予操作员不同的能力以处理不同的消息组，这也包括对他们响应消息的限制。另外，还可以禁止他们查看选定组的消息。网络事件管理系统的安全功能应该同正常的操作分离开，这是通常的推荐做法。只有安全人员及其指定的用户才可以访问同安全相关的警报。

根据所接收到的消息的类型和严重性，管理系统可能会进行很多动作。对消息的响应可以包括：

- 要求安全操作员确认消息。

• 在网络上切除设备来禁止访问——通过自动禁用集线器上的相关端口。

• 向安全管理人员报告。

如果本地系统使用一个代理同事件管理系统进行通信，那么就可以开发自动响应功能。这可以包括：

• 禁用一个登录失败次数太多的账号。

• 删除未经授权的文件或者目录。

• 启动一个特殊的审计过程来跟踪某个用户的所有击键动作。

所有动作（包括操作员动作和自动响应）的日志保存在一个审计跟踪中。消息本身通常存储在一个RDBMS中，我们可以对其进行定制从而使其包括附加的安全控制和报告能力。

后面，我们会介绍对网络事件管理能力的一种特殊使用，为黑客构建一个探查网络攻击行为的工具。我们把此工具称做Andromeda。

22.2.4 动态监视

动态监视产品可用来向安全人员提供即时警报。这些产品可以实时监视目标系统以寻找系统破坏行为。很明显，并不是对所有的资源都可以使用这种方法来监视。然而，很多同安全相关的重要事件都是逻辑的候选监视目标。如果探查到一个系统破坏行为，那么警报可能会转发到一个事件管理系统。

系统的动态监视是通过三个关键工具来完成的，它们是一个本地代理、网络管理软件以及集中式报警管理软件。本地代理是作为事件管理解决方案的一部分提供的，它运行在被监视的系统上，并且可以开发来探查很多同安全相关的事件。这些事件包括了对系统安全程序的改变。最近24小时之内的三次失败登录尝试也意味着一个安全事件。对密码认证和授权机制的访问控制（包括登录程序和密码文件）的改变也构成一个安全事件。还有，如果探查到对授权控制（如系统文件和目录的许可）的未经授权的改变，可能也说明出现了问题。更深入的安全检查还涉及到对密码操作系统程序特性的检查。

数字签名可用来保证操作系统程序的有效性。数字签名根据一个程序或者数据文件的内部特征而产生数字值。即使是程序中的微小改变，也会导致数字签名的明显变化。我们可以为密码程序（例如登录程序）产生数字签名，并把结果值同一个已知的数字签名值进行比较。如果二者不匹配，那么这就表明有可能程序已篡改了。图22-1给出了使用远程检查技术的一个概图。

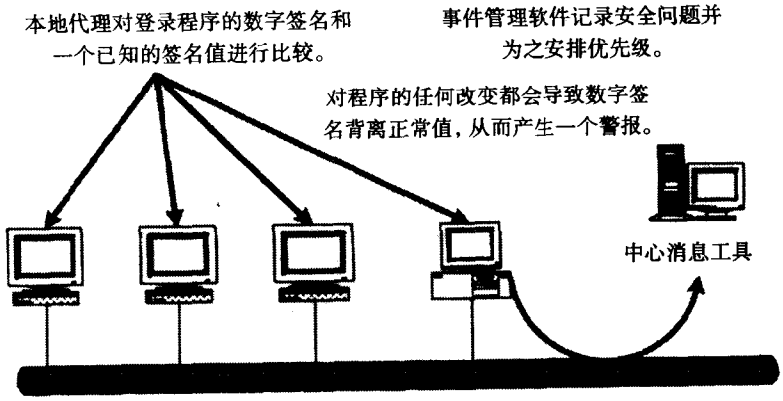


图22-1 安全问题的远程检查

很多商业解决方案为本地代理提供了动态监视大量安全控制或事件的能力。这些解决方案能够监视很多目标系统，并向事件管理系统转发审计信息。它们也可以不断检查文件和目录许可，并在发现未经授权访问的时候实时（近乎）产生警报。本地控制机制的存在和正确执行，例如C2遮蔽密码文件，也可以被监视。甚至运行中的程序也可以被监视。例如，本地代理本身可以被监视。缺少或者伪造代理都可以检查出来。

科学应用国际协会（Science Application International Corporation，SAIC）的计算机误用检查系统（Computer Misuse Detection System，CMDS）提供了安全事故的实时检查。把本地审计数据和用户轮廓结合起来，CMDS能够检查可疑行为，并提供集中式的警报。

22.2.5 Andromeda

如果一个入侵者获得了你内部网络的访问权，那么他就有机会使用工具来扫描网络中的脆弱点。这些工具使用通配符技术（例如，15.37.*.*）来联系每个可用的网络地址。从第一个可能的地址开始，并逐渐递增至每个可能的地址，攻击者可以寻找网络服务中的脆弱点。可以检测的网络服务包括FTP、TFTP、伯克利以及网络文件系统服务。

如果系统在一个安全组的授权下部署在网络上，而在该安全组中任何访问认为是未经授权的，那么这会出现什么情况呢？部署这些系统的目的在于检查入侵者的网络扫描行为，并向安全人员转发警报。这些系统应该在如下前提上构建，任何同他们进行通信的尝试都是破坏公司互联网络安全性的尝试。

老资历的计算机人员——他们通常运行一个提供源代码的UNIX版本，是完成此任务的理想人选。通过编制UNIX端口映射器和网络守护程序，他们可以很容易地开发出捕获所有网络访问尝试的软件。通过修改源代码，对这些服务的任何网络访问都会记录下来。有关访问请求的信息可以转发到一个事件管理系统，例如HP IT/Operations。然后，HP IT/Operations可以把一个关于可能的网络攻击的警报转发给安全人员。图22-2说明了检查网络扫描行为的途径。

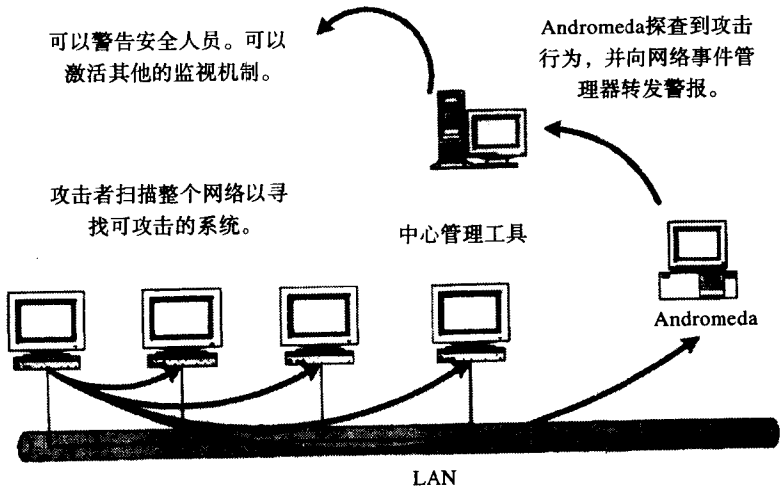


图22-2 Andromeda

这种思想由来已久，这并不是我们的功劳。作为对网络探查行为的响应，如Courtney和Gabriel之类的捐赠软件工具已经开发出来了。我们的解决方案（我们称之为Andromeda）使

用了同样的思想，但它是作为一个独立的解决方案实现的，它并没有使用本地操作系统日志文件。

> 据说，埃塞俄比亚国王克普斯的妻子卡西帕娥娅因为骄傲自大而惹怒了宁芙海神。一气之下，宁芙海神释放了一个海怪去破坏埃塞俄比亚的海岸。在一个预言者的建议下，库普斯国王献出了他的女儿Andromeda。Andromeda被锁在一块岩石上，但是帕尔休斯杀死了海怪救出了Andromeda。

构建Andromeda需要考虑如下建议：

- 使用老的、过时的UNIX系统。不需要使用新硬件，因为监听网络流量并在发现问题的时候转发警报只是机器的责任。
- 系统上只应该有一个root用户。如果可能的话，root访问应该限定在系统控制台上（使用/etc/security）。
- 一定要保证机器的物理安全。
- 当前所有的网络守护程序都应该替换成一个代理版本，它将捕捉流量并创建警报。
- 代理网络守护程序应该尽可能地捕获有关探查的信息，并把警报信息转发到事件管理软件。警报应该也自动通知给安全组。
- 定期改变Andromeda机器在网络上的位置。

把警报发送给网络事件管理器主要有两种方法。可以生成一个基于SNMP的警报并转发到事件管理器。但是最简单且最安全的做法是格式化警报并把它们放在本地文件中，然后由事件管理器代理来访问这些文件。

关于客户机是否要泄露有关Andromeda机器的信息给内部雇员，我们已经讨论过了。我们认为，它们的存在并不是一件坏事。如果因为它们的存在而使得员工不再不经授权测试网络（使用一个诸如SATAN的网络），那么就更好了。

22.2.6 安全忠告

如果厂商发现了其产品中的安全问题，那么他们会例行公事般地发布一些安全忠告。他们的意图是让他们的客户尽快解决问题，从而免遭自身攻击。对于组织来说，与其等待操作系统或者应用程序的升级，还不如及时获得并应用这些厂商忠告。因为脆弱性往往是非法行为所发现的结果，并且可能已经人所共知了，所以组织必须要尽快的应用厂商忠告。许多厂商也提供了数字签名以保证其忠告的完整性和正确性。使用数字签名，用户可以很容易验证每条忠告，并准确地发现伪造品。安全忠告的收集和分布是每个组织必须要注意的中心问题。

22.2.7 飞虎队

如果发生了安全事故，那么很多组织都会发现自己根本没有做好有效处理事故的准备。根据我们的经验，安全事故的发生是很快的。当事故来临时，情况往往一团糟，即使最有经验的系统管理员也往往会束手无策。根据我们的认识，组织最好预先建立一个小组来响应对计算环境的攻击。该小组应该由一些能够处理紧急安全问题的精英人员组成。这种小组一般被称做“飞虎队”，它不但包括各种领域（数据库、连网以及操作系统）的专家，而且还应该包括安全人员和管理人员。飞虎队应该同其他的飞虎队保持联系，这包括厂商、法律实施人

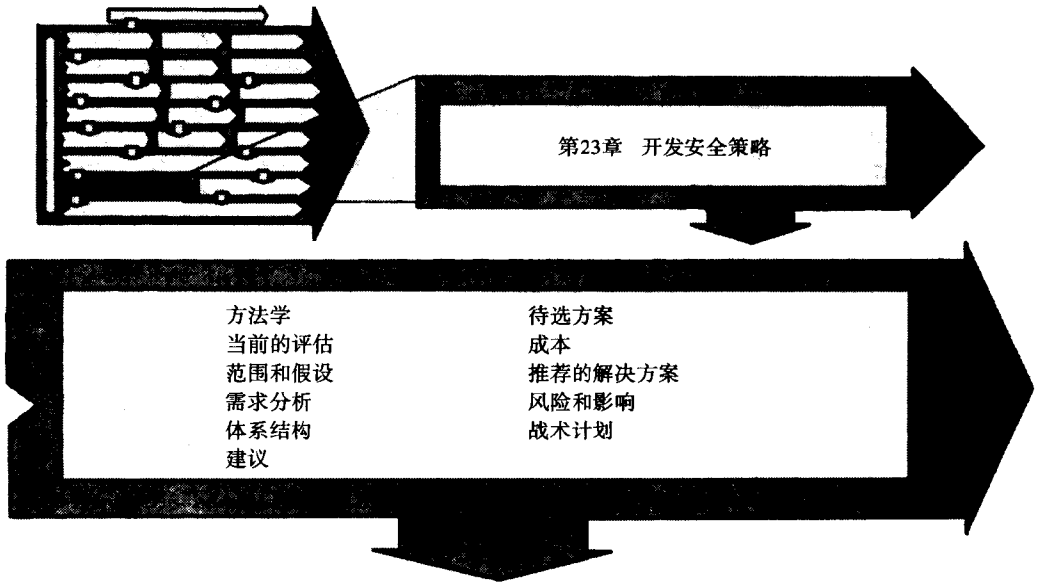
员、通信提供商以及因特网CERT组织（Computer Emergency Response Team，计算机紧急响应组）。根据我们的认识，为了分析情况并及时合理地做出响应，飞虎队必须能够访问计算机化的工具和大范围的系统与网络信息。准备、培训、装备（工具）以及获得信息出处的意识，都是很重要的。如果事到临头才意识到这一点，那么你会发现已经晚了。

22.3 结论

集中管理分布式系统的安全性能够提高分布式系统的有效性。集中式安全管理解决方案能够在整个计算群体范围内实施标准。另外，这种解决方案还具有降低总体管理工作量和集中专家知识的能力。但是集中式安全解决方案并不是解决安全问题的“万能药”。他们可以看做对本地系统管理员提供支持。如果在每个系统上都实现本地安全解决方案，并用其来探查使用广播技术的攻击者，那么组织只需要付出很少的成本。集中式方案允许使用更复杂的方法，它要求组织把专家知识集中到少数人身上。

若要解决计算安全的问题，组织需要使用成功的方法来建立一个公司策略。在下一章中，我们将讨论一个已经被多个组织成功使用过的方法。

第23章　开发安全策略



企业所面临的计算安全问题是非常复杂的。在前面几章中我们已经知道，计算安全问题涉及到了许多不同的方面。例如，解决安全问题需要实现强大的、健壮的安全控制、需要监视控制以及培训用户群。只有技术上的解决方案是不能解决问题的。事实上，各种不同的技术方案会为组织带来不协调的巨大开销，这会加深解决问题的难度。

计算机安全和家庭安全有很多的相似之处。保证家庭安全的第一道措施是锁门。虽然这不能绝对保证家庭安全（坏人可以破门而入），但是它肯定会给小偷带来麻烦。同家庭安全一样，为计算资产锁上大门也是非常重要的。

不管是保证家庭安全还是计算安全，我们的方法一定要有一个平衡。如果房子没有后门就那么敞着，那么即使给前门装上最好的锁也毫无意义。好钢应该用在刀刃上，所以我们应该把重点放在最有可能的安全暴露点上。小偷通常不会带着梯子去行窃。因此，最先花钱的地方应该是为低位置的窗户添加铁丝网。

许多组织已经为一些独立的、不协调的安全技术花了很多钱。然而这些投资并没有解决企业的整体安全需求。更好的做法应该是实现更强大的综合解决方案，这能降低管理和培训成本。把许多独立的、不协调的解决方案结合起来只能一团糟，不但不全面而且成本也不小，并且用户也不会接受。尽管这些解决方案可以解决独立的安全问题，但是它们通常并不满足组织的总体目标。如果员工不能明确自己的责任，那么再好的技术也毫无用处。例如，当你不在家的时候，如果你的小孩没有锁门就出去玩了，那么即使你在门上已经装上了世界上最好的锁，那也毫无意义！

关于安全性的最后一点是，安全性必须反映出我们的组织和IT体系结构。如果组织要实

现分布式计算，那么设计基于集中式体系结构的安全方案就不会有意义。安全不应该看做是孤立于环境之外的。家庭的安全同左邻右舍的安全是直接相关的。只顾一个而不顾其他是绝对不能解决问题的。

23.1 安全策略

计算安全是影响整个公司的问题。从这种意义上讲，计算安全是一个业务问题而不仅仅是一个技术或者计算问题。如果要解决一个业务问题，那么组织需要对问题有一个战略上的观点。许多组织已经建立了策略以解决计算安全问题，我们将在后面讨论这些组织所使用的一个方法学。

一个安全策略是一系列的具体步骤，组织可以利用这些步骤来提高现有的安全级别。为了制定安全策略，组织需要使用一个评估过程来确定他们在计算环境整体安全性方面所处的位置，同时还需要定义他们的安全目标并计划实现这些目标所需要的步骤。

使用一个已定义的方法可以保证所有的门和窗户都已经锁上。当建造房屋时，使用安全门窗是一个必不可少的环节。各式各样的组织都已成功地使用这种方法解决了计算安全问题。安全策略可以保证组织的全部目标都能得到满足。它是一个把公司从今天推向未来的计划。策略是在组织明确了当前环境后开发的，它为安全问题的解决带来了一个公共的解决方案。应该认识到，理想环境不是一蹴而就的，相反，只有经过多年的发展环境才可以接近理想状态。图23-1说明了一个示例安全策略，它是在一个延长的时间周期上实现的。

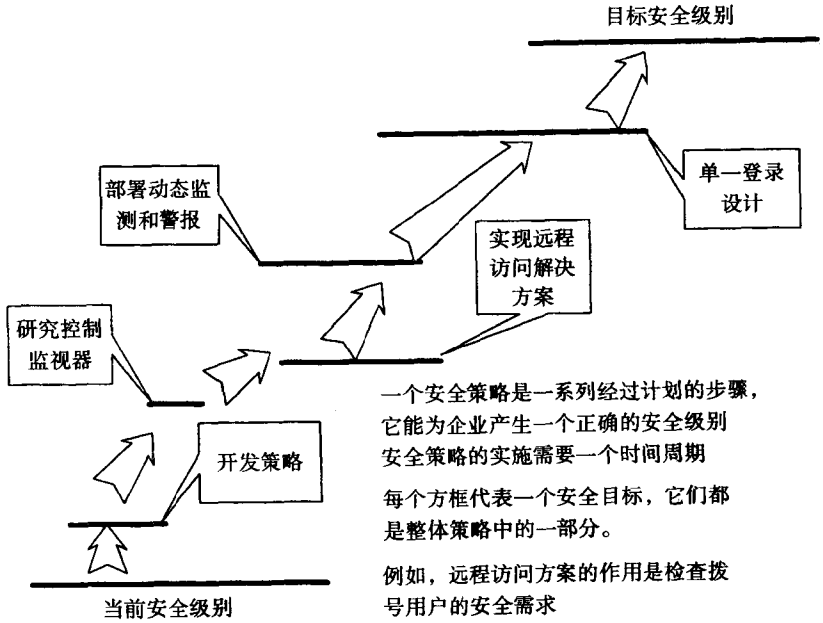


图23-1 一个安全策略示例

对于大多数组织来说，计算安全问题没有解决好的原因并不是因为安全技术的缺乏。事实上，大多数组织之所以为这个问题而头疼恰恰就是因为有太多的计算安全技术。为了解决一些独立的问题，他们已经为技术进行了投资，但是并没有从整体角度上考虑。问题在于，如何根据组织的具体需求来选取最合适的技术？重点绝对不能放在片面的安全问题

