第六章 访问控制

互联网络的蓬勃发展，为信息资源的共享提供了更加完善的手段，企业在信息资源共享的同时也要阻止非授权用户对企业敏感信息的访问。访问控制的目的是为了保护企业在信息系统中存储和处理的信息的安全。

本章主要涉及以下几个方面：访问控制的模型；访问控制的策略；访问控制的的实现；
安全级别与访问控制；访问控制与授权；访问控制与审计。

6.1 访问控制的模型

访问控制模型是一种从访问控制的角度出发，描述安全系统，建立安全模型的方法。

访问控制是指主体依据某些控制策略或权限对客体本身或是其资源进行的不同授权访问。访问控制包括三个要素，即：主体、客体和控制策略。

主体（Subject）：是指一个提出请求或要求的实体，是动作的发起者，但不一定是动作的执行者。主体可以是用户或其它任何代理用户行为的实体（例如进程、作业和程序）。我们这里规定实体（Entity）表示一个计算机资源（物理设备、数据文件、内存或进程）或一个合法用户。

主体（Subject）：是可以对其它实体施加动作的主动实体，简记为S。有时我们也称为用户（User）或访问者（被授权使用计算机的人员），记为U。主体的含义是广泛的，可以是用户所在的组织（以后我们称为用户组）、用户本身，也可是用户使用的计算机终端、卡机、手持终端（无线）等，甚至可以是应用服务程序程序或进程。

客体（Object）：是接受其他实体访问的被动实体, 简记为O。客体的概念也很广泛，凡是可以被操作的信息、资源、对象都可以认为是客体。在信息社会中，客体可以是信息、文件、记录等的集合体，也可以是网路上的硬件设施，无线通信中的终端，甚至一个客体可以包含另外一个客体。

控制策略：是主体对客体的操作行为集和约束条件集, 简记为KS。简单讲，控制策略是主体对客体的访问规则集，这个规则集直接定义了主体对可以的作用行为和客体对主体的条件约束。访问策略体现了一种授权行为，也就是客体对主体的权限允许，这种允许不超越规则集，由其给出。

访问控制系统三个要素之间的行为关系见图6.1.1，可以使用三元组（S，O，P）来表示，其中S表示主体，O表示客体，P表示许可。当主体S提出一系列正常的请求信息I1，…，In，通过信息系统的入口到达控制规则集KS监视的监控器，由KS判断是否允许或拒绝这次请求，因此这种情况下，必须先要确认是合法的主体，而不是假冒的欺骗者，也就是对主体进行认证。主体通过验证，才能访问客体，但并不保证其有权限可以对客体进行操作。客体对主体的具体约束由访问控制表来控制实现，对主体的验证一般会鉴别用户的标识和用户密码。用户标识（UID：User Identification）是一个用来鉴别用户身份的字符串，每个用户有且只能有唯一的一个用户标识，以便与其他用户区别。当一个用户注册进入系统时，他必须提供其用户标识，然后系统执行一个可靠的审查来确信当前用户是对应用户标识的那个用户。

多级安全信息系统：由于用户的访问涉及到访问的权限控制规则集合，对于上图6.1.1中将敏感信息与通常资源分开隔离的系统，我们称之为多级安全信息系统。多级安全信息系统的实例见Bell-LaPadula模型，分类见本章第四节。多级安全系统必然要将信息资源按照安全属性分级考虑，安全类别有两种类型，一种是有层次的安全级别（Hierarchical Classification），分为TS，S，C，RS，

U五级：绝密级别（Top Secret），秘密级别（Secret），机密级别（Confidential），限制级别（Restricted）和无级别级（Unclassified）；另一种是无层次的安全级别，不对主体和客体按照安全类别分类，只是给出客体接受访问时可以使用的规则和管理者。

访问控制的实现首先要考虑对合法用户进行验证，然后是对控制策略的选用与管理，最后要对没有非法用户或是越权操作进行管理。所以，访问控制包括认证、控制策略实现和审计三方面的内容：
1、 认证：主体对客体的识别认证和客体对主体检验认证。主体和客体的认证关系是相互的，当一个主体受到另外一个客体的访问时，这个主体也就变成了客体。一个实体可以在某一时刻是主体，而在另一时刻是客体，这取决于当前实体的功能是动作的执行者还是动作的被执行者。

2、 控制策略的具体实现：如何设定规则集合从而确保正常用户对信息资源的合法使用，既要防止非法用户，也要考虑敏感资源的泄漏，对于合法用户而言，更不能越权行使控制策略所赋予其权利以外的功能。

3、 审计：审计的重要意义在于，比如客体的管理者即管理员有操作赋予权，他有可能滥用这一权利，这是无法在策略中加以约束的。必须对这些行为进行记录，从而达到威慑和保证访问控制正常实现的目的。

访问控制安全模型一般包括主体、客体，以及为识别和验证这些实体的子系统和控制实体间访问的参考监视器。由于网络传输的需要，访问控制的研究方发展很快，有许多访问控制模型被提出来。建立规范的访问控制模型，是实现严格访问控制策略所必须的。20世纪70年代，Harrison， Ruzzo和Ullman提出了HRU模型。接着，Jones等人在1976年提出了Take-Grant模型。随后，1985年美国军方提出可信计算机系统评估准则TCSEC，其中描述了两种著名的访问控制策略：自主访问控制模型（DAC）和强制访问控制模型（MAC）。基于角色的访问控制（RBAC）由Ferraiolo和Kuhn在1992年提出的。考虑到网络安全和传输流，又提出了基于对象和基于任务的访问控制。

本节在探讨现有信息系统安全模型的基础上，主要分析信息流模型、Bell-LaPadula（BLP）模型和Biba模型等访问控制模型的优缺点，并针对信息安全的现实要求，对基于角色、基于对象、基于任务的模型做阐述。

6.1.1自主访问控制模型

自主访问控制模型（DAC Model，Discretionary Access Control Model）是根据自主访问控制策略建立的一种模型，允许合法用户以用户或用户组的身份访问策略规定的客体，同时阻止非授权用户访问客体，某些用户还可以自主地把自己所拥有的客体的访问权限授予其它用户。自主访问控制又称为任意访问控制。LINUX，UNIX、WindowsNT或是SERVER版本的操作系统都提供自主访问控制的功能。在实现上，首先要对用户的身份进行鉴别，然后就可以按照访问控制列表所赋予用户的权限允许和限制用户使用客体的资源。主体控制权限的修改通常由特权用户或是特权用户（管理员）组实现。