上。技术解决方案是整体策略中的一部分，但是策略决不仅限于技术，它必须还要包括组织和业务问题。

许多组织都已经采用了部门业务模型。这种模型把组织分割成多个半自治的业务实体。各个业务实体各不相同，它们都具有自己的文化和业务目标，这就提高了开发整体安全策略的复杂程度。策略应该考虑到公司文化和业务实体的区别，并给出可被所有实体支持的一个公共指导和一套动作。安全策略也必须要考虑到组织的将来发展，并明确组织的新需求和发展方向。

方法

我们已经为很多不同的客户开发了安全策略，其多样化可以同墨西哥政府和一个基于加拿大能源的工具所媲美。这些策略各有其独特的目标和目的，这反映出了组织间的不同。在一种情况下，策略的目标是为组织带来一个公共的安全技术基础；在其他情况下，策略定义了一些步骤来增强计算环境的总体安全性。在最近的提议请求中，我们的一个客户把他对安全策略的需求总结如下：

本项目的目标是为我们的信息技术环境定义一个安全策略。该策略将为安全解决方案的实现提供框架，从而在一个高度分散的、任务重要的、复杂的环境中维护控制并保证控制的安全性。

在安全策略的开发过程中，我们开发了一个帮助我们完成该过程的方法。该安全方法有5个主要的阶段：

- 阶段1——明确当前位置。
- 阶段2——定义希望具有的环境。
- 阶段3——分析待选方案。
- 阶段4——明确所采用的最佳路径。
- 阶段5——开始。

安全策略的第1阶段从评估组织的当前安全级别开始。在该阶段中，我们要分析组织所使用的技术、它的策略和程序以及它的管理命令。另外，我们还要进行风险评估——使用测试当前控制强度的技术和工具。研究的边界，以及研究过程中所做的重要假设，也在该阶段编辑。例如，如果研究是要排除一个选定的领域，例如微波传输的安全性，那么要陈述该限制。假设组织的发展方向是朝着某一套技术前进，那么要陈述该假设。还要明确组织所面对的主要问题，并详述其中的每一个问题。

在第2阶段中，定义了总体目标。编译了一个需求分析并开发一个安全体系结构。在该需求分析和体系结构的基础上，提出一套总体建议。这些建议能够提高组织及时监测未经授权行为的能力。

第3阶段的重点在于评估可用来满足需求的待选方案。检查每个建议，并且研究可用来满足该建议的待选方案。

在第4阶段中，要提出针对特定方案和动作过程的建议，分析预计成本、风险和影响，并开发一个战术计划。在这里，已创建了安全策略，其实现的主要任务可以开始进行。

第5阶段是最后一个阶段。在这一阶段中，我们要继续执行计划，并实现所建议的技术上的和程序上的解决方案。由于没有安全策略，许多组织一不小心就到了策略的最后一个阶段，

从而只好不得不胡乱地部署技术方案。然而，这种做法是很少有作用。

在描述建立安全策略的过程之前，必须首先明确几个有关安全策略的概念。尽管对于个体组织来说，每个策略都是不同的，但是策略之间毕竟存在一些共性。若要掌握安全策略的开发，需要注意如下四点：

- 首先，策略的目标是为组织的将来发展提供一个指导，而不是试图为每个安全问题马上提供一个解决方案。策略可以为当前某个问题建议一个暂时的解决方案，但是重点一定要放在整体目标上。
- 其次，我们很难满足组织中的每个实体所提出的每个独立安全目标。策略必须要在组织的公共目标和个体目标之间取得一个平衡。
- 再次，并不是所有的目标都可以立即实现。所建议的许多解决方案可能在将来才部署。但明智的做法是，现在采取一些小的步骤以推动这些将来的解决方案。
- 最后，策略会改变和发展。业务目标和组织方向的变化会推动安全策略的发展。组织应该定期分析策略，并在需要的时候对其进行修改。

23.2 安全策略路线图

我们开发了一个路线图来说明安全策略开发过程中的各个步骤和阶段。在开发一个安全策略时，你会不断地学习有关组织的新知识、技术以及问题。建立策略所使用的构造技术必须能够解决这些新问题。因此在方法的使用中我们应该尽可能地保持灵活。图23-2说明了在开发安全策略时所使用的路线图。

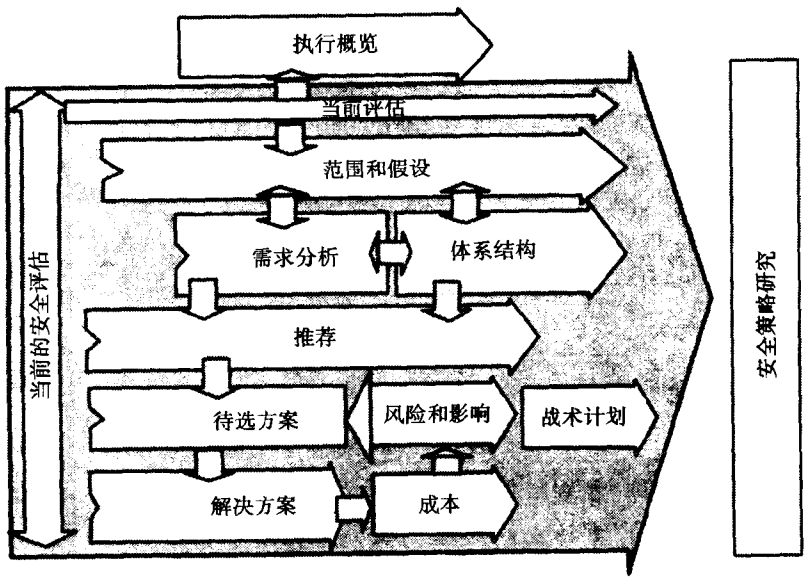


图23-2 安全策略路线图

下面我们来看一下建立一个安全策略所涉及到的各个步骤，首先从当前评估开始。

23.2.1 当前评估

为了掌握如何最好地完成目标，首先明确我们所处的起始位置是非常重要的。当前评估

的目的是为安全策略提供起始点，它包括以下5个主要步骤：

1) 允许当前的策略、程序以及最佳实践。

2) 访问大量的人，包括IT技术专家和管理人员。

3) 分析当前所利用的技术。

4) 进行风险评估以学习有关环境的新知识。

5) 分析将来的应用和发展方向。

当前评估的第一步是理解公司文化。例如，在公司文化中，员工的行为是可信的吗？如果是，那么这种思想就意味着所有的员工都是默认可信的，因而可以访问公司信息，除非有好的理由来解释为什么不能访问。与这种原则所对立的思想则假设所有的员工都默认是不可信的，如果要访问所有信息，那么他们必须要进行特殊授权。一般来说，安全策略不是改变公司文化的机制（那样做会遇到很大的阻力）。安全策略应该理解和补充公司文化。

获得公司文化的一种方法是分析现有的策略和程序。记住：如果现有的策略和程序不完善，那么请修改它们。我们在前面的章节中已经讨论过了计算策略问题。现在我们来回顾一下开发计算策略时所涉及到的主要问题：

- 策略不应该反映特定的技术。它们应该反映出同所有平台所相关的问题。在具体平台上实现一个策略是一个过程。
- 策略应该反映出组织的文化。背离公司文化的计算策略是不可以遵循的。
- 组织的策略应该被员工所理解。如果大多数员工不知道策略的存在，那么策略就不会起到应有的作用。

分析完公司文化以后，下一步就要访问组织内的大量人员。若要理解当前的问题和指导，那么同大量不同级别的员工和管理人员进行公开的讨论绝对是最好的办法。讨论的目的是为了明确人们当前所面对的问题，并获得他们对将来发展的看法。一个好办法是预先发送一个示例问卷，但讨论本身通常是非正式的和无结构的。

> 我们需要让我们所访问的人员知道我们的行为并不是某次审计的一部分。我们的意图并不是想批评任何个人或者组；相反，是为了获得他们对将来发展方向的观点和思想。我们发现，经过几百次会谈以后，大部分人都非常愿意讨论安全问题和解决方案。

我们应该访问大量的个人，其中包括用户群体、IT专家、审计人员和管理人员。应该如何实施安全功能？对于这个问题，应用程序的实际用户能够提供非常好的信息。审计部门能够提供有关控制力度的信息和他们对整体安全环境的感觉。从技术专家身上，你可以学到有关当前问题的大量知识。管理人员通常不懂技术细节，但是他们能对将来的发展发表见解。例如，如果有计划要在一个新项目中实现一个新的密码技术，那么管理人员可能具有该计划的知识。

> 我们已经发现，一次会谈的最大人数是三个。如果房间内的人数多于三个，那么就会有人打断讨论。这很有趣，但是，当人数增加时，每个人的参与就会减少。更重要的是，人们一般愿意在一个较小的组中互相交流思想和观点。作为一种礼貌行为，我们会同会谈者一起检查会谈笔记，并对他们所要求的地方进行修改。如果会谈者同意，那么会谈笔记就会被包括到最后的报告中。如果会谈者对笔记持有异议，那么我们就会修改笔记，或者重新进行讨论。

会谈过程也是掌握和记录组织所用当前技术的一个好地方。你会惊奇地发现，组织中的许多组都对开发策略过程中的这一部分十分感兴趣。许多组织出版IT业务计划，既为公司IT员工也为业务单元。这些计划是了解组织当前工作的一个很好的起始点。

会谈完成以后，我们就会建立一个矩阵来说明所发现的主要问题。该矩阵列出了每次会谈中所标识的问题领域。例如，如果一个会谈者认为组织的策略和程序是不够的，那么我们就在相关的行和列中打上一个标记。这有助于让我们把注意力放在主要的问题上。

图23-3给出了一个问题和需求矩阵示例。

问题和需求	Dave	Karen	Fred	Andy	John	Carol	Kim	Mike
策略和程序	■			■		■		
数字签名		■			■		■	
审计跟踪				■				
网络控制			■			■		■
应用程序控制	■	■						
数据库同步						■		
集中式控制、监视器以及自动警报	■		■	■	■			■
系统认可						■		
数据库访问控制			■				■	
责任分离	■			■				
软件的保护和分发	■		■		■		■	
物理安全						■		
远程拨号访问		■	■					
安全教育和意识				■	■			■
密码和认证标准		■		■			■	
审计				■				■
数字签名	■					■		

图23-3 会谈结果——问题和需求矩阵示例

最后要注意，公平会谈！请平等地倾听所有的问题、考虑和建议。试着学习，不要卖弄！

会谈过程使我们明确了组织内的问题。在下一步中，我们要对计算环境进行一次有限的风险评估，这会使我们发现当前组织所没有意识到的问题。在该过程中，我们要对网络、操作系统以及应用程序控制进行积极的测试。进行这种受限风险评估的目的是测量组织当前的总体安全问题。应该指出，计算风险的总体评估不仅仅包括控制分析。它还包括对技术控制的分析，以及对物理安全、组织程序和攻击可似性等问题的分析。

为了明确控制的整体力度，我们使用了很多工具，其中包括第22章和第24章所讨论的一些工具。审计部门所提供的报告也可以提供有用的信息（如果可用的话）。但最好的做法是进行自己的评估。

在会谈、分析和评估的基础上，我们把这些步骤的结果编译到当前评估部分中。这份总结是对组织当前状态的说明，其中包括了组织所面对的问题及其对将来发展的看法。

23.2.2 范围和假设

对于策略开发人员来说，当建立策略时，列出策略希望实现的目标是一个很好的做法。还有，说明那些不希望涉及到的领域也是同样重要的。例如，如果策略不希望涉及到构建物理安全问题，那么请在范围中表示清楚。如果策略排除了某些计算资源，例如大型机，也请记录在此。如果没有一个实际的决策，那么可以代表客户做一个假设以代替决策。例如，我们假设某个特定的项目对于一个组织来说是非常关键的，但是我们不知道如何实现它。把有关策略的假设纪录下来有助于使读者明白为什么要制定决策和建议。我们可以在完成当前评估后开始记录范围和假设，但是它们通常会在整个过程中修改和改变。

23.2.3 需求分析

需求分析实际上是我们所发现的问题的一个检查列表，这些问题是当前评估的一部分。使用当前和风险评估的结果，我们可以建立一套总体需求，例如：

- 拥有一个公共认证机制，该机制在用户初始访问的时候认证用户，并为多个应用程序和环境所使用。
- 把安全管理从系统管理功能中尽可能地分离出来。
- 提供对远程和移动用户的控制。
- 促进因特网的商业使用。

一般来说，我们可以使用有限数目的语句（一般为20到30个）来表达企业的总体需求。

23.2.4 体系结构

在第3章中，我们提出了一个安全环境体系结构。该体系结构用来为特定环境中的安全实现提供一个模型。总是伴随安全策略的一个问题是：

如果没有公司系统和网络管理体系结构，安全策略可以开发一个体系结构吗？

理想情况下，安全策略是组织在开发了一个网络和系统管理体系结构之后才建立的。但实际情况并不是这样。通常情况下，由于新项目和应用程序的需要，安全策略在开发公司体系结构之前就已是公司所需的了。安全策略的开发和分布式环境的总体体系结构的建立往往是同时进行的，或者说是重合的。

如果没有一个总体体系结构，那么可以在考虑到其将来发展的基础上自己定义一个体系结构，并把其作为策略的一部分。应该把策略范围和假设部分中的全部假设都包括在内。

23.2.5 建议

总体需求定义完成并且体系结构也开发完成以后，我们就可以开发一套建议。这些建议为组织的安全性提供了将来的战略指导。你需要定义那些具有如下需要的领域：需要更深的调研、需要制定一个决策，或者启动一个单独的项目。建议示例如下：

- 为高级管理层对计算环境（包括分布式系统和应用程序）使用的批准开发一个策略。
- 在系统类型的基础上，为所有系统访问公司互联网络开发一个认可过程。
- 实现一个健壮的认证机制，该机制应该为应用程序和系统提供一个单一登录解决方案。
- 使用安全监视器来查阅分布式计算环境中的现有控制。

- 实现可信子网——这些子网应该在最小特权的基础上限制网络访问，从而减少关键子网的安全暴露问题。
- 实现动态监测安全事故的能力。解决方案应该记录下事故，并根据警报的类型以决定是否通知安全人员并采取后续动作。

强烈推荐在每个建议得以继续之前，它们首先应该得到管理部门的批准。为了得到最佳解决方案，我们需要对每个建议都作深入地调查研究。但是如果只分析管理部门可以支持的那些建议，我们就可以节省时间。你肯定不愿意把时间浪费在调查不会被遵循的建议上。

23.2.6 候选方案

同意建议以后，我们就可以寻找可能的解决方案。开发一个安全策略所需要努力的大部分都花在这一步上。这一步的工作就是分析各种候选的解决方案是否满足前面所提出的建议。这些解决方案可能牵扯到也可能没牵扯到技术，我们评价它们所基于的因素包括了解决需求的能力、对已提议的体系结构的适应性、相关成本以及对组织可能产生的影响。这一部分也可用来提供对安全技术的教育、候选方案的强度和脆弱性的讨论以及关于待选技术的细节信息。

23.2.7 成本

除了考虑采纳所提出的建议会带来的风险和影响以外，我们还必须要在建议的利益与它的成本之间取得一个平衡。对每个候选方案所预期的成本（包括购买、维护、培训和支持）都应该详细分析。如果可能的话，应该使用一次收费和年度收费的形式来表示这些成本。为了简化确定成本的任务，我们建议使用预算定价。但是需要强调的是，提供这些成本的目的仅在于指导作用，任何具体解决方案的实际成本通常都是客户方和提供商进行协商的结果。

23.2.8 推荐的解决方案

本阶段的任务是开发行为指导（技术上的和非技术上的）从而实现前面所定义的建议。这些建议的范围可以从具体硬件设备或者软件产品的获取一直到启动对策略开发之类的领域的研究。在某种情况下，这些建议可能只是一些已经启动的过程的延续而已，例如：

- 为所有基于系统和数据分类的系统开发一个认可过程。作为一个最小单元，认可过程应该定义和实施认证标准、访问控制、监视和审计。
- 购买和部署个人计算机访问控制解决方案。
- 实现一个集中式的访问解决方案——使用手持一次密码生成器（针对远程和移动用户）。
- 使用当前的网络管理工具来捕获安全警报。提议的解决方案会记录下所有的安全破坏行为。如果发现严重警报就立即通知安全人员。
- 启动一个引导项目来调查和明确在关键应用中使用OSF/DCE的意义。

23.2.9 风险和影响

每个建议的解决方案和行为指导都会对组织产生影响。它们会改变当前的程序、教育以及人员配置等。并且，它们能够影响到决策的执行情况，从而为组织带来风险。作为策略开

发中的一个环节，我们需要检查和文档化每条建议所具有的潜在影响和风险。

进行风险分析的主要原因是它能帮我们提出一个明确的战术计划。那些不需要组织费多大劲儿的以及低成本低风险的行为都是可供选择的起始分析对象。这些都是能立即产生结果的行为。那些会为组织带来重大影响的行为一般都需要更长的实现时间，并且可能需要另外的计划。