任意访问控制对用户提供的这种灵活的数据访问方式，使得DAC广泛应用在商业和工业环境中；由于用户可以任意传递权限，那么，没有访问文件File1权限的用户A就能够从有访问权限的用户B那里得到访问权限或是直接获得文件File1；因此，DAC模型提供的安全防护还是相对比较低的，不能给系统提供充分的数据保护。

自主访问控制模型的特点是授权的实施主体（1、可以授权的主体；2、管理授权的客体；3、授权组）自主负责赋予和回收其他主体对客体资源的访问权限。DAC模型一般采用访问控制矩阵和访问

控制列表来存放不同主体的访问控制信息，从而达到对主体访问权限的限制目的。

6.1.2强制访问控制模型

强制访问控制模型（MAC Model：Mandatory Access Control Model）最开始为了实现比DAC更为严格的访问控制策略，美国政府和军方开发了各种各样的控制模型，这些方案或模型都有比较完善的和详尽的定义。随后，逐渐形成强制访问的模型，并得到广泛的商业关注和应用。在DAC访问控制中，用户和客体资源都被赋予一定的安全级别，用户不能改变自身和客体的安全级别，只有管理员才能够确定用户和组的访问权限。和DAC模型不同的是，MAC是一种多级访问控制策略，它的主要特点是系统对访问主体和受控对象实行强制访问控制，系统事先给访问主体和受控对象分配不同的安全级别属性，在实施访问控制时，系统先对访问主体和受控对象的安全级别属性进行比较，再决定访问主体能否访问该受控对象。MAC对访问主体和受控对象标识两个安全标记：一个是具有偏序关系的安全等级标记；另一个是非等级分类标记。安全等级的层次我们在上节做过阐述，主体和客体在分属不同的安全类别时，都属于一个固定的安全类别SC，SC就构成一个偏序关系（比如TS表示绝密级，就比密级S要高）。当主体s的安全类别为TS，而客体o的安全类别为S时，用偏序关系可以表述为SC(s)≥SC(o)。考虑到偏序关系，主体对客体的访问主要有四种方式：
（1）向下读（rd，read down）：主体安全级别高于客体信息资源的安全级别时允许查阅的读操作；
（2）向上读（ru，read up）：主体安全级别低于客体信息资源的安全级别时允许的读操作；
（3）向下写（wd，write down）：主体安全级别高于客体信息资源的安全级别时允许执行的动作或是写操作；
（4）向上写（wu，write up）：主体安全级别低于客体信息资源的安全级别时允许执行的动作或是写操作。

由于MAC通过分级的安全标签实现了信息的单向流通，因此它一直被军方采用，其中最著名的是Bell-LaPadula模型和Biba模型：Bell-LaPadula模型具有只允许向下读、向上写的特点，可以有效地防止机密信息向下级泄露；Biba模型则具有不允许向下读、向上写的特点，可以有效地保护数据的完整性。

下面我们对MAC模型中的几种主要模型：Lattice模型，Bell-LaPadula模型（BLP Model）和Biba模型（Biba Model）做简单的阐述：
1．Lattice模型
在Lattices模型中，每个资源和用户都服从于一个安全类别。这些安全类别我们称为安全级别，也就是我们在本章开始所描述的五个安全级别，TS，S，C，R，U。在整个安全模型中，信息资源对应一个安全类别，用户所对应的安全级别必须比可以使用的客体资源高才能进行访问。Lattices模型是实现安全分级的系统，这种方案非常适用于需要对信息资源进行明显分类的系统。

2．Bell-LaPadula模型
BLP[Bell and LaPadula，1976]模型是典型的信息保密性多级安全模型，主要应用于军事系统。Bell-LaPadula模型通常是处理多级安全信息系统的设计基础，客体在处理绝密级数据和秘密级数据时，要防止处理绝密级数据的程序把信息泄露给处理秘密级数据的程序。BLP模型的出发点是维护系统的保密性，有效地防止信息泄露，这与我们后面讲的维护信息系统数据完整性的Biba模型正好相反。

Lattice模型没有考虑特洛伊木马等不安全因素的潜在威胁，这样，低级安全用户有可能复制和拷贝比较敏感的信息。在军方术语中，特洛伊木马的最大作用是降低整个系统的安全级别。考虑到这种攻击行为，Bell和LaPadula设计了一种模型抵抗这种攻击，我们称为Bell-LaPadula模型。Bell-LaPadula模型可以有效防止低级用户和进程访问安全级别比他们高的信息资源。此外，安全级别高

的用户和进程也不能向比他安全级别低的用户和进程写入数据。上述Bell-LaPadula模型建立的访问控制原则可以用以下两点简单表示：（1）无上读（2）无下写。

BLP模型的安全策略包括强制访问控制和自主访问控制两部分：强制访问控制中的安全特性要求对给定安全级别的主体，仅被允许对同一安全级别和较低安全级别上的客体进行"读"；对给定安全级别上的主体，仅被允许向相同安全级别或较高安全级别上的客体进行"写"；任意访问控制允许用户自行定义是否让个人或组织存取数据。Bell-LaPadula模型用偏序关系可以表示为：（1）rd，当且仅当SC(s)≥SC(o)，允许读操作（2）wu，当且仅当SC(s) ≤SC(o)，允许写操作。虽然BLP模型"只能从下读、向上写"的规则忽略了完整性的重要安全指标，使非法、越权篡改成为可能。

BLP模型为通用的计算机系统定义了安全性属性，即以一组规则表示什么是一个安全的系统，尽管这种基于规则的模型比较容易实现，但是它不能更一般地以语义的形式阐明安全性的含义，因此，这种模型不能解释主-客体框架以外的安全性问题。例如，在一种远程读的情况下，一个高安全级主体向一个低安全级客体发出远程读请求，这种分布式读请求可以被看作是从高安全级向低安全级的一个消息传递，也就是"向下写"。另一个例子是如何处理可信主体的问题，可信主体可以是管理员或是提供关键服务的进程，像设备驱动程序和存储管理功能模块，这些可信主体若不违背BLP模型的规则就不能正常执行它们的任务，而BLP模型对这些可信主体可能引起的泄露危机没有任何处理和避免的方法。