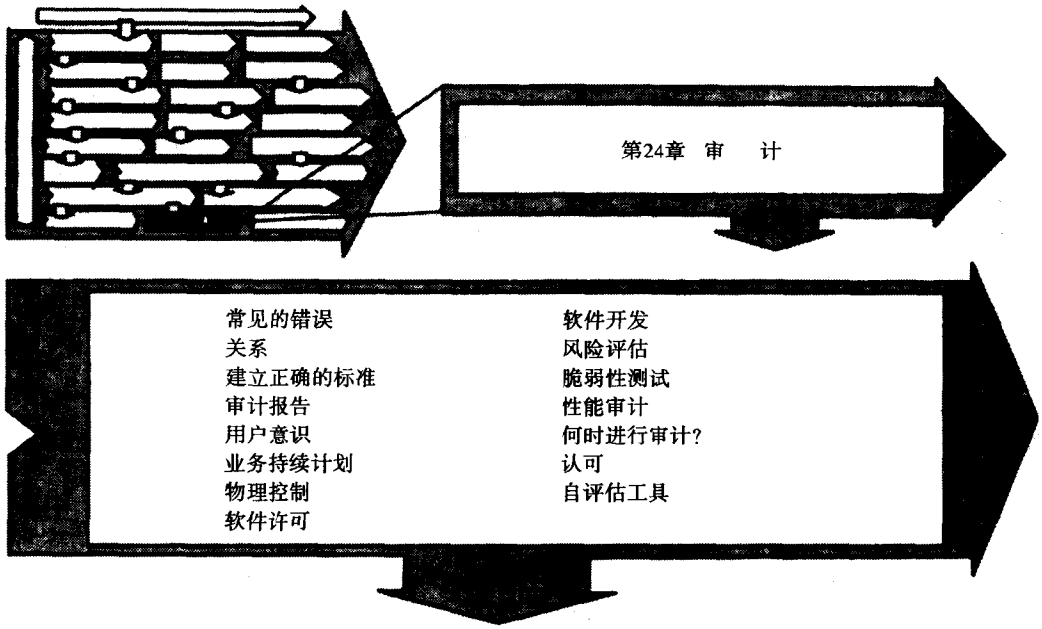
23.2.10 战术计划

组织一般没有足够的时间去立即实现每个建议。推荐做法是，组织应该为安全策略开发一个战术上的实现计划。为了促进该计划的开发，每个推荐的解决方案和行为指导都需要进行优先级排队，其实现方法是根据影响、风险以及实现时间对所建议的解决方案进行分门别类。这一步完成以后，我们就可以为安全策略的实现开发一个战术计划。根据其约束和先决条件，我们可以把解决方案分组到相关的项目中并在计划中予以管理。

23.3 结论

今天，分布式计算环境所面临的最大的信任和安全问题并不是缺乏技术，而是缺乏一个解决问题的策略计划。如果组织能够对将来的发展情况进行分析并计划出正确的步骤，那么计算安全领域中的目标是可以实现的。然而，如果组织没有使计算安全体系结构同计算体系结构保持一致，那么问题的解决最终只能会陷入一片混乱之中！在计算安全问题的解决过程中，组织需要有一个战略上的眼光来看待安全问题并部署集中式的安全机制，这是非常关键的步骤。另外，对计算安全及其相关行为进行独立的分析也是同样重要的一步。在下一章中，我们要对审计在分布式计算中的角色进行探讨分析。

第24章 审　　计



前面我们已经讨论了对策略和指导原则的需要，并且知道了如何为安全性建立正确的基础。那么如何保证这些策略和指导原则，例如对防止用户进行未经授权的访问和非法拷贝软件的需要，能够得到遵循呢？我们知道，密码管理和访问授权之类的控制可以用来实施一致性。但是我们如何才能知道必需的控制已经实现了并且仍旧在发挥作用呢？如果没有锁门的话，那么再好的门锁也起不到任何作用。

对这些问题的回答就是，我们需要不断地检查策略和已建立的安全指导原则的一致性。在大多数组织中，这种对计算环境的维护管理工作是由内部审计部门完成的。

在很多情况下，系统专家与他们在审计部门中的搭档都保持着良好的关系。在许多组织中，系统管理人员和审计人员的合作是很常见的。尽管他们从不把计算机系统的审计工作看成是多么有趣的任务，但是他们总能够在最小的时间内完成应该完成的工作，并且把失败的可能性降至最低。

在另外一些组织中，系统管理人员和审计人员之间却存在着对抗行为。审计过程本身就是具有对抗性的工作，它往往需要很长的完成时间，并且结果也不一定让人满意。我们认为问题的根源在于双方缺乏对计算机审计目的、目标以及方法的共同认识。

在本章中，将对审计在分布式计算环境中的角色进行探讨分析。理解了审计的目的以后，分析一下审计部门和其他部门之间的关系。接着，介绍一些审计工具和方法。最后，要探讨多种不同类型的审计，这些审计都把重点放在计算控制的传统检查范围之外的领域上。

24.1 审计的概念

关于究竟什么是审计，可以说是“仁者见仁，智者见智”。如果你向一组IS专家问术语“审计”在他们眼中具有什么意义，可能你会得到多种不同的回答。有些人认为，审计就是收集关于一个计算机系统行为和资源的数据的过程；另外一些人可能把审计看做是对现有控制的检查或者对欺骗行为的监测；还有一些人则认为审计是Grim Reaper的公司具体化。

审计的定义是“一个过程——在该过程中，为了汇报可以计量的信息同已建立的标准之间的关系，一个独立的个体积累和评估事实”（出自：Alvan A.Arens和James K.Loebbeche的《Auchiting-An Integrated Approach》，Englewood Cliffs，NJ：Pretentice Hall,1994）。审计的本质是对一个给定的目标进行一次独立的检查，并且汇报所发现的实际情况同可接受的标准之间的一致性。在本章中，我们把检查可接受标准一致性的过程称做审计，把负责独立检查计算安全的部门称做审计部门。

例如，财政审计就是对财政状况进行正式的检查。投资方和股东通常不愿意只依靠管理人员对财务状况的报告。他们需要有一个单独的、公正的会计实践和财务状况报告，从而准确地反映出组织的经济状况。

审计的作用并不是重新建立财务报告。准确地说，进行审计是为了检查财务报告是否同已建立的会计实践相一致。如果财政状况同已接受的会计实践相一致，那么它们就看做是公正的和准确的。如果二者之间有偏差——这可能会导致对财政状况的错误判断，那么审计就要汇报这种偏差。如果在审计过程中发现了很多偏差，并且审计人员认为财政状况有误导倾向，那么审计人员必须在审计报告中明确指出这些疑虑。

在计算环境中，审计的主要目的是独立地检查应用程序、系统或者网络同已建立的策略、指导原则和标准的一致性。如果应用程序设计者、系统或者网络管理员没有遵守可接受的标准，那么审计过程要把这种情况报告给管理人员。然后责任方应该采取预防动作并实现正确的安全控制，从而提高企业的整体安全级别。

然而，如果审计过程中没有其他组的参与和交互，也就是说审计是孤立进行的，那么它就不会起到预期的作用。相反，它会导致审计部门同被审计部门之间产生不必要的对抗。稍后，我们将探讨审计部门同其他部门应该保持什么样的关系。首先，我们看一下应该由谁来进行审计。

24.1.1 审计者

如果有人要进行审计，那么他必须先具备如下三个条件：

- 知识。
- 独立。
- 可信赖。

计算机审计包括控制检查、审计跟踪数据分析以及关于改变计算程序的建议。很明显，这些行为需要审计人员具有相关技术的知识。但是客观性也是必需的。组织的计算群体通常不是真正的审计人员——尽管我们所见过的一些最困难的审计工作都是由他们的伙伴完成的。

本质上，审计就是对计算机部门的安全相关行为进行独立的检查。计算机专业人员对他们同伴的行为可能保持沉默——不下判断，不做不得人心的建议。他们也可能把重点完全放

在技术问题上，这样就缩小了审计的总体范围。因为审计涉及到了控制的检查，所以它的行为和结果必须要保持机密。这些因素使得审计人员必须要具有高度的可信赖性。因此，大多数组织都不愿意让他们不完全信任的第三方（特别是黑客）来进行审计。

> 有些人开玩笑说审计人员是在战斗结束以后出来射击伤员的。作为一个专业会计师和审计员，我对此嗤之以鼻。众所周知，我们是只允许佩刀的。

理想的计算机审计人员应该是一个疑心病患者，他应该具有审计经验和强大的计算机背景。如果找不到这样的人才，那么组织就应该建立一个由IS和审计专业人员所组成的计算机审计团队。审计部门在审计分布式计算环境时常常会出现一些错误，现在我们看一下常见的一些错误。

24.1.2 常见的错误

这些年来，我们有机会接触了很多组织中的审计部门。不幸的是，审计部门、安全部门以及用户计算群体之间的交互根本就没有，或者就是互相对抗。在大多数情况下，脆弱的安全环境或者时间和资源的低效使用都是由这种糟糕的关系造成的。

出现问题是因为存在如下关键错误：

- 审计部门没有参与计算安全问题的解决。
- 计算群体误解了审计的作用。
- 用户部门，包括安全部门在内，没有同审计部门保持一个良好的关系。
- 用来进行实际系统审计的标准是不正确的。
- 最后的审计报告没有涉及到被审计的计算环境的关键问题。

24.1.3 计算审计重要的原因何在

大多数企业都有一个正式的审计部门，并且把该部门的权力范围扩展到了计算领域。计算机审计涉及到了对计算机控制的检查，并且还要把检查结果同可接受标准的符合情况做一份详尽的正式报告。审计的目的在于提出一些可以提高计算环境安全性的建议，这是非常重要的，其原因在于：

- 对于大多数组织来说，计算机系统意味着一项很大的投资。对其保管员（例如系统管理员）保护这些投资的能力进行独立地检查是一项很好的业务实践。
- 它能很好地检查企业的整体安全水平。
- 它提高了一致性，并且向组织灌输一个纪律化的安全方法。审计行为的主题可能是这么一句谚语："你可以不做我所期待的工作，但是你必须要做我要检查的工作"。
- 它能检查出我们对关键系统和应用程序的总体可信任程度。

许多审计部门忽略了分布式环境的审计工作，或者只是草草应付了事。出现这种情况的原因在于审计部门缺乏足够的知识，并且组织也缺乏合理的工作约束。还有一种原因在于审计人员存在这么一种误解，即应该把重点放在更重要的大型机系统和应用程序上！这种态度的问题是它没有认识到，由于分布式环境中的脆弱控制，大型机环境和应用程序也会受到安全问题的牵连。由于没有采取一种纪律化的安全和审计方法，组织无法采用分布式的客户机-服务器体系结构和技术。如果组织要使用这些技术，那么安全问题必须要尽可能早地得到解

决，这是非常重要的。审计部门应该尽快地审计分布式环境，这是我们所能提出的最好建议。如果要部署分布式环境，那么组织应该尽早地建立正确的标准和指导原则，而不是事后重新添加这些东西，那样就困难多了。

24.2 审计的角色

对于不同的组织来说，审计和安全部门的角色和职责以及他们同用户群体之间的关系，是各不相同的。在许多组织中，系统审计和安全功能都是IT部门的任务。对于那些负责操作和管理组织计算资源的人员来说，不管安全和审计部门在结构上是否与其保持独立，在功能上他们是必须要保持独立的。图24-1从广泛意义上说明了系统管理员、安全部门以及审计部门的角色。

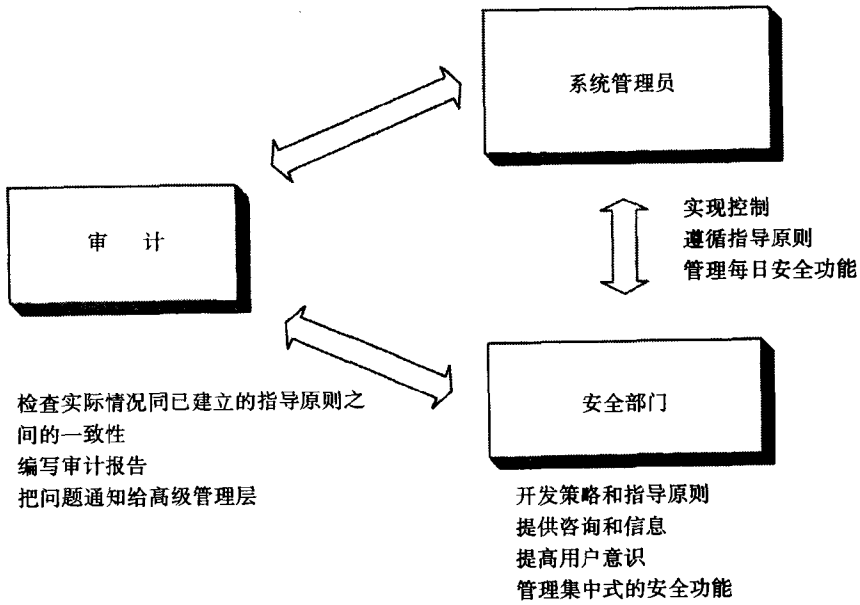


图24-1 审计、安全和用户部门的角色

前面已经提到，审计部门在计算中的角色是对计算环境的安全性进行独立地检查，并且编写一份详尽的检查结果报告。审计报告的本质上是对当前计算环境同可接受的标准集的一致性所做的分析。

通常情况下，标准和指导原则是由安全部门公布的。我们一般把负责安全策略和指导原则的部门称做是安全部门，但是该部门还有很多不同的名字。我们将在24.2.2节中介绍应该如何建立标准和指导原则。安全部门也负责教育用户的安全意识。进行集中式的安全管理任务（例如维护用户ID）一般也是安全部门的责任。在系统上实现公司标准和指导原则一般是系统管理员的责任。另外，他们一般还要管理每日安全功能。

24.2.1 关系

良好的警民关系是保证警察正常工作的基本因素。警察依靠公民向其报告犯罪行为以及提供信息。没有目击者的参与，逮捕罪犯就会非常困难。没有它所服务的公民的支持，警察

局很难开展工作。

审计部门同计算群体之间的关系同警民关系非常类似，上面的结论在这里也是非常正确的。同警察局一样，审计部门必须依靠用户群体的支持才能完成其工作。审计部门应该让用户群体知道他的工作目标，这是非常重要的。审计部门必须要同用户群体和安全部门一起工作才能加快审计过程。在实际审计标准的建立中，这种良好的工作关系是非常重要的。

24.2.2 建立正确的标准

审计就是把当前的环境同一个已建立标准进行比较。在第23章中，我们讨论了安全纪律、策略和指导原则的建立。尽管策略和指导原则通常不是由安全部门首先建立的，但是它们必须要经过审计部门的评审。有些审计部门不愿意认可安全部门所提交的标准，这是一个很常见的问题。他们认为如果公布标准，那么检查的作用就降低了。但事实上，通常反面情况才是正确的！审计部门必须要同安全和用户部门一起工作才能建立一套可被认可和遵循的指导原则。

所使用的标准必须要反映出正审计的环境。审计一台UNIX服务器同审计一台大型机没有太多的共同性。许多指导原则不能实现在UNIX系统上，或者不能解决实际安全问题。利用用户部门的输入，安全和审计必须要协同工作才能建立一套实际的指导原则。实现它们则是独立用户部门和系统管理员的责任。

24.3 UNIX审计标准示例

下面列表列出了几个用于UNIX系统的审计标准。注意，该列表并不详的。

- 每个账号都应该有一个密码，否则禁止登录。
- 超级用户账号的权力应该受到限制。
- 如果不需要，禁用TCP/IP服务。
- 每次用户登录以后，系统应该显示用户的上次登录时间。
- 每次用户登录以后，系统应该显示一条关于该用户授权使用状况的消息。
- “.”不应该出现在PATH变量中。
- 用户的$HOME目录和文件不应该是其他人可写的。
- /tmp和/usr/tmp目录应该有粘滞位。
- 对系统文件和目录的写访问权必须要受到限制。
- 系统应该被物理保护起来。
- 禁用主机等效性。
- 及时应用厂商的安全忠告。

审计标准是把计算策略和指导原则结合起来创建的。一般情况下，它们被编译到一个包含很多项的审计检查列表中。审计检查列表通常涉及了许多不同的问题，这包括密码管理、系统资源访问控制以及连网服务。

审计报告

如果你的孩子在100道考试题目中回答对了98道，但成绩却被判为不及格，那么你会怎么生气？大多数人都会勃然大怒！孩子们还有兴趣继续努力吗？有些审计部门有这样的假设，

任何背离标准的行为都会导致一份拙劣的审计报告。他们没有认识到，由于应用程序或者连网需求，并不是在每种环境中每个指导原则都会得到满足。有些操作系统错误根本不是用户部门所能处理得了的，怪罪他们根本就没什么意义。

我们知道有这么一种情况，虽然某个审计标准在部门服务器上没有得到满足，但是实际的安全暴露问题却被一个补充控制消除了。审计部门向高级管理层报告了一个不可接受的控制故障，这使得那个不幸的系统管理员遭到了令人不快的工作检查。然后，该管理员流露出了对继续处理安全问题的不情愿情绪。对于审计部门来说，更好的做法是根据整体安全性来编写最终报告。如果整体安全很差，那么就可以写一个差的报告。但是如果只是一些小问题没有得到解决，那么盼个“不及格”就是不应该的了。重要的是，报告应该公正地反映出整体安全性级别。它不应该在系统管理员的一点小错误上大做文章。