3．Biba模型
Biba模型[Biba,1977]在研究BLP模型的特性时发现，BLP模型只解决了信息的保密问题，其在完整性定义存在方面有一定缺陷。BLP模型没有采取有效的措施来制约对信息的非授权修改，因此使非法、越权篡改成为可能。考虑到上述因素，Biba模型模仿BLP模型的信息保密性级别，定义了信息完整性级别，在信息流向的定义方面不允许从级别低的进程到级别高的进程，也就是说用户只能向比自己安全级别低的客体写入信息，从而防止非法用户创建安全级别高的客体信息，避免越权、篡改等行为的产生。Biba模型可同时针对有层次的安全级别和无层次的安全种类。

Biba模型的两个主要特征是
（1）禁止向上"�"，这样使得完整性级别高的文件是一定由完整性高的进程所产生的，从而保证了完整性级别高的文件不会被完整性低的文件或完整性低的进程中的信息所覆盖。

（2）Biba模型没有下"读"。

Biba模型用偏序关系可以表示为：
（1）ru，当且仅当SC(s) ≤SC(o)，允许读操作
（2）wd，当且仅当SC(s) ≥SC(o)，允许写操作。

Biba模型是和BLP模型相对立的模型，Biba模型改正了被BLP模型所忽略的信息完整性问题，但在一定程度上却忽视了保密性。

MAC访问控制模型和DAC访问控制模型属于传统的访问控制模型，对这两种模型研究的也比较充分。在实现上，MAC和DAC通常为每个用户赋予对客体的访问权限规则集，考虑到管理的方便，在这一过程中还经常将具有相同职能的用户聚为组，然后再为每个组分配许可权。用户自主地把自己所拥有的客体的访问权限授予其它用户的这种做法，其优点是显而易见的，但是如果企业的组织结构或是系统的安全需求出于变化的过程中时，那么就需要进行大量繁琐的授权变动，系统管理员的工作将变得非常繁重，更主要的是容易发生错误造成一些意想不到的安全漏洞。考虑到上述因素，我们引入新的机制加以解决。首先要介绍一下角色的概念，角色（Role）是指一个可以完成一定事务的命名组，不同的角色通过不同的事务来执行各自的功能。事务（Transaction）是指一个完成一定功能的过程，可以是一个程序或程序的一部分。角色是代表具有某种能力的人或是某些属性的人

的一类抽象，角色和组的主要区别在于：用户属于组是相对固定的，而用户能被指派到哪些角色则受时间、地点、事件等诸多因素影响。角色比组的抽象级别要高，角色和组的关系可以这样考虑，作为饰演的角色，我是一名学生，我就只能享有学生的权限（区别于老师），但是我又处于某个班级中，就同时只能享有本"组"组员的权限。

6.1.3基于角色的访问控制模型

基于角色的访问控制模型（RBAC Model，Role-based Access Model）：RBAC模型的基本思想是将访问许可权分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的访问许可权。这是因为在很多实际应用中，用户并不是可以访问的客体信息资源的所有者（这些信息属于企业或公司），这样的话，访问控制应该基于员工的职务而不是基于员工在哪个组或是谁信息的所有者，即访问控制是由各个用户在部门中所担任的角色来确定的，例如，一个学校可以有教工、老师、学生和其他管理人员等角色。

RBAC从控制主体的角度出发，根据管理中相对稳定的职权和责任来划分角色，将访问权限与角色相联系，这点与传统的MAC和DAC将权限直接授予用户的方式不同；通过给用户分配合适的角色，让用户与访问权限相联系。角色成为访问控制中访问主体和受控对象之间的一座桥梁。

角色可以看作是一组操作的集合，不同的角色具有不同的操作集，这些操作集由系统管理员分配给角色。在下面的实例中，我们假设Tch1，Tch2，Tch3……Tchi是对应的教师，Stud1，Stud 2，Stud3 …Studj是相应的学生，Mng1，Mng 2，Mng 3…Mngk是教务处管理人员，那么老师的权限为TchMN={查询成绩、上传所教课程的成绩}；学生的权限为Stud MN={查询成绩、反映意见}；教务管理人员的权限为MngMN={查询、修改成绩、打印成绩清单}。那么，依据角色的不同，每个主体只能执行自己所制定的访问功能。用户在一定的部门中具有一定的角色，其所执行的操作与其所扮演的角色的职能相匹配，这正是基于角色的访问控制（RBAC）的根本特征，即：依据RBAC策略，系统定义了各种角色，每种角色可以完成一定的职能，不同的用户根据其职能和责任被赋予相应的角色，一旦某个用户成为某角色的成员，则此用户可以完成该角色所具有的职能。

系统管理员负责授予用户各种角色的成员资格或撤消某用户具有的某个角色。例如学校新进一名教师Tchx，那么系统管理员只需将Tchx添加到教师这一角色的成员中即可，而无需对访问控制列表做改动。同一个用户可以是多个角色的成员，即同一个用户可以扮演多种角色，比如一个用户可以是老师，同时也可以作为进修的学生。同样，一个角色可以拥有多个用户成员，这与现实是一致的，一个人可以在同一部门中担任多种职务，而且担任相同职务的可能不止一人。因此RBAC提供了一种描述用户和权限之间的多对多关系，角色可以划分成不同的等级，通过角色等级关系来反映一个组织的职权和责任关系，这种关系具有反身性、传递性和非对称性特点，通过继承行为形成了一个偏序关系，比如MngMN>TchMN>Stud MN。RBAC中通常定义不同的约束规则来对模型中的各种关系进行限制，最基本的约束是"相互排斥"约束和"基本限制"约束，分别规定了模型中的互斥角色和一个角色可被分配的最大用户数。RBAC中引进了角色的概念，用角色表示访问主体具有的职权和责任，灵活地表达和实现了企业的安全策略，使系统权限管理在企业的组织视图这个较高的抽象集上进行，从而简化了权限设置的管理，从这个角度看，RBAC很好地解决了企业管理信息系统中用户数量多、变动频繁的问题。

相比较而言， RBAC是实施面向企业的安全策略的一种有效的访问控制方式，其具有灵活性、方便性和安全性的特点，目前在大型数据库系统的权限管理中得到普遍应用。角色由系统管理员定义，角色成员的增减也只能由系统管理员来执行，即只有系统管理员有权定义和分配角色。用户与客体无直接联系，他只有通过角色才享有该角色所对应的权限，从而访问相应的客体。因此用户不能自主地将访问权限授给别的用户，这是RBAC与DAC的根本区别所在。RBAC与MAC的区别在于：