现在我们已经分析了审计的总体目的，下面我们探讨一下审计应该检查的目标类型。

24.4 计算机审计基础

传统的计算机系统审计主要着眼于三个方面，策略一致性、控制结果检查以及审计跟踪检查。应该指出，许多公司的审计重点起源于大型机计算环境，并且可能偏向那个环境。计算审计的一个重点审计对象是策略和标准的一致性。不幸的是，许多组织所使用的策略只是为大型机开发的。把这种策略用在分布式环境中将会引发很多问题。我们的观点是，策略应该是通用的，它应该独立于任何计算平台。标准是策略的应用，它们需要专门为分布式计算平台开发。

计算机审计的一项重点内容是检查被审计的系统、应用程序或者网络的控制结构。这种检查一般会包括对认证需求的检查，如密码和统一用户ID的使用。密码管理功能，包括健壮密码和密码时效性的使用在内，也要检查。另外，访问控制（例如拒绝未经授权访问的机密信息）和高级访问特权（例如UNIX超级用户特权的使用）也在检查的范围之内。

计算机审计跟踪是从计算机系统的行为中生成的数据集合。审计跟踪提供了关于用户和进程动作的信息，它可以用来跟踪非法动作。审计跟踪的目的包括：

- 监测未经授权的行为，例如通过重复的失败访问尝试来猜测密码。
- 通过记录登录和访问权限中的变化来监测未经授权访问的出现方式。
- 跟踪对系统资源的访问。
- 提供从灾难中进行恢复的能力。

有时候，审计人员会检查审计跟踪以寻找证明未经授权行为的证据。但通常情况下，审计人员只是想保证在出现未经授权行为事件的时候审计跟踪能够跟踪用户行为。

24.5 扩大重点

传统的审计技术和方法都是为大型机环境开发的，它们把重点放在围绕着单一计算机系统的已知控制上。但是随着分布式计算的出现，传统的控制检查机制变成了大画卷中的一个方面，它应该进行扩展以结合分布式计算所引入的其他方面。这些方面包括正确的责任分离、用户意识、业务恢复计划、物理控制、软件许可以及应用程序开发。图24-2说明了在分布式计算环境中扩展审计重点的新行为。

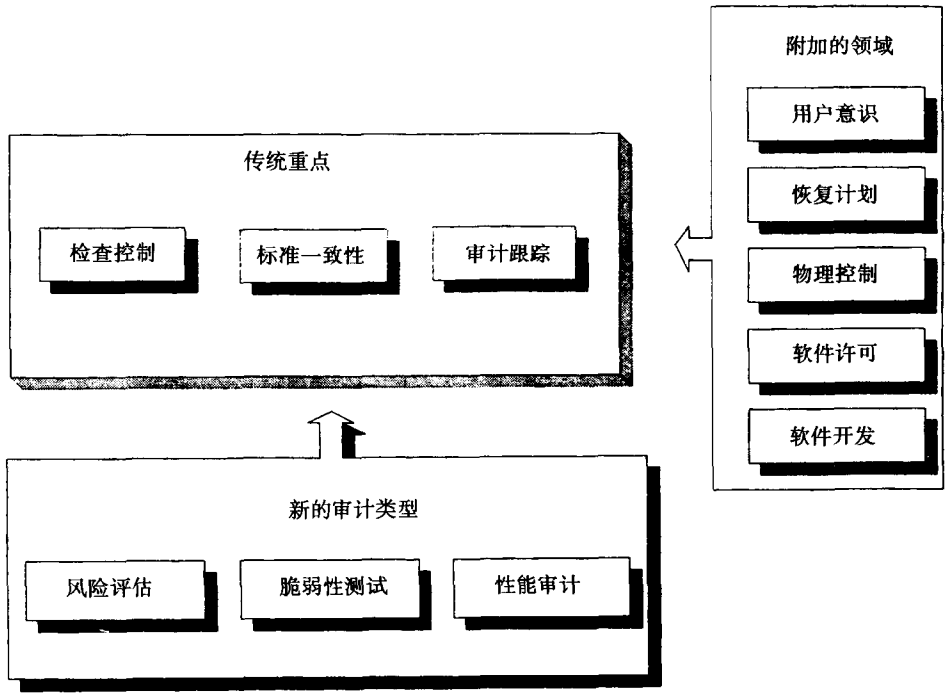


图24-2 提议的安全审计扩展重点

下面我们来讨论这些新的审计领域。

24.5.1 用户意识

用户必须要意识到他们在维护计算机安全中所具有的责任。如果不知道自己的责任，那么用户就不可能维护和实践正确的安全程序。向用户群体灌输安全意识是各级管理人员的责任。意识程序起始于向新员工交流安全问题和程序。员工应该正式地确认他们对安全责任问题的理解。分布式计算审计应该包括对用户意识程序存在性和内容的检查。

24.5.2 业务持续计划

业务持续计划（Business Continuity Planning，BCP）提供一个计划和方法——在出现影响计算服务的灾难以后，组织可以利用该计划和方法进行业务恢复。这是一个复杂的课题，但是它的核心问题——备份调度和灾难恢复计划——已经实现了。分布式计算审计不仅仅要检查备份进程，还要检查应用程序或者系统从灾难中进行恢复的能力。

24.5.3 物理控制

对于一台计算机或者网络设备来说，如果攻击者能够得到对它的物理访问权，那么他就可以通过使用各种技术而得到对该设备的未经授权的访问。分布式计算审计应该检查目标系统和网络设备防御物理攻击的能力。物理控制上的不足一定要得到解决，例如我们可以把关键系统放到绝对安全的环境中。

24.5.4 软件许可

版权法保护了软件厂商创建和发布软件的权利。商业软件包以及一些建议的软件包都在其厂商许可协议中包括了有关软件使用的限制。大多数厂商允许用户拷贝软件作为备份或者工作拷贝。然而，许可协议一般都规定软件一次只能安装在一台机器上。如果要为其他机器拷贝软件，那么用户必须要获得附加许可证或者拥有一份现场许可协议。

拷贝软件用于许可协议声明之外的目的是非法的。分布式计算审记必须要检查软件许可以保证严格支持许可规定和版权法。

24.5.5 软件开发

测试工具和产品工具一般是分离的，因为谁也不愿意测试活动影响到产品机器。然而，公司可能没有对软件从“测试”到“产品”的移动做到足够的控制。我们曾看到过这种情况，软件补丁和应用程序版本没有经过测试就热加载到产品中去了。产品数据在测试环境中的使用也是一个值得考虑的地方。在这里，分类数据（例如个人数据）有可能被暴露给开发人员，并且产品数据和测试数据有可能混淆在一起——如果测试数据不慎加载到产品环境中的话。一般来说，测试环境同产品的分离是必需的。

24.6 其他的审计类型

其他的审计类型有助于我们明确环境的可信任程度，它们包括风险评估、脆弱性测试、自评估和性能审计。

24.6.1 风险评估

虽然风险评估在许多情况下是IT部门的责任，但是它也是计算机系统审计中的一个有用工具。风险评估是结合考虑事件发生的可能性来预计事故或者事件可能造成的损失的尝试研究。进行风险评估的对象可以是应用程序、系统或者网络，它们被称做主体。对主体中的每个控制，风险评估都要估计如果控制破坏的话可能造成的损失。事件发生的百分可能性称做风险系数或者因子，该值要应用到预期的损失上。二者相乘的结果就是一个预期损失数字。我们的目的就在于把重点放在那些具有最高损失数字的问题上，这些问题会为组织带来最大的损失。

> 术语“风险评估”也经常用来描述对计算机系统控制的主动测试。这种测试通常被审计人员称做脆弱性测试，其目的是通过发现现有控制中的脆弱区域来标识潜在的威胁。标识过以后，组织就可以加紧控制，并且避开潜在的威胁。

风险评估所产生的主要结果是一份评估报告——该报告详细说明了控制及其相关的脆弱性，并且点出了如果这些脆弱性被利用的话可能造成的潜在损失。从审计角度上看，使用风险评估可以：

- 允许审计行为把重点放在具有最高预期损失的问题上。
- 把同计算环境相关的风险和潜在损失通知给管理人员。
- 警告开发人员和系统管理人员注意潜在的问题。

风险评估的使用也有一些局限性。如果不从其他源访问数据，那么我们很难确定出一个

合适的风险因子。如果所有系统或者应用程序被破坏了，那就会造成很多间接损失，在确定潜在损失的时候，所有这些间接损失也必须要包括在内，这是非常重要的。例如，破坏一个测试系统可能不会造成重大的财政影响。但是如果因为这次破坏而使得入侵者后来可以访问员工系统，那么这就有可能造成重大损失了。

24.6.2 脆弱性测试

风险评估并不对被检查的环境控制进行任何直接的行为，所以认为它是被动的；另一方面，脆弱性测试真正对环境进行可疑脆弱性的测试。它们模仿和使用黑客所使用的工具来获得对系统的未经授权的访问。一般的检查包括：

- 脆弱的或者不存在的密码。
- 尝试得到密码文件。
- 无保护的系统二进制数据和文件。
- 尝试访问分类数据。
- 利用已知的bug。
- 实现控制不完全。

脆弱性测试的目标在于使用攻击者可能会使用的方法来突出系统中的脆弱性。根据从测试中获得的知识，我们就可以使用一个适当的控制或者对策来消除脆弱性。测试的范围可以是一个系统或者应用程序，也可以包括大量的连网系统。推荐做法是把网络测试作为初始行为，这是因为网络测试具有很完整的途径。网络测试善于发现潜在的脆弱系统，并且可以检查出系统总体安全水平。网络测试完成以后，通常进行的是对关键系统的综合测试。

必须要注意的是，从脆弱性测试中得到的工具和信息绝对不可以被未经授权的个体所利用！当测试完成以后，报告和工具应该从候选系统中删除。用于UNIX和TCP/IP环境的脆弱性测试软件包括捐赠的COPS和SATAN工具。COPS监视UNIX系统控制，而SATAN则提供了测试TCP/IP网络脆弱性的能力。

24.6.3 自评估

自评估审计允许系统管理员对照已建立的审计标准检查他们的一个或多个系统。自评估的目的在于让系统管理员在审计之前满足或者超过安全标准。如果定期使用而不是只在审计部门造访之前才临时抱佛脚的话，自评估可以极大地提高组织的整体安全级别。我们将在后面介绍自评估工具的使用。

24.6.4 性能审计

性能审计是一种特殊类型的审计，它的工作是测量计算机系统和应用程序的效率和有效性。这类审计的重点通常都是应用程序。性能审计的目标是：

- 测量一个系统是否满足它的预期实现目标。
- 分析新系统的成本-作用。
- 测量新系统是否满足它的性能设计目标。

由于进行性能审计需要付出相对很高的成本，所以它们通常都限定在大型的、复杂的系统和应用程序上。

24.6.5 审计时间

财政年度结束以后，财政状况审计主要每3-6个月就进行一次。尽管先前已经进行了一些预备调研，但是大部分工作还是在年后进行。计算机系统审计也遵循同样的循环，它每年进行一次。但是我们认为应该采取一个更及时的方法。如软件许可检查之类的工作可以每年进行一次，但是控制监视和脆弱性检查应该全年进行——作为正常的系统管理任务的一部分。

24.6.6 认可

认可是把安全标准公平地应用到不同类别的系统上的过程。不同的操作系统需要不同的审计系统，许多组织已经认识到了这一点。他们已经开发了一个认可过程，该过程在系统类型及其用途的基础上定义了一套用于认证、访问控制、监视和审计的标准。这并不意味着标准的减少，而是表明在特定平台上可用的全套安全控制都使用了。

不同的操作系统需要不同的对待，这是因为它们的控制结构是不同的。但是许多组织也在功能分类的基础上认可系统，并且根据系统的性质而使用不同的标准。例如，应用程序开发机器一般要比运行工资程序的产品机器具有更低的标准集。但是作为低许可的一个结果，应用程序开发环境可能会被拒绝某些特权，如直接访问公司互联网络。

根据功能，需要认可的系统一般可以分为如下几类：

- 无类别。
- 个人工作站。
- 演示。
- 产品。
- 网络管理。
- 应用开发。
- 安全和审计。

表24-1列出了基于认可分类的建议连网限制。

表24-1 认可需求

认可类别	需求	限 制
无类别	无	系统必须隔离，不能直接连接到公司互联网络上
个人工作站	访问控制 病毒检查	系统是一个单用户工作站
演示	访问控制	系统重新加载了厂商提供的资料。除非按照产品系统得到认可，否则不可以访问网络
产品	访问控制 物理控制 网络控制 审计	经过认可的系统可以访问互联网络
开发	用户控制	系统不能直接连接到互联网络上，而是必须要使用一个已认可的系统上的代理服务
操作支持	按照产品	系统为基础结构（如软件分发和连网）的持续操作提供了支持
安全性	物理控制 访问控制 审计 网络控制	系统提供了同安全相关的功能

认可的主要目的是提高整个环境的总体安全性。不符合认可标准的系统都会拒绝特权，如完全连接互联网络。认可过程强制最脆弱的系统变得更强大——如果这些系统希望拥有一个扩展特权的话。

24.6.7 自评估工具

今天很多自动软件工具都可用于监视计算环境安全。安全人员一般把这些软件工具称做控制监视器，而审计人员则把它们术语化为自评估工具。自评估工具的使用通常都是针对于特定平台的。用于Novell网络的自评估工具有Blue Lance的LT Auditor和Intrusion Detection的Kane Secruty Analyst，用于UNIX平台的有捐赠的COPS软件和VeSoft的Security Audit/UX，另外Axent Technologies的OmniGurad家族产品也具有安全监视功能。

在今天的组织中，大多数的审计部门都受到财政和工作强度的制约，所以我们推荐使用已有的控制监视软件来进行安全性评估。一般来说，只要有，审计就应该使用已有的安全工具。对于审计来说，最好的方法就是促进一个公共的安全监视工具的使用，并且定期检查系统管理员运行它的情况。

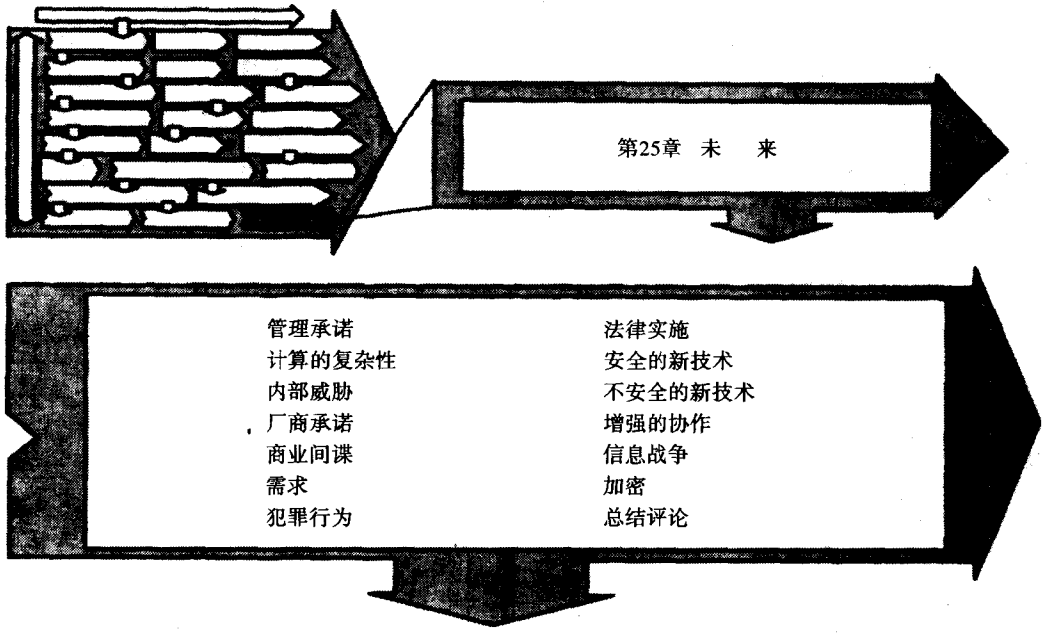
> 审计部门应该促进和采纳子评估工具（控制监视器）的使用，这是我们向你强烈推荐的做法。自评估工具在审计希望建立的标准和系统管理员制定的控制之间提供了一个公共链接。除非有欺骗行为，否则审计部门和系统管理员应该共享公共的安全工具。系统管理员应该定期使用这些工具来检查安全问题。审计人员只能周期性地使用这些工具。审计不应该完全依赖于控制监视器的使用，而是应该把它们的使用同审计检查列表的使用以及与关键人员的会谈结合起来。

24.7 结论

计算机系统的审计是计算安全问题解决方案中的一个重要组件。审计促进了实际环境同已建立的策略和指导原则的一致性，因而提高了企业计算安全的整体级别。

用于大型机的传统年度检查形式不能应用在分布式环境中。在分布式环境中，计算机审计行为的重点、范围以及时间安排必须要改变。传统的控制环境审计应该扩展成包括其他领域，例如许可防备和策略一致性。传统的审计行为也要扩展，不同的审计类型要包括审计内容中，如脆弱性测试和风险评估。然而，更重要的是审计部门必须要同他所服务的计算群体保持一个良好的工作关系。

第25章 未 来



我们能够解决计算安全问题吗？从乐观角度上看，计算安全问题会被解决。新的技术，或者将要可用的技术，将会解决该问题。这些技术具有如下功能，安全的单一登录、强大的访问控制以及安全事故的动态监测。在大多数公司中，管理层逐渐提高的意识和承诺将会让这些高级技术得到利用。另外，实现强大的、健壮的安全控制、监视控制以及教育用户群——这些需要也会得到满足。

悲观者则认为计算安全问题在将来也不会被解决。新技术（例如无线LAN）会带来新安全问题。欺骗以及其他非法行为仍旧继续吸引着组织犯罪的关注。武装了复杂的工具并且使用了欺骗和其他智力犯罪知识以后，组织犯罪会发动一波新的计算机犯罪热潮。他们使用国际边界来保护自己，从而安全隐秘地进行犯罪活动，这一事实对于新的计算机犯罪人员来说是非常有吸引力的。最后，在将来的战争中，敌方的计算资源将会是逻辑目标。对于大多数组织来说，如果外国军事力量要对其计算资源进行先进的攻击，那么这些组织是否有能力保护自己是非常值得怀疑的。