MAC是基于多级安全需求的，而RBAC则不是。

6.1.4基于任务的访问控制模型

上述几个访问控制模型都是从系统的角度出发去保护资源（控制环境是静态的），在进行权限的控制时没有考虑执行的上下文环境。数据库、网络和分布式计算的发展，组织任务进一步自动化，与服务相关的信息进一步计算机化，这促使人们将安全问题方面的注意力从独立的计算机系统中静态的主体和客体保护，转移到随着任务的执行而进行动态授权的保护上。此外，上述访问控制模型不能记录主体对客体权限的使用，权限没有时间限制，只要主体拥有对客体的访问权限，主体就可以无数次地执行该权限。考虑到上述原因，我们引入工作流的概念加以阐述。工作流是为完成某一目标而由多个相关的任务（活动）构成的业务流程。工作流所关注的问题是处理过程的自动化，对人和其他资源进行协调管理，从而完成某项工作。当数据在工作流中流动时，执行操作的用户在改变，用户的权限也在改变，这与数据处理的上下文环境相关。传统的DAC和MAC访问控制技术，则无法予以实现，我们讲过的RBAC模型，也需要频繁地更换角色，且不适合工作流程的运转。这就迫使我们必须考虑新的模型机制，也就是基于任务的访问控制模型。

基于任务的访问控制模型（TBAC Model，Task-based Access Control Model）是从应用和企业层角度来解决安全问题，以面向任务的观点，从任务（活动）的角度来建立安全模型和实现安全机制，在任务处理的过程中提供动态实时的安全管理。

在TBAC中，对象的访问权限控制并不是静止不变的，而是随着执行任务的上下文环境发生变化。TBAC首要考虑的是在工作流的环境中对信息的保护问题：在工作流环境中，数据的处理与上一次的处理相关联，相应的访问控制也如此，因而TBAC是一种上下文相关的访问控制模型。其次，TBAC不仅能对不同工作流实行不同的访问控制策略，而且还能对同一工作流的不同任务实例实行不同的访问控制策略。从这个意义上说，TBAC是基于任务的，这也表明，TBAC是一种基于实例（instance-based）的访问控制模型。

TBAC模型由工作流、授权结构体、受托人集、许可集四部分组成。

任务（task）是工作流程中的一个逻辑单元，是一个可区分的动作，与多个用户相关，也可能包括几个子任务。授权结构体是任务在计算机中进行控制的一个实例。任务中的子任务，对应于授权结构体中的授权步。

授权结构体（authorization unit）：是由一个或多个授权步组成的结构体，它们在逻辑上是联系在一起的。授权结构体分为一般授权结构体和原子授权结构体。一般授权结构体内的授权步依次执行，原子授权结构体内部的每个授权步紧密联系，其中任何一个授权步失败都会导致整个结构体的失败。

授权步（authorization step）表示一个原始授权处理步，是指在一个工作流程中对处理对象的一次处理过程。授权步是访问控制所能控制的最小单元，由受托人集（trustee-set）和多个许可集（permissions set）组成。

受托人集是可被授予执行授权步的用户的集合，许可集则是受托集的成员被授予授权步时拥有的访问许可。当授权步初始化以后，一个来自受托人集中的成员将被授予授权步，我们称这个受托人为授权步的执行委托者，该受托人执行授权步过程中所需许可的集合称为执行者许可集。授权步之间或授权结构体之间的相互关系称为依赖（dependency），依赖反映了基于任务的访问控制的原则。授权步的状态变化一般自我管理，依据执行的条件而自动变迁状态，但有时也可以由管理员进行调配。

一个工作流的业务流程由多个任务构成。而一个任务对应于一个授权结构体，每个授权结构体由特定的授权步组成。授权结构体之间以及授权步之间通过依赖关系联系在一起。在TBAC中，一个授权步的处理可以决定后续授权步对处理对象的操作许可，上述许可集合称为激活许可集。执行者许可集和激活许可集一起称为授权步的保护态。

TBAC模型一般用五元组（S，O，P，L，AS）来表示，其中S表示主体，O表示客体，P表示许可，L表示生命期（lifecycle），AS表示授权步。由于任务都是有时效性的，所以在基于任务的访问控制中，用户对于授予他的权限的使用也是有时效性的。因此，若P是授权步AS所激活的权限，那么L则是授权步AS的存活期限。在授权步AS被激活之前，它的保护态是无效的，其中包含的许可不可使用。当授权步AS被触发时，它的委托执行者开始拥有执行者许可集中的权限，同时它的生命期开始倒记时。在生命期期间，五元组（S，O，P，L，AS）有效。生命期终止时，五元组（S，O，P，L，AS）无效，委托执行者所拥有的权限被回收。

TBAC的访问政策及其内部组件关系一般由系统管理员直接配置。通过授权步的动态权限管理，TBAC支持最小特权原则和最小泄漏原则，在执行任务时只给用户分配所需的权限，未执行任务或任务终止后用户不再拥有所分配的权限；而且在执行任务过程中，当某一权限不再使用时，授权步自动将该权限回收；另外，对于敏感的任务需要不同的用户执行，这可通过授权步之间的分权依赖实现。

TBAC从工作流中的任务角度建模，可以依据任务和任务状态的不同，对权限进行动态管理。因此，TBAC非常适合分布式计算和多点访问控制的信息处理控制以及在工作流、分布式处理和事务管理系统中的决策制定。

6.1.5基于对象的访问控制模型

基于对象的访问控制（OBAC Model：Object-based Access Control Model）：DAC或MAC模型的主要任务都是对系统中的访问主体和受控对象进行一维的权限管理，当用户数量多、处理的信息数据量巨大时，用户权限的管理任务将变得十分繁重，并且用户权限难以维护，这就降低了系统的安全性和可靠性。对于海量的数据和差异较大的数据类型，需要用专门的系统和专门的人员加以处理，要是采用RBAC模型的话，安全管理员除了维护用户和角色的关联关系外，还需要将庞大的信息资源访问权限赋予有限个角色。当信息资源的种类增加或减少时，安全管理员必须更新所有角色的访问权限设置，而且，如果受控对象的属性发生变化，同时需要将受控对象不同属性的数据分配给不同的访问主体处理时，安全管理员将不得不增加新的角色，并且还必须更新原来所有角色的访问权限设置以及访问主体的角色分配设置，这样的访问控制需求变化往往是不可预知的，造成访问控制管理的难度和工作量巨大。在这种情况下，有必要引入基于受控对象的访问控制模型。

控制策略和控制规则是OBAC访问控制系统的核心所在，在基于受控对象的访问控制模型中，将访问控制列表与受控对象或受控对象的属性相关联，并将访问控制选项设计成为用户、组或角色及其对应权限的集合；同时允许对策略和规则进行重用、继承和派生操作。这样，不仅可以对受控对象本身进行访问控制，受控对象的属性也可以进行访问控制，而且派生对象可以继承父对象的访问控制设置，这对于信息量巨大、信息内容更新变化频繁的管理信息系统非常有益，可以减轻由于信息资源的派生、演化和重组等带来的分配、设定角色权限等的工作量。

OBAC从信息系统的数据差异变化和用户需求出发，有效地解决了信息数据量大、数据种类繁多、数据更新变化频繁的大型管理信息系统的安全管理。OBAC从受控对象的角度出发，将访问主体的访问权限直接与受控对象相关联，一方面定义对象的访问控制列表，增、删、修改访问控制项易于操作，另一方面，当受控对象的属性发生改变，或者受控对象发生继承和派生行为时，无须更新访

问主体的权限，只需要修改受控对象的相应访问控制项即可，从而减少了访问主体的权限管理，降低了授权数据管理的复杂性。

6.1.6信息流模型

从安全模型所控制的对象来看，一般有两种不同的方法来建立安全模型：一种是信息流模型；另一种是访问控制模型。

信息流模型主要着眼于对客体之间的信息传输过程的控制，通过对信息流向的分析可以发现系统中存在的隐蔽通道，并设法予以堵塞。信息流是信息根据某种因果关系的流动，信息流总是从旧状态的变量流向新状态的变量。信息流模型的出发点是彻底切断系统中信息流的隐蔽通道，防止对信息的窃取。隐蔽通道就是指系统中非正常使用的、不受强制访问控制正规保护的通信方式。隐蔽通道的存在显然危及系统敏感信息的保护。信息流模型需要遵守的安全规则是：在系统状态转换时，信息流只能从访问级别低的状态流向访问级别高的状态。信息流模型实现的关键在于对系统的描述，即对模型进行彻底的信息流分析，找出所有的信息流，并根据信息流安全规则判断其是否为异常流。若是就反复修改系统的描述或模型，直到所有的信息流都不是异常流为止。信息流模型是一种基于事件或踪迹的模型，其焦点是系统用户可见的行为。现有的信息流模型无法直接指出哪种内部信息流是被允许的，哪种是不被允许的，因此在实际系统中的实现和验证中没有太多的帮助和指导。

6.2 访问控制策略

6.2.1安全策略

安全策略建立的需要和目的：安全的领域非常广泛繁杂，构建一个可以抵御风险的安全框架涉及很多细节。就算是最简单的安全需求，也可能会涉及到密码学、代码重用等实际问题。做一个相当完备的安全分析不得不需要专业人员给出许许多多不同的专业细节和计算环境，这通常会使专业的框架师也望而生畏。如果我们能够提供一种恰当的、符合安全需求的整体思路，就会使这个问题容易的多，也更加有明确的前进方向。能够提供这种帮助的就是安全策略。一个恰当的安全策略总会把自己关注的核心集中到最高决策层认为必须值得注意的那些方面。概括地说，一种安全策略实质上表明：当设计所涉及的那个系统在进行操作时，必须明确在安全领域的范围内，什么操作是明确允许的，什么操作是一般默认允许的，什么操作是明确不允许的，什么操作是默认不允许的。我们不要求安全策略作出具体的措施规定以及确切说明通过何种方式能够够达到预期的结果，但是应该向安全构架的实际搭造者们指出在当前的前提下，什么因素和风险才是最重要的。就这个意义而言，建立安全策略是实现安全的最首要的工作，也是实现安全技术管理与规范的第一步。

安全策略的具体含义和实现：安全策略的前提是具有一般性和普遍性，如何能使安全策略的这种普遍性和我们所要分析的实际问题的特殊性相结合，即，使安全策略与当前的具体应用紧密结合是我们面临的最主要的问题。控制策略的制定是一个按照安全需求、依照实例不断精确细化的求解过程。安全策略的制订者总是试图在安全设计的每个设计阶段分别设计和考虑不同的安全需求与应用细节，这样可以将一个复杂的问题简单化。但是设计者要考虑到实际应用前的前瞻性，有时候我们并不知道这些具体的需求与细节是什么；为了能够描述和了解这些细节，就需要我们在安全策略的指导下对安全涉及到的领域和相关做细致的考查和研究。借助这些手段能够迫使我们在下面的讨论中，增加我们对于将安全策略应用到实际中、或是强加于实际应用而导致的问题的认知。总之，我们对上述问题认识的越充分，能够实现和解释的过程就更加精确细化，这一精确细化的过程有助于帮助我们建立和完善从实际应用中提炼抽象凝练的、用确切语言表述的安全策略。反过来，这个重新表述的安全策略就能够使我们更易于去完成安全框架中所设定的细节。

ISO7498标准是目前国际上普遍遵循的计算机信息系统互连标准，1989年12月国际标准化组织（ISO）颁布了该标准的第二部分，即ISO7498-2，并首次确定了开放系统互连（OSI）参考模型的信息安全体系结构。我国将其作为GB/T9387-2标准，并予以执行。按照ISO 7498-2中OSI安全体系结构中的定义，访问控制的安全策略有以下两种实现方式：基于身份的安全策略和基于规则的安全策略。目前使用的两种安全策略，他们建立的基础都是授权行为。就其形式而言，基于身份的安全策略等同于DAC安全策略，基于规则的安全策略等同于MAC安全策略。

安全策略的实施原则：安全策略的制定实施也是围绕主体、客体和安全控制规则集三者之间的关系展开的。

（1）最小特权原则：最小特权原则是指主体执行操作时，按照主体所需权利的最小化原则分配给主体权力。最小特权原则的优点是最大限度地限制了主体实施授权行为，可以避免来自突发事件、错误和未授权用主体的危险。也就是说，为了达到一定目的，主体必须执行一定操作，但他只能做他所被允许做的，其它除外。

（2）最小泄漏原则：最小泄漏原则是指主体执行任务时，按照主体所需要知道的信息最小化的原则分配给主体权力。

（3）多级安全策略：多级安全策略是指主体和客体间的数据流向和权限控制按照安全级别的绝密（TS）、秘密（S）、机密（C）、限制（RS）和无级别（U）五级来划分。多级安全策略的优点是避免敏感信息的扩散。具有安全级别的信息资源，只有安全级别比他高的主体才能够访问。