计算安全挑战能被满足吗？这个问题是很难回答的。图25-1揭示了两种对立观点（能和不能）后面的原因。

计算安全问题能否被解决？对此我们很难预测。现在我们来看一下两种答案背后的原因。

管理承诺

在过去，高级管理层几乎认识不到计算安全的重要性。然而这几年来，这种情况在许多组织中都有了变化。高级管理层已经认识到了组织计算资源的安全问题是一个很重要的业务

问题。他们已经承诺要解决该问题，并且使用很多方式表明了这种承诺。在过去，要想让高级管理层为了计算安全问题而接见你，真可谓困难之极。根据我个人认识，这一点已不再正确。许多安全问题都有了高级管理部门的直接参与。他们对安全问题的解决过程和结果都非常热心。在许多组织中，信息安全官员的位置已经从一个信息服务位置提高到了直接向副总裁做报告的位置。高级管理部门的承诺为计算安全问题的解决提供了财政支持。在许多组织中，管理部门的承诺已经不仅仅只限于为问题的解决提供财政支持了。高级管理部门对解决计算安全的承诺对于公司安全行为的成功起到了至关重要的作用。

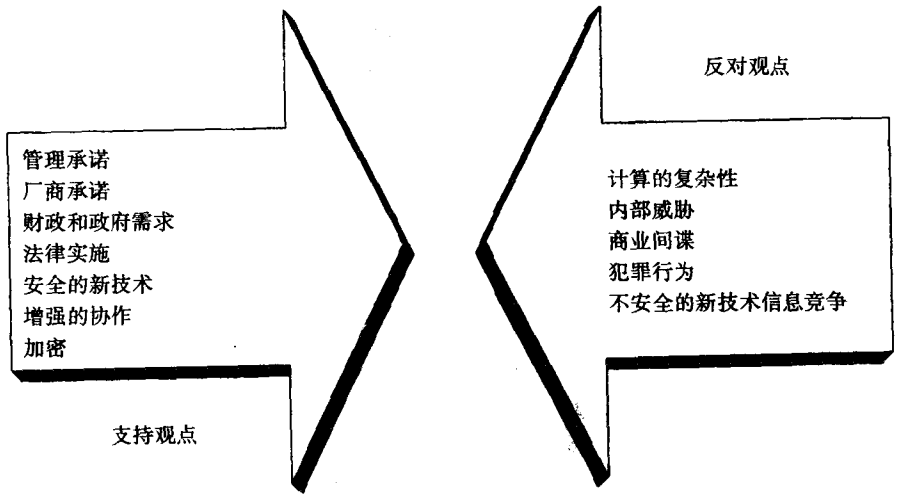


图25-1 我们能对付计算安全挑战吗?

计算的复杂性

新的客户机-服务器技术虽然很容易使用，但是同其他环境比起来，它是非常复杂的。随着数据和应用逻辑的分布化，环境的复杂性提高了。如果考虑到连网需求，那么复杂的计算环境又涉及到了许多不同的方面。从安全角度上看，复杂是非常糟糕的。它增加了出现bug和其他安全漏洞的可能性，入侵者可以利用这些bug和漏洞来攻击系统。高级软件使得分布式服务的实现变得很容易，但是它也提高了最终方案的复杂性。

内部威胁

现在的公司都在进行着巨大的变化。大量的公司裁员和缩减规模使得员工对公司不再像以前那样忠诚。很多事故都同不满意的员工，或者那些只要有机可乘就自饱私囊的员工有关系，相信这种情况还会继续下去。很多员工都有能力使用自己的系统或者网络访问权限进行报复行为，他们的攻击对于大多数组织来说都是很难防范的。这种内部威胁极大地破坏了组织的安全性，因而有可能会为组织带来最大的损失。

厂商承诺

在过去，厂商从不把其产品的安全性作为一项迫切需要来对待。但是今天这种情况已经有了变化，产品的高安全性被看做是有竞争力的优势。许多厂商已经为具体的客户安全需求开发了解决方案。市场上从来没有过如此多的安全产品。然而，这为分布式安全问题的解决带来了困难。各式各样的解决方案都在互相竞争，人们迫切需要有一个工业或者事实标准。但实际情况是，我们不但有互相竞争的解决方案，而且还有互相竞争的标准。

商业间谍

业务中的竞争压力从来没有像今天这么大过。由于全球竞争的压力，公司需要提高质量、压缩成本并且提高产品推向市场的速度，并且这种需要还在日益增长。在解决这些需求的过程中，信息是一个非常关键的因素，并且毫无争议地成为公司可以拥有的最关键的资源。今天，大多数的公司秘密都保存在计算机系统中。由于市场竞争的压力，公司间的机密窃取行为逐渐增多，至少有一些公司不可避免地希望通过这种行为来压倒竞争对手。

工业界秘密不仅仅包括竞争对手的信息，而且还包括外国政府的信息。特别是在新出现的领域中，窃取别国的资源要比自己开发容易的多。我们怀疑，大多数国家都会采取多种形式来纵容间谍行为。使用国际通信链路（例如因特网），商业间谍可以很安全地窃取邻国机密，相信这种行为会逐渐增多。有了国家政府的支持，国家安全局可以利用最复杂的方法和技术来获得竞争对手的信息。

金融业和政府需求

政府部门和金融行业是最需要解决好安全需求的地方。可以预见的是，这两种单位的安全需求将不得不由厂商来解决。他们的市场购买力非常大，因而厂商绝对不可以忽视他们的存在。由于他们的需求和购买力，市场上出现了更强大更复杂的安全解决方案，从而提高整个计算群体的安全水平。另外，他们的需求对标准的开发和部署也会起到积极的推动作用。

犯罪行为

到现在为止，大多数的计算机黑客行为都不是因为钱的原因才发生的。计算和通信服务中的偷窃行为已经变得非常常见，但是通常它们的出现一般都不是为了钱。除了少数一些大案以外，欺骗和电子盗窃之类的犯罪行为一般很少出现。黑客知识和犯罪知识的联姻好像势在必行。当这发生了以后，对商业计算系统的犯罪攻击可能会变得更流行。犯罪分子可以为其同谋提供购买高级装备和计算设备的资金。

这种行为让人头疼的一个地方在于作案者不需要靠近受害者。作案者可以来自一个与受害者所在的国家关系紧张的另一个国家，他们可以在因特网上进行攻击，这种行为是很难遭到起诉的。尽管国际组织正在积极采取多种方式来解决该问题，但是仍然存在很多让人棘手的国家——从这些国家引渡和起诉计算机罪犯是非常困难的。这些国家里的罪犯可以免遭国际起诉。因特网上的欺骗行为也是非常难以起诉的。骗子们使用多种方法来骗钱。这些方法可能只有在因特网上才能发挥作用。如果读者希望深入了解这一重要课题，请参考《Computer Crime in Canada》[Davis & Hutchinson,1994]。

法律实施

在《The Cuckoo's Egg》(Cliford Stoll,1989)一书中，作者引述了一个有关计算机间谍行为的真实故事：在这个故事里，一个入侵者使用了大量的计算机网络。如果要揭发入侵者，本地、州以及联邦警署的协助是必需的。但在早期的时候，这种协助的缺乏是最让人头疼的地方之一。即使一个入侵者有嫌疑入侵中央情报局的计算机，官方的反映也是很冷淡的。今天，大多数管辖范围内的执法机构的反映是不同的。公众已经提高了对计算机犯罪的认识，这也包括执法机构在内。在许多情况下，司法争论以及实际损失的问题使得起诉变得很复杂。北美和欧洲的法律实施团体开始愿意参与对计算机犯罪的调查。北美的许多警署已经有了专门调查信息技术犯罪的特殊小组。

安全的新技术

一些新出现的技术，例如Kerberos，使得我们可以开发强大的、安全的应用程序。一些使用公开加密和其他高级安全技术的新应用程序，例如Lotus Notes，正在逐渐进入市场。很多新厂商也提供了安全解决方案。Windows NT操作系统的开发把安全作为一项需求。只要市场上有需求，厂商就会立即做出响应，并设计出新技术以满足这些需求。每天都会出现新的安全技术和产品。新技术的使用使得入侵者破坏组织计算安全变得更加困难。

不安全的新技术

可以肯定的是，新技术的出现会不可避免地带来新的安全问题。例如，使用无线电波而不是物理电缆进行通信的无线LAN技术就带来了新的安全问题。无线LAN上的传输是基于无线电频率的，只要使用适当的扫描技术，攻击者就可以截获无线LAN的信号。从简单的扫描器到复杂的光谱分析器，只要使用这些工具，入侵者就可以监视公司装置周围的无线电波。实际上，弄清楚传输的意思并不是十分的困难。根据攻击者的水平和所使用工具的复杂程度，分析无线数据流是很有可能的。用于无线LAN的诊断工具可以分离选定的节点以进行故障诊断。然后攻击者也可以使用这样的工具。同样，根据所使用设备的复杂程度，攻击者不需要呆在组织的物理范围内就可以进行攻击。对于许多组织来说，当使用无线技术时，对敏感数据进行加密是必需的。

增强的协作

为了解决计算安全问题，政府、厂商以及计算群体的协作正在逐渐增加。大量有关安全领域的因特网RFC已经说明了人们和组织对共享其安全思想和观点的意愿。大量的委员会和来自各种标准委员会的工作组也表明了人们对解决安全问题的共享承诺。卡内基-梅隆大学的计算机紧急响应组（Computer Emergency Response Team，CERT）和美国能源部的计算机事故咨询能力机构（Computer Incident Advisory Capability，CIAC）会及时地向计算公众通知严重的安全问题，他们的作用可以说是无价的。

信息战争

1995年8月21日，美国时代杂志发表了一篇题为“Cyberway—The U.S.Yushes to turn computers into tomorrows weapons of destruction. But how vulnerable is the home front?”的文章。该文章描述了一个国家的计算资源如何会成为未来战争中的攻击目标。这种攻击同大多数犯罪行为不一样，它的重点不是放在获取钱财上，而是在于如何破坏基于计算机的服务。如果战争合理的话，那么这种攻击就会被合理地从军事站点扩展到商业和政府计算系统上。大多数的商业组织没有能力抵御对其计算资源的集中式的军事打击，至少这种防御能力是值得怀疑的。这些攻击还可以把攻击重点放在公共事业上，例如它们可以破坏用于石油和电气事业、本地和州政府以及执法机构的控制系统。实际上，任何其服务中断会为公众带来影响的目标都可以被看做是一个目标。网络本身的中断会带来巨大的损失。

加密

加密在操作系统和网络控制之上为公司信息资产提供了另一层保护。随着商业和捐赠解决方案逐渐增长的可用性，加密技术特别是公开密钥技术的使用也逐渐在增多。如果密钥管理的问题得到解决，加密技术应该为计算安全问题的解决提供一个强大的武器。然而，加密技术并不排斥强大的操作系统和网络控制。如果一台计算机的操作系统控制是脆弱的，那么即使使用了加密，攻击者也可以成功地使用伪装特洛伊木马程序来窃取用户的加密密钥。

总结评论

如果要解决计算安全问题，那么组织必须要把它看做是一个业务而不是计算问题。新技术肯定会有助于问题的解决，但是它们只是答案的一部分。策略、管理承诺以及对问题的一个战略观点都是必需的。

组织必须要重视计算安全问题。管理部门的承诺是必需的，他们需要分配人力和财力资源。不幸的是，这种承诺往往是在发生一次严重事故以后才流行一段时间。但是分配资源并不是惟一的需求。组织需要有一个安全策略来保证资源的高效利用。该策略要标识那些解决问题所需的动作，其中也获取相关技术在内。但更重要的是，该策略必须要让安全需求同业务目标和方向保持一致。公司策略中应该包含一个进行用户安全意识教育的程序，这是非常重要的。然而，由于目标的广泛性，一个安全策略可能需要数年的执行时间。另外，计算安全问题本身也会发展，从而使解决方案变得落伍。因此，公司需要有一个长期的解决方案承诺。英国的幽默历史（注：W.C.Sellar和R.J.Yeatman，1066和All That（London:Meuthen & Company,1930））已经提供了一段非常适合说明问题长期性的话：

> Gladstone发明了教育速度……，并且在老年时花费了很长的时间来猜测爱尔兰问题的答案。不幸的是，只要他一有点头绪，爱尔兰就会秘密地改变问题。

解决计算安全问题需要花费几年的时间，并且需要大量的承诺。

附　录

附录A　强 认 证

我们已经看到，使用“保密的”密码来认证用户存在着一些问题。首先，通过使用一个字典攻击，任何“保密的”密码都有可能被猜出或者被揭示。密码也有可能被用户记下来被别人看到。但更重要的是，密码通常是以明文形式在网络上传输的。即使密码加密了，它们仍旧有可能被截获并重发。

为了解决这些问题，我们需要实现更强大的认证方法。现在简要回顾一下可用来进行认证的各种形式：你知道的某样东西（例如密码）、你具有的某样东西（例如特殊卡或者令牌）以及个人特征（例如指纹或者虹膜扫描）。我们越需要确信标识的正确性，就越需要有强大的认证机制。在本附录中，我们将介绍一些可用于强认证的方法。

一次性密码

一次性密码是在特定情况下及时生成的惟一密码，它们只在有限的时间内有效。这些惟一的密码是根据一个或多个因素生成的，例如根据时间，或者向一个算法输入一个字符串的事件。因为用来创建密码的输入因素是惟一的，所以密码是不可能重用或者重新生成的。因此，从网络上截获的密码不能重用于后面的未经授权访问。

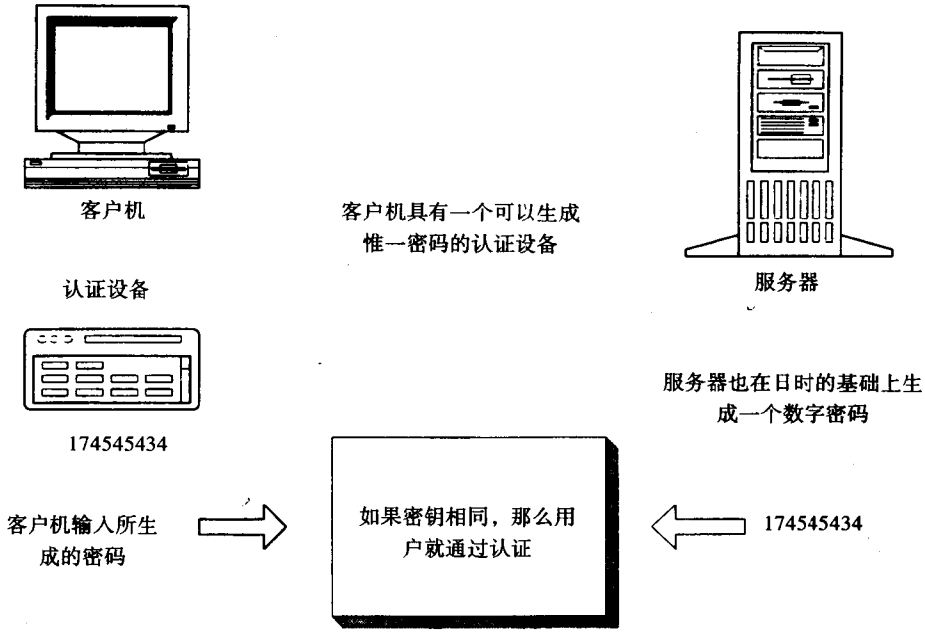
实现一次性密码需要使用到当前时间，把当前时间输入到一个算法中生成一个惟一的数字串。一种简化的实现方法是根据时间和日期生成一个预定义的密码列表。然后用户参考该生成列表以寻找对当前时间有效的正确密码。目标系统也会使用该生成列表。在某个具体的时间段内，列表中的密码将符合系统所期待的密码。

基于令牌的认证

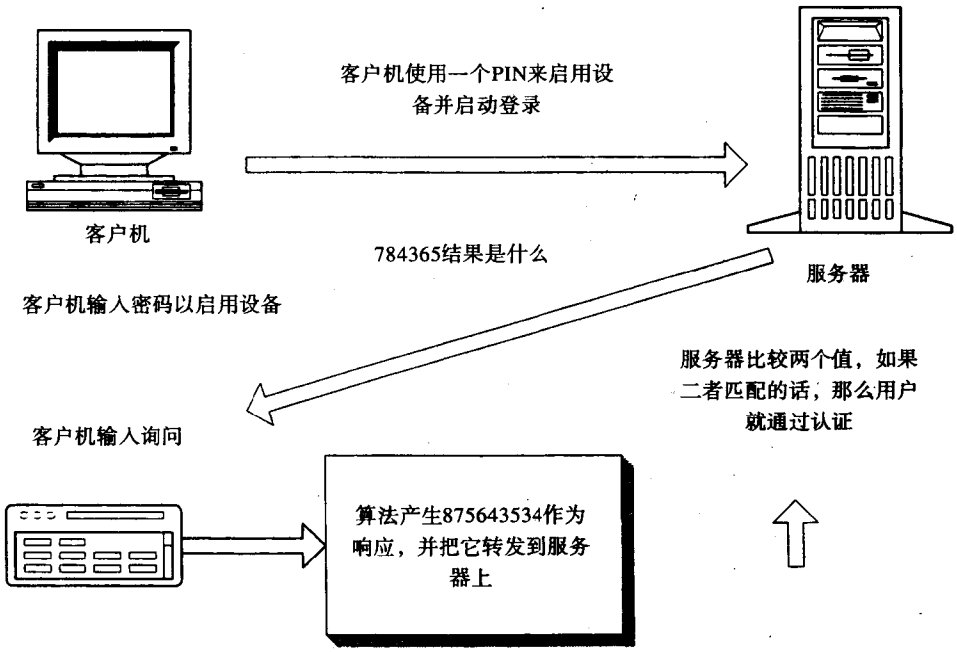
有两个解决方案使用了“你所拥有的某样东西”这种方法，这种方法涉及到了令牌或卡的使用，并且使用了一次性密码或者询问-响应认证。在一次性密码解决方案中，一个手持认证设备或者内部硬件卡用来每隔60秒生成一个有效密码。设备本身受到一个内部保存的密码的保护，在得到生成的密码之前，用户必须向令牌提供该密码。服务器有能力为用户生成针对具体时间段的恒等密码。客户设备生成的密码需要同服务器生成的密码相匹配。然而，基于日时的认证也存在缺点，它们需要协调和维护认证设备和服务器之间的时钟。