6.2.2基于身份的安全策略

基于身份的安全策略（IDBACP：Identification-based Access Control Policies）与我们在上章阐述的鉴别行为一致，它的目的是过滤对数据或资源的访问，只有能通过认证的那些主体才有可能正常使用客体的资源。基于身份的安全策略的实例见图6.2.1，这是以访问控制矩阵的形式实现的。基于身份的策略包括基于个人的策略和基于组的策略。

1．基于个人的策略：
基于个人的策略（IDLBACP：Individual-based Access Control Policies）是指以用户为中心建立的一种策略，这种策略由一些列表来组成，这些列表限定了针对特定的客体，哪些用户可以实现何种策操作行为。例如，在图6.2.1中，对文件2而言，授权用户B有只读的权利，授权用户A则被允许读和写；对授权用户N而言，具有对文件1、2和文件N的读写权利。

上图策略的实施默认使用了最小特权原则，对于授权用户B，只具有读文件2的权利。

2．基于组的策略：
基于组的策略（GBACP：Group-based Access Control Policies）是基于个人的策略的扩充，指一些用户被允许使用同样的访问控制规则访问同样的客体。在图6.2中，授权用户A对文件1有读和写的权利，授权用户N同样被允许读和写对文件1，则对于文件1而言，A和N基于同样的授权规则；对于所有的文件而言，从文件1、2到N，授权用户A和N都基于同样的授权规则，那么A和N可以组成一个用户组G。图6.2.1.实现可以用图6.2.2表示，并且访问控制矩阵可以省略一行。

基于身份的安全策略有两种基本的实现方法：（1）能力表（2）访问控制列表。这两种实现机制我

们在下一节阐述，这是按照被授权访问的信息为访问者所拥有，还是被访问数据的一部分而区分的。

6.2.3基于规则的安全策略

基于规则的安全策略中的授权通常依赖于敏感性。在一个安全系统中，数据或资源应该标注安全标记。代表用户进行活动的进程可以得到与其原发者相应的安全标记。

基于规则的安全策略在实现上，由系统通过比较用户的安全级别和客体资源的安全级别来判断是否允许用户可以进行访问。

6.3 访问控制的实现

6.3.1访问控制的实现机制

建立访问控制模型和实现访问控制都是抽象和复杂的行为，实现访问的控制不仅要保证授权用户使用的权限与其所拥有的权限对应，制止非授权用户的非授权行为；还要保证敏感信息的交叉感染。为了便于讨论这一问题，我们以文件的访问控制为例对访问控制的实现做具体说明。通常用户访问信息资源（文件或是数据库），可能的行为有读、写和管理。为方便起见，我们用Read或是R表示读操作，Write或是W表示写操作，Own或是O表示管理操作。我们之所以将管理操作从读写中分离出来，是因为管理员也许会对控制规则本身或是文件的属性等做修改，也就是修改我们在下面提到的访问控制表。

6.3.2访问控制表

访问控制表（ACLs：Access Control Lists）是以文件为中心建立的访问权限表，简记为ACLs。图6.3.1清晰的表明了这种关系。目前，大多数PC、服务器和主机都使用ACLs作为访问控制的实现机制。访问控制表的优点在于实现简单，任何得到授权的主体都可以有一个访问表，例如授权用户A1的访问控制规则存储在文件File1中，A1的访问规则可以由A1下面的权限表ACLsA1来确定，权限表限定了用户UserA1的访问权限。

6.3.3访问控制矩阵

访问控制矩阵（ACM：Access Control Matrix）是通过矩阵形式表示访问控制规则和授权用户权限的方法；也就是说，对每个主体而言，都拥有对哪些客体的哪些访问权限；而对客体而言，又有哪些主体对他可以实施访问；将这种关连关系加以阐述，就形成了控制矩阵。其中，特权用户或特权用户组可以修改主体的访问控制权限。访问控制的实现见图6.3.2。访问控制矩阵的实现很易于理解，但是查找和实现起来有一定的难度，而且，如果用户和文件系统要管理的文件很多，那么控制矩阵将会成几何级数增长，这样对于增长的矩阵而言，会有大量的空余空间。

6.3.4访问控制能力列表

能力是访问控制中的一个重要概念，它是指请求访问的发起者所拥有的一个有效标签（ticket），它授权标签表明的持有者可以按照何种访问方式访问特定的客体。访问控制能力表（ACCLs：Access Control Capabilitis Lists）是以用户为中心建立访问权限表，ACCLs的具体实现见图6.3.3。例如，访问控制权限表ACCLsF1表明了授权用户UserA对文件File1的访问权限，UserAF表明了UserA对文件系统的访问控制规则集。因此，ACCLs的实现与ACLs正好相反。定义能力的重要作用在于能力的特殊性，如果赋予哪个主体具有一种能力，事实上是说明了这个主体具有了一定对应的权限。能力的实现有两种方式，传递的和不可传递的。一些能力可以由主体传递给其他主体使用，另一些则不能。能力的传递牵扯到了授权的实现，我们在后面会具体阐述访问控制的授权管理。

6.3.5访问控制安全标签列表

安全标签是限制和附属在主体或客体上的一组安全属性信息。安全标签的含义比能力更为广泛和严格，因为它实际上还建立了一个严格的安全等级集合。访问控制标签列表（ACSLLs： Access Control Security Labels Lists）是限定一个用户对一个客体目标访问的安全属性集合。访问控制标签列表的实现示例见图6.3.4，左侧为用户对应的安全级别，右侧为文件系统对应的安全级别。假设请求访问的用户UserA的安全级别为S，那么UserA请求访问文件File2时，由于S<TS，访问会被拒绝；当UserA请求访问文件FileN时，因为S>C，所以允许访问。

用户 安全级别
UserA S
UserB C
…… ……
UserX TS
文件 安全级别
File1 S
File2 TS
…… ……
FileN C

图6.3.4 访问控制标签列表的实现示例
安全标签能对敏感信息加以区分，这样就可以对用户和客体资源强制执行安全策略，因此，强制访问控制经常会用到这种实现机制。

6.3.6访问控制实现的具体类别

访问控制是网络安全防范和保护的重要手段，它的主要任务是维护网络系统安全、保证网络资源不被非法使用和非常访问。通常在技术实现上，包括以下几部分：

（1）接入访问控制：接入访问控制为网络访问提供了第一层访问控制，是网络访问的最先屏障，它控制哪些用户能够登录到服务器并获取网络资源，控制准许用户入网的时间和准许他们在哪台工作站入网。例如，ISP服务商实现的就是接入服务。用户的接入访问控制是对合法用户的验证，通常使用用户名和口令的认证方式。一般可分为三个步骤：用户名的识别与验证、用户口令的识别与验证和用户帐号的缺省限制检查。

（2）资源访问控制：是对客体整体资源信息的访问控制管理。其中包括文件系统的访问控制（文件目录访问控制和系统访问控制）、文件属性访问控制、信息内容访问控制。文件目录访问控制是指

用户和用户组被赋予一定的权限，在权限的规则控制许可下，哪些用户和用户组可以访问哪些目录、子目录、文件和其他资源，哪些用户可以对其中的哪些文件、目录、子目录、设备等能够执行何种操作。系统访问控制是指一个网络系统管理员应当为用户指定适当的访问权限，这些访问权限控制着用户对服务器的访问；应设置口令锁定服务器控制台，以防止非法用户修改、删除重要信息或破坏数据；应设定服务器登录时间限制、非法访问者检测和关闭的时间间隔；应对网络实施监控，记录用户对网络资源的访问，对非法的网络访问，能够用图形或文字或声音等形式报警等。文件属性访问控制：当用文件、目录和网络设备时，应给文件、目录等指定访问属性。属性安全控制可以将给定的属性与要访问的文件、目录和网络设备联系起来。

（3）网络端口和节点的访问控制：网络中的节点和端口往往加密传输数据，这些重要位置的管理必须防止黑客发动的攻击。对于管理和修改数据，应该要求访问者提供足以证明身份的验证器（如智能卡）。

6.4 安全级别与访问控制

6.4.1安全级别介绍

安全级别有两个含义，一个是主客体信息资源的安全类别，分为一种是有层次的安全级别（Hierarchical Classification）和无层次的安全级别；另一个是访问控制系统实现的安全级别，这和计算机系统的安全级别是一样的，分为四级：具体为D、C（C1、C2）、B（B1、B2、B3）和A四部分。

6.4.2安全级别的内涵

（1）D级别：
D级别是最低的安全级别，对系统提供最小的安全防护。系统的访问控制没有限制，无需登陆系统就可以访问数据，这个级别的系统包括DOS，WINDOWS98等。

（2）C级别有两个子系统，C1级和C2。

C1级称为选择性保护级（Discrtionary Security Protection）可以实现自主安全防护，对用户和数据的分离，保护或限制用户权限的传播。

C2级具有访问控制环境的权力，比C1的访问控制划分的更为详细，能够实现受控安全保护、个人帐户管理、审计和资源隔离。这个级别的系统包括UNIX、LINUX和WindowsNT系统。

C级别属于自由选择性安全保护，在设计上有自我保护和审计功能，可对主体行为进行审计与约束。C级别的安全策略主要是自主存取控制，可以实现
①保护数据确保非授权用户无法访问；
②对存取权限的传播进行控制；
③个人用户数据的安全管理。

C级别的用户必须提供身份证明，（比如口令机制）才能够正常实现访问控制，因此用户的操作与审计自动关联。C级别的审计能够针对实现访问控制的授权用户和非授权用户，建立、维护以及保护审计记录不被更改、破坏或受到非授权存取。这个级别的审计能够实现对所要审计的事件，事件发生的日期与时间，涉及的用户，事件类型，事件成功或失败等进行记录，同时能通过对个体的识别，有选择地审计任何一个或多个用户。C级别的一个重要特点是有对于审计生命周期保证的验证，这样

可以检查是否有明显的旁路可绕过或欺骗系统，检查是否存在明显的漏路（违背对资源的隔离，造成对审计或验证数据的非法操作）。

（3）B级别包括B1、B2和B3三个级别，B级别能够提供强制性安全保护和多级安全。强制防护是指定义及保持标记的完整性，信息资源的拥有者不具有更改自身的权限，系统数据完全处于访问控制管理的监督下。

B1级称为标识安全保护（Labeled Security Protection）。

B2级称为结构保护级别（Security Protection），要求访问控制的所有对象都有安全标签以实现低级别的用户不能访问敏感信息，对于设备、端口等也应标注安全级别。

B3级别称为安全域保护级别（Security Domain），这个级别使用安装硬件的方式来加强域的安全，比如用内存管理硬件来防止无授权访问。B3级别可以实现：
①引用监视器参与所有主体对客体的存取以保证不存在旁路；
②审计跟踪能力强，可以提供系统恢复过程；
③支持安全管理员角色；
④用户终端必须通过可信话通道才能实现对系统的访问；
⑤防止篡改。

B组安全级别可以实现自主存取控制和强制存取控制，通常的实现包括：
①所有敏感标识控制下的主体和客体都有标识；
②安全标识对普通用户是不可变更的；
③可以审计(a)任何试图违反可读输出标记的行为(b)授权用户提供的无标识数据的安全级别和与之相关的动作(c)信道和I/O设备的安全级别的改变(d)用户身份和与相应的操作；
④维护认证数据和授权信息；
⑤通过控制独立地址空间来维护进程的隔离。

B组安全级别应该保证：
①在设计阶段，应该提供设计文档，源代码以及目标代码，以供分析和测试；
②有明确的漏洞清除和补救缺陷的措施；
③无论是形式化的，还是非形式化的模型都能被证明该模型可以满足安全策略的需求。监控对象在不同安全环境下的移动过程（如两进程间的数据传递）
（4）A级别称为验证设计级（Verity Design），是目前最高的安全级别，在A级别中，安全的设计必须给出形式化设计说明和验证，需要有严格的数学推导过程，同时应该包含秘密信道和可信分布的分析，也就是说要保证系统的部件来源有安全保证，例如对这些软件和硬件在生产、销售、运输中进行严密跟踪和严格的配置管理，以避免出现安全隐患。

6.5 访问控制与授权

6.5.1授权行为

授权是资源的所有者或者控制者准许他人访问这种资源，这是实现访问控制的前提。对于简单的个体和不太复杂的群体，我们可以考虑基于个人和组的授权，即便是这种实现，管理起来也有可能是困难的。当我们面临的对象是一个大型跨国集团时，如何通过正常的授权以便保证合法的用户使用公司公布的资源，而不合法的用户不能得到访问控制的权限，这是一个复杂的问题。