这种认证类型要比单一密码认证强大的多。在这种方案中，被截获的密码绝对不会重新用来获得系统的进入权。然而，由于密码通常在60秒的时间内有效，所以攻击者可以在这段时间内截获并重新使用密码。图A-1说明了一个基于日时的双因素认证示例。

询问-响应解决方案并不依赖于一个基于时间的密码，而是通过处理一个询问值来产生一个响应值。在这种解决方案中，用户首先向服务器提出一个认证请求，然后服务器向用户返



图A-1 基于日时的一次性密码



图A-2 询问-响应认证

回一个数字代码。与此同时，服务器使用一个加密算法的拷贝并使用该数字代码生成一个一次性密码。使用一个认证令牌或者软件，用户把该代码输入到自己的本地系统上。本地令牌或者软件使用该数字代码（作为一个密钥）并执行服务器上的加密算法来生成一个一次性密码，并接着把结果发回到服务器上。如果这个密码同服务器所生成的密码相一致，那么用户就通过认证。一旦用户通过认证，密码就不再有效，并且再也不会重用。图A-2说明了这种询问-响应技术。

询问-响应方法的优点在于密码真正是一个一次性密码并且绝对不会被截获并重发。另一方面，许多用户对必须进行两个来回（初始登录请求和询问响应）颇有抱怨。基于事件的一次性密码技术并没有得到用户的青睐，这是因为它们要求用户既要使用一个密码来启用设备又要对服务器的请求提交一个响应。

智能卡的使用可以极大地提高和加强认证方法。有些智能卡实现方案在卡本身中保存了个人标识号（Personal Identification Number，PIN）信息、加密算法、加密密钥以及其他一些受保护的信息。内嵌于智能卡中的微芯片具有处理能力，可用来进行所需的加密。在这种方式中，客户机系统不需要关心加密密钥或者算法的安全存储问题。

使用密码保护的认证设备要比单一密码认证强大的多，因为一个认证因素的丢失不会使攻击者有机可乘。而两个因素同时失败的可能性则是非常小的。另外，这种方案还提供了多种形式的移动强认证。

基于生物特征的认证

基于物理特征或者个人特征的认证是一种更强大的认证方式。这种基于生物信息（你自身具备的某样东西）的认证称做第三因素认证。很多不同的生物特征类型都可用于认证，例如指纹、手印、语音或者虹膜扫描等。击键模式速度之类的瞬时测量也可用来区分人。手写签名也是提供认证的一种方式，但是以图形形式来捕获它是非常困难的。

认证手写签名的一种方式是测量签名笔的速度和移动而不是分析签名结果。这种方式能比分析实际签名提供更好的认证。

由于生物特征数据是通过传感器或者扫描器获得的，因此所得到的结果就有可能是不完整的或者不正确的。生物特征方法的性能是建立在统计学上的误拒绝和误接受百分比上的。当一个有效的个体被认证过程拒绝时，这就发生了一次误拒绝；当一个无效的个体被认证过程接受时，这就发生了一次误接受。如果不对生物特征数据进行另外加密，网络上的生物特征认证仍旧有可能被截获并重发。

基于位置的认证

基于地理位置的认证也是一种可用的认证形式。这种认证方法需要认证方精确地定位用户的具体位置，然后把结果同所预期的位置进行比较。通过全球定位系统（Global Positioning System，GPS）卫星所提供的信息，认证方可以生成一个具体的位置特征。然后认证系统把结果同所预期的客户机系统位置（数字形式）进行比较。位置特征随着卫星的轨道移动而变化，因而攻击者无法进行重发攻击。使用这种方式，连续的认证也是可能的。客

户机也可以认证服务器的具体位置，从而做到互相认证。当一个用户的标识设备被窃取了以后，如果窃贼再继续使用该设备的话，那么认证系统可以查明窃贼的精确位置。

高级认证标准

联邦信息处理标准出版物190（Federal Information Processing Standards Publication 190，FIPS 190）为用户选择使用高级认证技术方案提供了指导原则。该文档是美国国家标准和技术协会（National Institute of Standards and Technology，NIST）提出的，它是对联邦部门和机构进行设计、获取以及实现认证系统的指导说明。这份指导书对包括密码技术和生物实现在内的多种认证方法提供了一个很好的概括说明。另外，它还概述了认证加密技术，并且提供了实现指导原则。

附录B 智 能 卡

智能卡通常指的是内嵌有微型芯片的卡，它们具有同信用卡一样的大小和形状。这种微型芯片可以提供只读的和可更新的数据存储器，并且可以包含一个带有操作系统的处理器，它具有执行存储程序的能力。使用一个特殊的读卡器，我们可以访问卡上的微型芯片（使用卡上的暴露触点）。有些智能卡不需要有触点，它们可以通过电子感应进行通信——只要让它们接近读卡器就可以。还有一些智能卡实现利用了可在远处读取和更新智能卡的无线电频率。这种类型的智能卡广泛用在收费公路或者收费桥对车辆进行的交通收费上。当车辆驶近收费站时，就可以读取卡上的信息。

最简单的智能卡是存储器卡，这种卡上内嵌的微型芯片具有特殊的数据存储能力。法国公共电话系统是这种类型智能卡的最早应用之一。智能卡可用来存储一定数额的预付资金，从而取代了硬币。例如，它可以包含25块钱的信用量。如果要打公共电话，只要把这种预付卡插到读卡器里面就可以了。通话结束以后，电话会自动计算通话费，并从卡上所保存的数额中减去相应的数额。这种卡可以一直使用直到卡上的钱花光为止，这时候持卡人需要买一张新的预付卡。

多个或者个人存储卡是存储卡的一个变种。这种卡上的数据存储可以分为几个独立的区域。对这些区域的访问受到了独立密码或者个人标识号（PIN）的保护。这种类型的卡可以用来存储具有不同用途的多项信息，也可以使用密码或者PIN来保护不同的特殊数据。更高级的智能卡可以在卡上带有一个处理器，并且拥有一个操作系统。这种处理器可用来执行卡本身的程序，从而可以实现非常复杂的应用。这种类型的卡已经用来存储和执行卡上的加密算法。加密算法和密钥保存在卡上，并且可以随处移动。这降低了管理函数的复杂程度，从而可以在特定的实现中提供强大的安全性。

除了基于处理器的卡以外，有些卡还带有一个小型键盘和显示器。这种类型的卡也称做令牌，它们的使用可以不需要读卡器。这种卡有时候用来提供安全凭证和一次性密码，或者提供询问-响应能力。卡上键盘的作用是允许用户输入一个PIN来证明自己是卡的真正拥有者，从而可以启用卡。

智能卡的用途

智能卡广泛用于同安全相关的多种不同功能。智能卡最常见的用途是提供第二和第三功能认证。第二功能认证指的是用户所拥有的某样东西（例如智能卡），第三功能则是确切的标识用户（例如指纹或者虹膜）。为了支持第二功能认证，智能卡有时候用来生成一个只能被用户访问的秘密数据。例如，用户输入一个PIN以启用智能卡，然后访问卡上的安全存储区域以便执行卡上的一个程序。下面是使用智能卡来支持安全需求的一些实现例子：

- 询问-响应——目标处理器以随机数的形式生成一个询问。然后智能卡处理该随机数，并且使用某种加密算法和一个保密密钥生成一个响应。该响应同处理器提供的一个计算好的响应进行比较。如果卡的响应同计算好的响应一致，那么该智能卡以及它的用户就

通过了认证。启用智能卡功能首先需要用户输入一个PIN。

- 加密——加密算法可以保存和执行在卡本身上。需要加密的字符集合被送到卡上，然后加密过的字符集合返回。保密的加密密钥不会泄漏，并且也永远不会离开卡。在这种方式下，PIN可以内部验证而无需通过网络发送。卡也可以用来生成附加到消息上的电子签名。
- 保存照片——使用智能卡来保存可用于认证用户的数字照片。
- 一次性密码——卡上的密码生成机制是与目标处理器上的机制同步执行的。每次使用卡的时候，它就生成一个惟一的密码，并与目标处理器上生成的密码进行比较。这个密码是惟一的，并且在一个短的时间间隔内就改变。如果两个密码一致，那么用户就通过了认证。
- 生物数据——数字化的指纹、虹膜扫描或者其他个人惟一信息保存在卡上。这些信息用来同用户的实际特征进行比较。
- 凭证的安全存储——卡可以用来安全地存储凭证或者会话密钥。卡也可以用来安全地存储分布式的加密密钥。
- 单一登录——卡可以用来存储一套标识和密码，这些数据本身受到一个加密PIN号或者密码的保护。这种安全的存储可以被访问，并且标识和密码可用来登录到各种应用程序或者系统上。这种方法安全地解决了多标识和多密码问题，同时不会要求用户非得记住这些信息。

标准

当前用于智能卡的标准是由ISO 7816标准集规定的。这些标准定义了卡的物理特性以及卡上微型芯片的尺寸和位置，它们还规定了电子信号和传输协议，以及用于交换的工业间命令。内嵌的微型芯片是使用位于卡上特殊位置的8个标准触点来访问的。ISO正在积极地把7816标准集扩展到应用程序和公共数据元素的注册领域中。当前的ISO规范包括：

- ISO 7816-1：标识卡——integrated circuit(s) cards with contacts; Part 1; Physical Characteristics, 1987（带有触点的集成电路卡；第1部分；物理特性，1987）
- ISO 7816-2：标识卡——integrated circuit(s) cards with contacts; Part 2; Dimensions and Location of Contacts, 1988（带有触点的集成电路卡；第2部分；触点尺寸和位置，1988）
- ISO 7816-2：标识卡——integrated circuit(s) cards with contacts; Part 3; Electronic Signals and Transmission Protocols, 1989（带有触点的集成电路卡；第3部分；电子信号和传输协议，1989）
- ISO 7816-2：信息技术（标识卡）——integrated circuit(s) cards with contacts; Part 4; Inter-Industry Commands for Interchange, 1993（带有触点的集成电路卡；第4部分；用于交换的工业间命令，1993）

分布式计算环境（DCE）正在试图把智能卡技术添加到它的当前安全功能中。DCE智能卡的介绍和规范包含在开放软件基金会的两个RFC文档中：OSF DCE RFC 57.0——Smart Card Introduction（智能卡介绍）和OSF DCE RFC 57.1——Smart Card Integration（智能卡集成）。介绍文档说明了智能卡概念以及在DCE中使用它们的通用方法。集成文档说明了如何把

智能卡用于DCE的双因素认证。提议标准调用智能卡存储用户的长期DCE密钥。为了访问该密钥并登录到DCE中，用户必须拥有智能卡，并且必须知道访问卡上的密钥文件所需的密码。

PCMCIA卡

另一项用来实现智能卡系统的技术是PCMCIA卡。成立个人计算机内存卡接口协会（Personal Computer Memory Card Interface Association，PCMCIA）的原因是希望为具有信用卡般大小的PC插入适配器开发一套标准。PCMCIA卡同信用卡尺寸相当，但要厚一些。PCMCIA卡定义了三种类型的卡，其中每个都具有不同的厚度。PCMCIA卡提供了扩展的数据存储和处理能力，它们广泛用来为笔记本电脑和膝上机提供调制解调器和LAN适配器。它们也提供了附加的存储器（RAM内存或者硬盘），并且可用于通信设备（以调制解调器和LAN连接的形式）。许多PC厂商都把PCMCIA功能包含在他们的机器中，这保证了该标准能够长期存在下去。

智能卡的问题

智能卡本身可以包含重要的信息或者安全访问功能，这是使用智能卡必须考虑的主要问题之一。如果卡失效了或者丢失了，那么其他资源可能就不容易可用了。关于智能卡的标准正在继续演化。ISO标准规定了卡的物理特性，但是卡的使用标准还没有达到同样的水平。如果所有的系统访问点都需要有一个读卡器，那么成本是非常高的。因为许多笔记本和膝上PC都把PCMCIA能力作为一项标准配置予以提供，所以芯片卡可能不需要加入到PCMCIA中。智能卡为安全实现提供了增强的能力，但它本身不是一个解决方案。

尽管存在着这些问题，智能卡的使用仍旧会越来越广泛。它们为认证密钥、加密算法以及其他安全机制提供了一个安全的存储机制。这样为计算安全问题的解决提供了一个有价值的工具。

附录C 个人计算机的安全

使用DOS、OS/2以及Windows环境的个人计算机具有独特的安全问题。使用这些操作系统的个人计算机几乎没有什么安全控制；即使有，它们也是非常容易被破坏的。这些控制通常只限于屏幕锁定和启动密码。屏幕锁定控制在一段无交互时间以后锁定屏幕，如果用户要继续操作机器，他们必须首先输入一个密码。有许多不同的屏幕锁定实现，但是通过重启系统，它们中的大多数都可以击破。

有些个人计算机允许使用密码来保护启动过程。同样，只需要简单的从一个软盘上启动，这些控制通常就可以破坏。保存在个人计算机上的屏幕锁定密码有很多问题。例如，有种屏保程序把屏幕锁定密码以加密的形式保存在文件control.ini中，如果该文件被拷贝，那么通过使用前面第12章中讨论过的密码破解技术，加密过的密码字符串就可以被破解出来。破解了屏幕锁定密码以后，攻击者就可以访问PC上的所有一切，所以安全风险就不仅仅只限于个人计算机及其数据了。用户可能在别处也使用该密码，从而造成更大的安全暴露问题。通过发现屏幕锁定密码，攻击者可能能够访问网络或者其他更重要的计算系统。

一些个人计算机软件，包括数据库、字处理器以及电子表格软件在内，提供了加密选项。不幸的是，它们所用的加密通常是脆弱的，并且可以很容易击破。因特网上的很多工具都可以用来破解流行PC软件产品所使用的加密方案。

基于DOS或者DOS/Windows的系统还有另外一个问题，即它们不是为多用户设计的。只要获得了系统访问权，任何人都可以完全访问所有的文件和目录。Windows 95提供了有限的多用户能力，但不幸的是，它是不安全的。WIN95提供了一个通过用户ID和密码保护的登录画面。然而，通过按Cancel或者Escape键，可以很容易的绕过默认设置。这会让你进入一个默认的用户设置。在该设置中，可以访问同样的系统文件和目录——只要有一个有效的密码！通过限制授予给默认用户的资源，安全性可能得到提高，但是这需要深入了解Windows 95的配置知识。普通用户很难实现这些控制。

在讨论用于个人计算机的安全解决方案之前，我们先来探讨一下这种计算环境中的最让人头疼也最具有破坏力的问题——病毒！

C.1 病毒

计算机病毒是造成桌面系统瘫痪的最常见的原因之一。造成损失的原因在于病毒导致的系统中断、某些病毒的破坏性，以及修复瘫痪机器所涉及的工作。对于该问题来说，病毒是最贴切的名字。如果不做防备，病毒会在毫无警告的情况下在“干净”机器上发作并扩散。有些病毒可能并不会造成什么破坏，但有些病毒则会造成极大的破坏效果。不管哪种，它们都会降低机器的工作效率。

一种相对较新的病毒形式已经攻击了字处理系统。原来认为，不可能存在由一个数据文件组成的病毒。但是当伴随着字处理数据文件的宏病毒出现以后，这种看法就不再正确了。当使用字处理软件打开或者保存文档时，宏就会被执行。与此同时，病毒就会被邮寄给你。

当发现计算机感染了病毒以后，最常见的响应就是使用一个病毒扫描器来查找并抑制任何发现的病毒。这种扫描器通常是一个软件程序，它可以主动扫描所有内存和磁盘存储器以寻找表明已知病毒的特定数据模式。发现以后，扫描器就会抑制它们。重新构建已经被破坏病毒修改过的数据通常是非常耗费时间和精力的工作。

使用反病毒程序来摆脱受感染的设备应该是你的最后一道病毒防线，而不是第一道防线。病毒管理程序应该具有一个可用的和定期的病毒扫描能力。组织应该实现一个具体的病毒安全策略以概括出保持低病毒感染率的责任。该策略应该限制个人软件在公司计算机上的使用。如果要在公司计算机上使用一个公司之外的磁盘，那么首先要对该磁盘进行查毒工作。公司应该定义一个在发现了病毒以后报告和处理病毒的过程。如果不认真对待，病毒有可能会严重消耗公司资源。

C.2 个人计算机访问控制

如果要解决桌面计算机的安全问题，那么除了迁移到具有全面安全控制的操作系统（例如UNIX或者Windows NT）上以外，剩下的办法就只有实现访问控制软件了。使用一个软件包，例如Mergent的PC DACS或者IPE的Protec，我们可以把安全性引入到个人计算机中。这些解决方案一般通过要求在登录时提供用户ID和密码来保护个人计算机。登录以后，用户可以被限制到已选定的文件、目录和程序上。

大多数的访问控制解决方案都允许对文件或者目录进行加密（可选）。在这种情况下，即使用户访问控制被破坏了，机密数据也受到了保护以免遭发现。这些解决方案所使用的加密一般是基于工业标准DES或者RSA算法的。一般说来，它们要比许多流行的PC产品所使用的加密方案健壮的多。在这些解决方案中，用户行为的审计跟踪也是一个常见的功能。

C.2.1 膝上PC

膝上电脑的使用带来了很多特殊的安全问题。移动员工经常在远程地点或者在家中使用膝上PC来办公。膝上PC的主要用户是管理人员、销售人员以及其他白领专业人士。它们是可移动的，但价格昂贵，这使得它们要比其他类型的个人电脑具有高得多的失窃率。如果自己的膝上PC落到了其他人的手中，这使他们可以查看该机器里的任何机密信息。许多组织已经强制员工必须要在膝上机上使用PC访问控制软件，并且必须要对敏感数据目录进行加密。这样，即使员工的膝上PC失窃，公司信息也不会丢失。当接收一台膝上电脑的时候，用户也会被警告谨防窃贼。他们不应该把一台膝上PC留在汽车里，并让路过的人看见。当员工在外携带膝上PC的时候，一定要小心谨慎。当出差时，一定要在酒店里看管好自己的膝上PC。员工应该意识到，如果小偷在工作时间知道他们把膝上PC留在办公室里，那会带来什么风险。在这种情况下，西装笔挺、手拿公文包的小偷可以大摇大摆地进入办公室并顺手拿走三台膝上电脑——整个过程不超过30秒钟的时间，可能会有20多个员工目睹这一幕，但肯定谁也不会问一句！

C.2.2 数据安全

最近媒体报道了很多安全信息泄漏事故——在这行事故中，很多敏感数据，包括医疗记录和其他信息在内，被不小心泄漏到公众中。有时候，旧计算机被送到公开拍卖地进行处理。

尽管拍卖者已经小心谨慎的删除了所有的文件和目录，但是要知道有时候删除过的文件是可以被“反删除”的——相信当责任方知道这一点以后，他们一定会非常震惊！在DOS操作系统中，删除文件只是删除文件在文件分配表（file allocation table，FAT）中的对应项。除非被其他数据覆盖了，否则实际文件仍然会留在磁盘上。反删除过程就是重新建立文件分配表，并重新建立指向仍旧留在磁盘上的实际数据的指针。

与此类似，如果软盘没有正确清除的话，机密信息也会泄漏。许多组织要求软盘和硬盘必须清除——使用商业的或者流行的文件删除工具。这些工具通过多次重写数据区域做到真正的删除文件。然而，使用非常复杂的工具，一些专门的服务可以从磁盘上恢复已经删除和重写的数据。当处理用来保存高度机密信息的磁盘时，请务必小心！

C.3 安全应该扩展的程度

当设计一个计算应用程序或者总体环境时，一个重要的安全考虑是，安全性是否只应该在服务器上实现？应该加密网络流量以防止密码欺骗或者重发攻击吗？如果网络安全实现了，那么需要在个人计算机上实现另外的安全控制吗？如果个人计算机用来存放机密数据，那么访问控制就应该是必需的。然而，即使不把敏感信息保存在个人计算机上，它们也有可能遭到特洛伊木马攻击。这种攻击类型会使用一个可以截获用户密码的伪造程序来替代PC程序（例如网络登录工具）。

C.4 结论

在过去，安全工作只集中在服务器上（例如大型机安全）。近来，人们的关注焦点已经移动到了网络上。客户机-服务器安全的根本在于一个把安全性从用户的键盘或者鼠标扩展到服务器上的环境。无缝的安全性应该集成到客户机-服务器模型中的全部三种主要组件中：客户机、服务器以及网络。

附录D 远 程 访 问

员工可能需要在组织的物理位置以外访问组织的应用程序、系统以及网络。具有这种需要的用户包括在家中工作的员工、厂商、公司支持人员以及移动的路边打架者。对于大多数组织来说，解决方案就是在串行电话线路上提供一个直接的拨号访问，或者使用一个公共网络（例如因特网）提供间接访问。因为这两种解决方案都会被外部人员访问，因此如果一个组织没有正确地保证其系统安全，它就会面临非常大的安全风险。我们已经在第14章中讨论了很多同使用因特网有关的安全问题。串行电话链路也可以被破坏。使用war dialer的工具（该工具访问一个给定范围内的所有电话号码以寻找一个与其通信的计算机），黑客可以很容易地发现用于远程访问的电话号码。幸运的是，目前市场上已经有了用于安全远程访问的解决方案。

然而，这些解决方案必须要用同户对连接能力、性能以及易用性的需要取得平衡。一般情况下，用户也需要在办公室里的桌面机上访问同样的应用程序和服务。通常这意味着访问大型机和基于LAN的系统。同大多数的安全机制一样，一个无效的解决方案会强迫用户群体实现他们自己的解决方案。使用廉价的远程访问软件，只要有一台计算机和一个调制解调器，用户就可以很容易地实现能够对公司网络提供远程访问的解决方案。

集中式解决方案

大多数组织都是通过一个集中式系统来满足远程用户的需要。这种整体解决方案不但要比用户实现的解决方案便宜，而且它还可以引入标准化的安全控制，例如审计跟踪和安全认证。然而，一个笨重的或者不提供对所需系统或网络进行访问的集中式远程访问解决方案可能会逼迫用户群体必须自己实现访问解决方案。用户实现的解决方案通常要比集中式的解决方案具有更差的安全性，并且会为组织带来严重的安全风险。

最常用的远程访问控制类型涉及到了用户ID、密码以及回拨安全性的使用。串行线路上的密码被看做是最低级的保护单位。黑客可以使用密码猜测攻击来获得非法系统入口。许多组织也要求使用一个回拨机制，当用户提供了用户ID及相关密码以后，回拨机制就挂断电话，然后在一个预定义的位置上回拨用户。这种方案对于远程的静态用户来说是非常好的。但对于那些不断改变拨叫电话位置的移动用户来说，这种方案是不可接受的。然而，通过保持电话线路的忙状态，黑客也可以破坏回拨方案。因此，许多组织要求使用在附录A和B中讨论过的令牌或者智能卡来实现强大的安全认证。

加密

即使使用了强大的安全认证，在公开线路或者因特网之类的不可信媒介上传输敏感数据仍然可能需要对数据进行加密。许多远程访问解决方案提供了对通信的按需加密或者完全加密功能。这些解决方案也提供了用户访问的审计跟踪，其中包括访问时间、访问长度以及网络访问。然而，远程访问解决方案一般并不提供一个完整的用户行为审计跟踪。尽管对一个

服务器的初始请求可能会记录下来，但是对另外的系统或者服务器所进行的网络访问一般并不会记录下来。

远程支持

许多厂商提供了串行电话线路上的远程故障诊断支持。除非支持队伍和客户有一个非常亲密的关系，否则大多数厂商都不愿意把他们的认证设备或者密码提供给厂商。支持人员对系统和公司网络的访问权通常只是在需要的时候才授予。几乎没有组织愿意把访问权扩展到厂商。

可以解决该问题最常见的方案是切断支持调制解调器的电源，或者把它放在一个无效的端口系统上。当厂商需要访问时，本地系统支持人员就打开调制解调器的单元或者激活端口。连接受到一个密码的保护，该密码是厂商支持人员在电话上提供的。厂商支持人员进入系统或者网络以后，该密码就会改变。

结论

扩展对公司系统和网络的访问权带来的问题是可以被解决的。集中式的远程访问解决方案可以用来解决问题。使用双因素认证设备和集中式审计跟踪，它们提供了强大的认证机制。用户通过认证以后，他同系统所进行的通信就会受到加密保护。然而，当考虑安全问题时，解决方案必须要同用户对连接能力和性能的需求取得平衡。如果用户没有在二者之间取得一个正确的平衡，那么会导致用户群体实现自己的解决方案。

词 汇 表

access control（访问控制） 允许对能够利用计算资源的人或者东西进行限制的访问机制。

Access Control List（访问控制列表，ACL） 由用户和与其相关的访问权限组成的一个列表，当系统确定是否应该把对资源的访问权授予给某个用户的时候，就会检查该列表。

ACID 原子性、一致性、隔离性、持久性。对于事务来说，如果组成该事务的操作全部完成，或者整个事务返回到操作开始之前的状态，那么该事务就具有“原子性”；如果该事务成功地把系统或者数据库从一个有效状态转换到另一个有效状态，那么该事务具有“一致性”；如果该事务虽然同其他事务并发执行，但只有在它完成以后才能被其他事务看见，那么该事务具有“隔离性”；如果当该事务提交时它对数据库所做的所有改变都成为永久性的，那么该事务具有“一致性”。

accountability（可计算性） 在计算机系统或者网络上重新构建事件的能力。它允许系统或者网络管理员跟踪在安全事故中实际发生的步骤并检查个体责任。

accreditation（许可） 根据一套预定义的特殊安全需求来评估安全控制。

Administrator Account（管理员账号） Windows NT系统上的一个具有高级特权的缺省账号。

Analysis of Risk（风险分析） 参见风险分析（Risk Analysis）。

Anonymous FTP（匿名FTP） 一个允许用户使用用户ID“anonymous”进行登录的特殊FTP实现。在这种实现中，用户的邮件地址用做登录密码。通过改变环境的根目录（chroot）保护文件。

API 应用程序编程接口。

APPC 应用程序到应用程序通信（Application Program-to-Program Communication）的简写形式。该协议由IBM所定义，IBM把它用在系统网络体系结构中以支持运行在独立操作环境下的应用程序之间所进行的通信。

Application Gateway（应用网关） 一种防火墙类型，它根据应用程序的通信性质进行访问判定。

APPN 先进对等联网协议（Advanced Peer-to-Peer Networking Protocol）的简写形式。该协议由IBM所定义，IBM把它用在系统网络体系结构中以支持计算系统之间所进行的通信。

ASCII 美国信息交换标准码（American Standard Code for Information Exchange）的简写形式。它是一种规定如何把数字值指定给字母、数字、标点符号以及控制字符的标准编码方案，它为异种计算平台提供了互兼容性。

audit（审计） 独立的、有知识的个体对计算系统或者网络进行的检查。安全审计的结果一般是对所检查的计算系统同可接受的标准之间的一致性所做的一份正式分析报告。

Audit ID（审计ID） 同每个用户相关的ID，用在构建审计跟踪中。

Audit Trail（审计跟踪） 一套记录，以时序顺序列举计算系统或者网络设备上的选定事

件。

authentication（认证） 建立个体标识或者程序标识的过程。

authorization（授权） 决定是否授予对资源的访问权。

availability（可用性） 计算系统对用户的可用量。

awareness（意识） 在安全术语中，意识指的是用户群体对对计算安全、安全措施以及同用户群体所需的安全性相关动作所具有的重要性的一般认识程度。

B1 Security（B1安全性） 一个操作系统安全级别，由TCSEC定义，该级别需要很多安全措施，包括强制的访问控制。

Bastion Host（堡垒主机） 直接同不可信网络进行交互的计算机系统。它们应该具有强大的安全控制。

Berkeley Services（伯克利服务） 也称做“r”服务（rlogin、remote shell——远程shell和remote copy——远程拷贝）。这些服务向其他可信系统扩展主机等效性。

Biometric Based Authentication（基于生物特征的认证） 基于特殊生物信息（例如指纹或者虹膜扫描）的认证。

Boot、Bootup（启动） 计算机系统进行初始化的一系列步骤。

Business Code of Conduct（管理业务代码） 一套有关员工业务行为的正式规定，通常由高级管理部门批准。

C2 Security（C2安全性） 一个用于操作系统的安全级别，由TCSEC定义，该级别指示了包括保护加密密码字符串在内的许多安全措施。

CERT 计算机紧急响应组（computer emergency response team），致力于跟踪计算机安全事故。

certificate（证书） 一套信息集，被转换成不可伪造的形式，用于认证。

Certification Authority（认证中心） 被一个或多个用户信任的管理中心，它可以为用户创建和指定认证证书。

Change Management（改变管理） 一个已定义的过程，用于有序地管理从一个定义状态到另一个定义状态的改变。

Challenge-Response（询问-响应） 一个过程，首先生成一个惟一的值或者标识符，然后作为一个询问发送到一个已定义的算法过程，然后该算法产生一个响应。如果该响应同预期值相等，那么用户就通过认证。

chroot 一个UNIX系统调用，它允许系统管理员把某个目录变成一个用户的根目录。这种行为有效地限制了用户遍历UNIX文件系统的能力。

criteria（标准） 在安全术语中，标准指的是可以作为测量安全控制根据的基础或者标准。

cryptanalysis（密码分析） 在不借助加密过程使用的加密密钥的情况下，使用加密算法把加密过的文本转化成普通文本的操作。

cryptography（密码术） 一门科学，使用一个专门的编码系统和一个编码键，把信息打乱成不可理解的格式再把信息还原成可理解格式。

CICS 客户信息控制系统（Customer Information Control System）的简写，这是IBM为支持事务执行而开发的一个程序产品。

Client/Server Computing（客户机-服务器计算） 一种计算类型，其中一台计算机系统（客户机）通过网络同一个资源提供者（服务器）进行通信。单一计算机可以同时扮演这两个角色。

Clipper Chip（Clipper芯片） Skipjack加密算法的一种硬件实现。

COFC 商业面向功能分类（Commercial oritentated functionality class）。欧洲计算机制造商协会为计算环境的商业使用开发了一个安全标准。

Command Interpreter（命令解释程序） 一个把击键截获为命令并根据其内容来执行动作的计算机程序。

Comment Line（注释行） 在大多数UNIX文件中，注释行指的是任何以一个“#”开头的行。操作系统会忽略注释行。

commitment（承诺） 管理部门和用户群体对解决计算安全问题的渴望程度。

Compliance Monitoring（一致性监视） 主动检查安全控制不但存在而且功能正常的动作。

confidentiality（机密性） 保护信息免遭未经授权的泄漏。

consistency（一致性） 对于一个过程来说，如果同一输入会导致同样的结果，那么被认为该过程是一致的。

Cooperative Processing（协作处理） 指的是把应用程序处理和逻辑分割到一个客户机和一个服务器上的能力。它们都参与应用程序的业务逻辑的执行。

credential（凭证） 安全环境（例如OSF/DCE和Kerberos）用来保存安全信息的一个安全令牌。

cron（时钟守护程序） 一个处理调度程序，是UNIX操作系统上的标准功能。

daemon（守护程序） 一个后台运行并且等待请求以执行任务的UNIX程序。

Data Classification（数据分类） 根据信息的敏感程度对信息进行标类。数据分类用来保证实现正确的控制以保护敏感信息。

Data Custodian（数据保管员） 一个负责保护公司信息资源的保管人。

Data Encryption（数据加密） 一种信息保护方法。加密后，实际数据被打乱，阅读者只有使用一个特殊的密钥才能阅读加密过的数据。

Data Guardian（数据管理者） 负责保护公司信息资源的个体。

Data Owner（数据拥有者） 创建一个信息资源或者被给与一个信息资源的拥有关系的个体。

Data WareHouse（数据仓库） 面向主体的、集成的、易变的、永久的数据集合，主要用来支持组织决策。

DB Groups（DB组） 组是由具有同一套数据库访问特权的用户所组成的集合。

DCE 分布式计算环境（Distributed computing environment），一个基于远程过程调用的综合中间件环境，它允许开发安全的、客户机-服务器应用程序。

DCE Cell（DCE单元） 包括DCE客户机和DCE服务器的管理域。

Debit Card（借卡） 标识卡，用于辅助金融事务（通常是资金转移）的直接访问和操作。

Delegation of Authority（权力委托） 转移部分高级特权，转移的特权受到时间和功能

的限制

Denial of Service（拒绝服务） 一种攻击类型，试图拒绝合法用户访问公司计算资源。

DES 数据加密标准（Data encryption standard），IBM开发的一种私有密钥加密技术。

detection（监测） 一个计算机系统、应用程序或者组织发现其资源被未经授权使用的能力。

Device File（设备文件） 一种特殊类型的UNIX文件，这种文件描述了一个物理设备，例如磁盘驱动器、磁带或者系统内存。

Diagnostic Attack（诊断攻击） 使用网络诊断工具来获得非法访问的一种黑客技术。

回拨（Dialback） 一种用于调制解调器用户的认证技术——中断通话并使用一个预定义的电话号码回拨用户。

Digital Signture（数字签名） 追加到一条消息后的数据块；使用数字签名，消息的接受方可以验证消息的内容和发送者。

Directory（目录） 一种包含其他文件或目录名字和位置的特殊文件。目录为组织信息提供了一个方便方法。

Directory Service（目录服务） 用来定位资源的DCE服务。

Distributed Database（分布式数据库） 数据和逻辑被分布到多个计算机系统上的数据库。这些系统是通过网络互相连接的。

Distributed Computing（分布式计算） 一种计算类型。在分布式计算中，计算控制、逻辑以及处理会在大量单独的计算机（通过网络互相连接）上进行。

Distributed File Service（分布式文件服务） DCE用来管理分布式文件的一种服务。

Distributed Systems（分布式系统） 通过一个网络进行通信并且共享资源和处理的多个计算机系统和应用程序。

DNS 域名系统（Domain name system），一个用来解析系统名字和IP地址的网络工具。

DSS 数字签名标准（digital sinature standard），美国政府提供的一种用于数字签名的标准。

Dual Homed Bastion Host（双穴桥头堡主机） 用来过滤网络流量的双LAN卡桥头堡主机，其中一块LAN卡连接到不可信网络上，而另一块连接到可信网络上。

durability（持久性） 系统长时间运行并且性能不受影响的能力。

EUID 有效用户ID，当前执行程序拥有者的用户ID。UNIX用户ID（UID）是登录时所用的用户ID。

Electronic Data Interchange（电子数据交换） 一个用于传递组织间业务事务的标准。

Electronic Commerce（电子商务） 通过电子媒介（例如因特网）进行的业务文档交换。

encapsulation（封装） 把数据和用于操作数据的程序打包在一起形成一个可标识的实体的技术，主要用在面向对象处理中。

encryption（加密） 通过一个算法处理和某种形式的保密密钥，把信息打乱成不可理解的格式；使用逆反过程，加密过的信息可以还原成可以理解的格式。

encryption Algorithm（加密算法） 用来把信息转化成不可理解的格式的算法，它需要使用一个密钥。

File access permission（文件访问权限） 用于确定是否授予对一个文件或者目录的访问权的一套标志。

File System（文件系统） 文件和目录在计算机磁盘、CD-ROM或者磁带上的组织。

Filtering Router（过滤路由器） 能够进行包过滤的路由器。

Firewall（防火墙） 对可信网络和不可信网络之间的流量进行限制的一个计算机系统或者网络设备。

FTP 文件传输协议（File transfer protocol），一种允许在系统间传输文件的TCP/IP服务。

gecos UNIX密码文件中的一个字段，用于存储用户信息。

GUEST Account（GUEST账号） NOS系统上为临时用户所用的账号，一般没有密码。

group（组） 多个用户所组成的一个逻辑联合体，这些用户的访问权限是建立在他们在组中的成员关系的基础上的。

groupware（群件） 该术语用来描述那些用来支持部门或者组织对共享信息需要的软件。

GUI 图形用户界面（Graphical user interface）。

GSSAPI 通用服务应用编程接口（Generic security service application programming interface），一个用于安全性编程接口的标准。

guideline（指导原则） 用于计算环境的最佳安全实践的建议。

heterogeneous（异构） 在计算机术语中，异构环境指的是由许多不同技术构成的一个混合环境。

High Level Privilege（高级特权） 只授予给系统的最可信用户（例如系统或者数据库管理员）的权限。

Home Directory（主目录） 用户登录后所进入的默认目录。

Host Equivalency（主机等效性） 描述如下情况的一种关系，一个系统信任另一个系统，并且不需要程序的用户在获得资源访问权前先认证自己。

HTML 超文本标记语言（Hypertext markup language），用来描述WWW文档组件的语言。

HTTP 超文本传输协议（Hypertext transfer protocol），客户机同WWW服务器之间使用的标准通信协议。

ICMP 因特网控制报文协议（Internet Control Message Protocol），用于监视和控制IP网络的IP协议。

identification（标识符） 确定计算机用户身份的一种方法（例如用户ID），一般是一个认证过程的结果。

Information Warfare（信息竞争） 一种竞争形式，其目的在于摧毁或者废除敌方的计算资源。

Integrity（完整性） 防止计算环境被未经授权修改的能力。

Integrity Check（完整性检查） 检查一个计算环境是否被未经授权修改的工具或者任务。

IPv6 因特网协议的新提议版本。比较现有的IP协议，它有了重大的安全改进。

IP Forwarding Attack（IP转发攻击） 一种试图把TCP/IP包发送到受保护网络中的黑客

技术。

ISQL 交互式SQL（Interactive SQL），允许用户使用标准SQL命令直接查询一个关系型数据库的工具。

JAVA Sun公司开发的一种独立于平台的编程语言。

LAN 本地网络。

LEAF Access（LEAF访问） 法律实施访问字段（law enforcement access field，LEAF）是加密芯片系统中的一个特殊部分。它允许法律实施机构访问加密过的数据——如果机构被授权了。

Leakage（泄漏） 公司防火墙或者路由器不小心把可信网络的信息泄漏给不可信网络。

LSA 本地安全管理中心（Local security authority）,运行在Windows NT系统上的一个安全子系统。

Location Based Authentication（基于位置的认证） 根据对请求认证设备所预期的准确位置进行的认证。

Login（登录） 获得对计算系统或者网络进行直接访问的过程。

Mandatory Access Control（强制访问控制） 一种安全控制，例如敏感性标签，用户不能关闭或者删除它。

Message Digest（消息摘要） 一种加密过的消息算法表示，它附加在消息后面以便进行完整性检查。

Middleware（中间件） 一个软件层，它在应用程序、数据以及操作系统之间提供一个公共接口和转化。

mount（挂接） 在UNIX中对一个远程或者本地文件系统获得访问的过程。

Mutual Authentication（互相认证） 一种认证形式，其中参与双方互相认证对方，通常通过一个可信的第三方。

NIS 网络信息服务（Network information service），Sun微系统公司开发的一种UNIX网络服务，它允许关键UNIX管理文件在系统间保持同步。

NOS 网络操作系统（Network operating system），对向LAN用户提供磁盘、目录和打印共享服务的网络环境的总称。

NFS 网络文件系统（Network file system），UNIX和DOS系统用来共享磁盘资源的一种工具，由Sun微系统公司开发。

nonrepudiation（认可） 对消息发送者和信息内容进行认证的能力。

NT Domain（NT域） 一组位于公共管理操作下的Windows NT系统。

NTFS NT文件系统（NT File System），Windows NT系统使用的固有文件系统。

OLTP 联机事务处理（On line transaction processing），事务发生时用于事务处理的系统或者过程。

One Time Password（一次性密码） 由一个特殊算法所生成的密码，永远不会重用。

OMG 对象管理组（Object Management Group），由100多个厂商组成的一个标准联盟，致力于解决面向对象环境的互操作性和可移植性。

Orange Book（桔皮书） 参见TCSEC。

OSF/DCE 开放软件基金会（Open Software Foundation）的分布式计算环境

(Distributed Computing Enviroment)。

owner(拥有者) 资源的拥有者通常是资源的建立者。

Packet Filtering(分组过滤) 根据流量类型、源IP地址和目标IP地址、端口或者其他信息，对TCP/IP流量进行限制。

Packet Monitoring(包监视) 主动监视网络上的流量。黑客使用这种方法来发现密码和其他敏感信息。

password(密码) 一个保密的单词、短语、数文混合字符串或者击键组合，用于认证。

Password Aging(密码时效) 一种用来强制用户改变密码的机制。

Password Cracking(密码破解) 一种发现密码的方法。把字典中的每一条都输入到一个密码算法中，并把结果同加密过的密码字符串逐一进行比较，如果二者匹配，那么密码就破解了。

Password Generator(密码生成器) 一个计算设备或者软件程序，可以产生惟一的和不可破解(臆测)的密码字符串。

Password Guessing(密码猜测) 一种用来获得对一个计算机或者网络设备进行未经授权访问的方法——逐一猜测用户账号的密码。

Password Policy(密码策略) 规定密码的组成、时效性以及其他方面的策略。

PATH Variable(PATH变量) 一个环境变量，用于告诉操作系统应该在什么目录下查找命令文件。

PEM 保密增强邮件(Privacy enhanced mail)。这是一个因特网标准草案，它定义了消息加密和认证过程，从而在因特网上提供了安全邮件功能。

PGP 良好保密(Pretty good privacy)。Philip Zimmerman在RSA加密算法的基础上开发的一个加密方案。

performance(性能) 计算环境及时执行所需操作的能力。

Physical Access(物理访问) 直接同计算机系统交互的能力(例如直接访问系统控制台、磁盘驱动器、CPU等)。

PIN 个人标识号(Personal Identification Number)。这是一个独立的数字，用做认证过程的一部分。

principal(负责者) 在DCE术语中，负责者是其重要性可以保证认证的任何实体。负责者可以是人、计算机、DCE单元或者应用程序。

principle(原则) 对那些为组织提供一个基本基础的价值、信念或者思想的陈述。

Private Key Encryption(私有密钥加密) 也称共享的保密密钥，指的是一种加密算法，在这种算法中，一个或多个实体(通常是一个用户和计算系统)共享一个密码(例如一个私有密钥)。这种方法主要用于认证，它也广泛用于保护敏感数据。

procedure(过程) 从计算安全策略者，过程是用来为一个特定的计算或者网络环境制定计算策略的实际方法。

process(进程) 一个环境，程序在其中执行。

Proxy Service(代理服务) 一个代表用户执行任务的因特网服务。使用代理服务通常要比允许用户直接访问因特网安全。

PUBLIC Account(PUBLIC账号) NOS系统上的一个账号，网络上的任何用户都可以

访问该账号。该账号一般没有密码。

Public Key Encryption（公开密钥加密） 一种加密算法。在这种算法中，生成两个在数学上有关联的密钥：其中，一个密钥是私有的，只有用户知道；另一个密钥是公开的，它可以被可信用户或者公众共享。这两个密钥可以用来加密和解密信息，并提供标识。

reboot（重新启动） 把一个计算机系统从运行状态——变成停止状态——再回到运行状态的过程。

Remote Access（远程访问） 通常指的是使用调制解调器在电话线路上同一台计算机或者网络进行通信的能力。

RAS 远程访问服务（Remote access service）——一个Windows NT服务，它通过调制解调器与用户通信。它具有用户认证和回拨功能。

Remote Execution Facility（远程执行工具） 一个允许命令远程执行但不需要用户登录的TCP/IP网络服务。

Remote Procedure Call（远程过程调用） 一种客户机-服务器通信方法。

Risk（风险） 在计算安全术语中，风险指的是一个安全暴露。根据上下文，它也可以指发生安全问题的可能性。

Risk Analysis（风险分析） 用常规话说，风险分析指的是在一个暴露（例如风险）的危害性与它所发生的可能性之间达成平衡。安全专业人员和审计人员也使用风险分析来指出对安全控制的主动测试。

Risk Assessment（风险评估） 也称风险分析。风险评估需要对控制的正确性和实用性进行分析。

Role（角色） 很多数据库允许根据任务功能指定访问，这种能力称做角色。

Role Based Security（基于角色的安全性） 一套在组织或者实体中定义的角色或者位置的安全属性。

root（根） UNIX系统上的一个具有高级特权的默认用户。Root也用来指最高级目录（根目录或者/）。

Routing Attack（路由攻击） 一种使用IP源路由选项把包路由到一个目标地址（包在这里可以操纵）的黑客技术。

Rule Based Security（基于规则的安全性） 一套安全属性<Glen>。

SATAN 用于审计网络的安全管理工具（Security administrators tool for auditing network）。这是由Dan Farmer开发的一个捐赠的安全监视器，它可以监测TCP/IP网络服务的脆弱性。

SDLC 同步数据链路控制（Synchronous data link control），一个用于IBM系统网络体系结构的通信协议。

S-HTTP 安全超文本传输协议（Security Hybertext transfer protocol），对HTTP协议的一个扩展，它提供了加密和验证功能。

Screened Subnet（屏蔽子网） 受过滤路由器或者其他网络设备保护的一个网络。

Secure RFC（安全RPC） SUN微系统公司开发的一种基于RPC的通信方法，它具有安全认证和数据保护机制。

SSL 安全套接字层（secure socket layer），客户机和服务器之间用于认证和加密的一种

协议。

Security Access Mechanism（安全访问机制，SAM） 用于Windows NT域的安全管理中心。

SDLC 系统开发生命周期（Systerm Development Life Cycle）。一种用于开发计算应用的方法，由6个阶段组成。

Security Advisories（安全忠告） 通常通过电子邮件发布的安全信息，其作用是把安全问题及其相关解决通告给计算群体。安全忠告可以从CERT处获得，也可以直接从许多厂商处获得。

Security Criteria（安全标准） 一套被认可的安全属性集合，它构成了用于测量和分析安全功能的一套规范。

Security Descriptor（安全描述符） Windows NT系统上的每个资源都有一个安全描述符，该描述符包含了有关拥有者的信息和访问控制单。

Security Policy（安全策略） 一套经过管理部门批准的正式指导，它为组织提供了对计算安全之各个方面的指导。

Security Service（安全服务） 向DCE提供安全服务，包括认证和密码管理。

Security Token（安全令牌） 一个包含安全信息的设备或者文件，通常受到加密保护。

Self Assessment（自评估） 系统管理员、用户部门或者任何非独立实体进行的一次审计。自评估通常是在真正的审计开始之前进行。

Sensitive Information（敏感信息） 如果一个信息被未经授权的泄漏，它就会导致一次重大的财政损失、困难或者带来其他负面效应，那么该信息就是敏感信息。

Sensitivity Labels（敏感性标签） 把分类标题（例如凭证或者保密排行）应用到文件、目录以及其他计算资源上的一种方法。

server（服务器） 当其他系统或者客户机发来请求时执行服务的计算系统。

SET 安全电子事务（secure electronic transaction），一个用于电子商务事务的安全通信和执行的消息标准。

Set User ID Bit（设置用户ID位） 在文件的拥有者授权而不是调用者的授权下执行的一个可执行的UNIX文件（或者程序）。

Sequencing Attack（序列攻击） 一种黑客技术（也称做欺骗攻击——hijacking），它使用TCP/IP序号的知识截获已建立的TCP/IP通信中的流量，并在其中插入伪造流量。

Single Sign-on（单一登录） 一个用户一次并且也只有一次执行访问全部所需网络和计算资源的认证过程的能力。

Single-User Mode（单用户模式） 一种特殊的UNIX操作系统级别，主要用于诊断。在许多UNIX版本中，使用该模式可以无限制地访问系统。

Smart Card（智能卡） 内嵌有用来存储数据或者执行程序代码的微型芯片的卡。

SMTP 简单邮件传输协议（Simple Mail Transport Protocol），一个用于电子邮件应用程序的协议，在使用UNIX操作系统的系统上非常流行。

SNA 系统网络体系结构（System network architecture），IBM开发的一种连网体系结构。

SNMP 简单网络管理协议（simple network management protocol），一个用来管理网络设备和计算系统的标准协议。

socket（套接字） IP地址和端口号的结合体。

spoof（欺骗程序） 伪装成一个合法程序——但却进行非法行为的程序，也称做特洛伊木马。

spoofing（欺骗） 一种黑客技术，使用一个伪造的网络地址来获得对网络或者系统的非法访问权。

SQL 结构化查询语言（Structured query language），用于管理和报告数据库信息和结构的一套标准命令。

Stored Procedure（存储过程） 一套预定义的SQL语句。

subnet（子网） 一个LAN段。

superuser（超级用户） UNIX系统管理员。该用户的真实或有效用户ID为0，并且具有高级访问特权。

SUPERVISIOR NOS系统上的一个特殊账号，它提供了一套完整的特权。

system（系统） 一台计算机。

System Classification（系统分类） 根据系统所处理的或者所保存的信息组织的安全性，对系统进行标类。一个工资应用的系统可以同时划分为关键型和机密型。

TCP 传输控制协议（Transport control protocol）。TCP是一个高级IP协议，它具有序号和其他传输完整性控制。

TCP/IP 对基于因特网协议（IP）的连网协议的服务族的统称。

TCSEC 可信计算机系统评估标准（Trusted computer system evaluaiton criteria）。TCSEC是美国国防部为评估操作系统中的计算安全性而开发的一种方法，它也称做桔皮书。该系列的其他书还包括用于用于审计的褐皮书和用于密码管理的绿皮书。

Technology Envy（技术嫉妒） 对成为技术领导的渴望。

TELNET 一个标准的因特网服务，它允许用户使用终端并通过网络访问一台远程计算机。

TFTP 普通文件传输协议（trivial file transfer protocol），一个用于下载信息的文件传输程序。

Tiger Team（飞虎队） 安全事故响应小组。

Timing Service（定时服务） 用来在分布式系统之间同步时钟的一个DCE服务。

Token Based Authentication（基于令牌的认证） 如果用户具有一个惟一的物理卡或者设备（这是认证过程的一个组件），那么用户就通过认证。

TP Monitor（TP监视器） 一个程序——当事务提供给该程序的时候，该程序就控制和执行事务。

transaction（事务） 一套操作集合，这些操作组成了一个完整的工作单元。该工作单元的执行必须要满足ACID事务属性。

trigger（触发器） 由一个预定义的事件所触发的数据库存储过程。

tripwire Gene Kim和Gene Spafford开发的一个捐赠软件工具，它检查操作系统和应用程序数据文件和程序中的未经授权修改。

Trojan horse（特洛伊木马） 一个看起来工作正常，但却会进行未经授权动作的计算机程序。

trust（信任） 安全性、可用性以及性能的综合体，即在保证完整性的情况下执行程序、保持机密信息的保密性、连续运行以提供所需功能。

Two-Phase Commit（两阶段提交） 当所有必需的过程都成功完成以后，才提交数据库更新。

UDP 用户数据报协议（user datagram protocol）。UDP是一个更高层IP协议，它只有有限的完整性检查。

URL 统一资源定位符（Uniform resource locator），标识可在WWW上访问的文档、图像或者其他存储资源的一个惟一地址。

User ID（用户ID） 用来标识计算机用户的指定名或者缩写。

User Profile（用户轮廓） 包含关于用户登录和安全信息的文件。

UUCP UNIX到UNIX拷贝，一个允许UNIX系统间通信的UNIX连网服务。

virus（病毒） 一种感染计算机的特洛伊木马程序，它们以多种方式让计算机失去完整性。

VPN 虚拟专用网（Vitual Private Network），一种对可信网络之间的流量进行加密和保护的技术。

VTAM 虚拟远程通信接入方法（Vitual telecommunication access method），一个用来控制网络的IBM产品。

Vulnerability Test（脆弱性测试） 对安全控制的主动测试，意在发现安全脆弱性和暴露问题

Web Browser（浏览器） 一个用于在WWW上访问文档的软件产品。

X.400 国际标准化组织维护的一个标准，用于消息存储和转发。

X.509 Certificate（X.509证书） 符合X.509认证框架标准的证书。

XBSS X/Open基准安全服务（baseline security service），X/Open组织定义的一套安全标准。

X/Windows UNIX所使用的窗口系统。