授权是指客体授予主体一定的权力，通过这种权力，主体可以对客体执行某种行为，例如登陆，查

看文件、修改数据、管理帐户等。授权行为是指主体履行被客体授予权力的那些活动。因此，访问控制与授权密不可分。授权表示的是一种信任关系，需要建立一种模型对这种关系进行描述。本节将阐述信任模型的建立与信任管理。

6.5.2信任模型

概念和定义：信任模型（Trust Model）是指建立和管理信任关系的框架。信任关系是这样一种情形，如果主体能够符合客体所假定的期望值，那么称客体对主体是信任的。信任关系可以使用期望值来衡量，我们用信任度表示。主客体间建立信任关系的范畴我们称为信任域，也就是主客体和信任关系的范畴集合，信任域是服从于一组公共策略的系统集。

信任模型：信任模型有三种基本类型：层次信任模型、网状信任模型和对等信任模型。

层次信任模型：层次信任模型是实现最简单的模型，使用也最为广泛。建立层次信任模型的基础是所有的信任用户都有一个可信任根。例如我们通常所说的根管理员，事实上就是处于根的位置。所有的信任关系都基于根来产生。层次信任模型的示意图见图6.5.1，这是一个简单的三层信任结构。层次信任关系是一种链式的信任关系，比如可信任实体A1可以表示为这样一个信任链：（R，C1，A1），说明可以由A1向上回溯到产生他的信任根R。这种链式的信任关系我们称为信任链。层次信任模型是一种双向信任的模型，假设Ai和Bj是要建立信任关系的双方，Ai和Bj间的信任关系很容易建立，因为他们都基于可信任根R。层次信任模型对应于层状结构，有一个根节点R作为信任的起点，也就是信任源。这种建立信任关系的起点或是依赖点我们称为信任锚。信任源负责下属的信任管理，下属再负责下面一层的信任管理，这种管理方向是不可逆的。这个模型的信任路径是简单的，从根节点到叶子节点的通路构成了简单唯一的信任路径。层次信任模型的优点在于结构简单，管理方面，易于实现。他的缺点是Ai和Xk的信任关系必须通过根来实现，而可信任根R是默认的，无法通过相互关系来验证信任。一旦信任根出现问题，那么信任的整个链路就被破坏了。现实世界中，往往建立一个统一信任的根是困难的。对于不在一个信任域中的两个实体如何来建立信任关系？这用一个统一的层次信任模型来实现需要在建立信任的框架中预留有未来的发展余量，而且必须强迫信任域中的各方都统一信任可信任根R。

层次信任模型适用于孤立的、层状的企业，对于有组织边界交叉的企业，要应用这种模型是很困难的。另外，在层次信任模型的内部必须保持相同的管理策略。层次信任模型主要使用在以下三种环境：
（1）严格的层次结构；
（2）分层管理的PKI商务环境；
（3）PEM（Privacy-Enhanced Mail，保密性增强邮件）环境。

对等信任模型：对等信任模型是指两个或两个以上对等的信任域间建立的信任关系，对等信任模型的示意图见6.5.2。相对而言，对等信任关系灵活一些，他可以解决任意已经建立信任关系的两个信任模型之间的交互信任。不同信任域的A1和X1之间的信任关系要通过对等信任域R1和R2的相互认证才能实现，因此这种信任关系在PKI领域中又叫做交叉认证。建立交叉认证的两个实体间是对等的关系，因为他们既是被验证的主体，又是进行验证的客体。对等信任模型不会建立在信任域以外，这是因为如果任意两个主客体都建立对等信任的话，那么对于N个主客体而言，需要建立N×（N-1）/2个信任链。

对等信任模型这种结构非常适合表示动态变化的信任组织结构，这样，引入一个可信任域是易于实现的。但是在构建有效的认证路径时，也就是说，假定A1和Xk是建立信任的双方，那么，很难在整个信任域中确定R2是否是Xk的最适当的信任源。

网状信任模型：网状信任模型可以看成是对等信任模型的扩充。我们没有必要在任意两个对等的信任域建立交叉认证，完全可以通过建立一个网络拓扑结构的信任模型来实现，也就是建立信任域间的间接信任关系。网状信任模型的示例见图6.5.3。假设R1，R2～R11是不同的信任域，他们之间的信任关系用实线箭头表示。那么分别位于R1和R5信任域下的主体A和B 间可以建立的信任链共有三条，通过图中的虚线来表示。

建立一个恰当合理的信任网络模型比我们想象的要复杂的多。我们在本章的第二节曾经探讨过安全标签列表的实现，这是引入安全级别和考虑保护敏感信息的必然。同样，在建立的对等或是非对等的信任集合中，很难想象一个安全级别低（例如C级别）的信任域和一个安全级别高（例如S级别）的信任域，在他们中间建立的信任模型是什么样子的。因为对整个信任域的信任链的可信程度很难不令人质疑，S级别可能需要通过使用智能卡才能通过访问控制最初的验证，而C级别也许只是进行简单的IP地址检验就可以任意访问客体的信息资源。在建立信任模型，实现访问控制的过程中，不但要选择合适的信任模型，保护客体的资源，也应该避免主体的信息资源暴露在攻击和危险的情况下；这种情况下，主客体信息的交换有时候更多的依赖于可信第三方。另外，网络资源和时限也是一个问题，尽管A和B间有三条信任链可以实现，但我们总是希望耗用最少的时间，也就是说，走最短的路径，那么，怎样来计算这条路径也是一个困难的问题。

其次，跨越多个可信域根建立的漫长的非层状的信任路径被认为是不可信的，显然在这样的信任关系实现上，构造合理的信任路径和检验适当的信任锚都是巨大的挑战。因为我们不得不对不同的信任锚进行验证，不得不要建立一个从被信任发起方开始到信任到达者所在信任域的完整的信任路径，每一个验证者还需要建立自己到信任锚的路径。同时，信任路径中的封闭环路一定要检测出来并丢弃掉，对可能存在的多条路径也要进行过滤和优先级的设置。

6.5.3信任管理系统

阐述信任模型很容易产生一个问题，这就是在实际中是由谁在管理信任？如果我们就是信任中的主体，我们凭什么信任他们？这就是信任管理需要解决的问题。

信任管理的产生和现状：信任管理的产生是一个漫长而复杂的过程，这和企业的发展与市场的制约有很大关系。现代企业有向大型化、集团化发展的趋势，一个企业往往包括多个职能部门，分别完成生产、管理、结算等功能，而这些职能部门又可划分为多个各司其职的更小的部门，与此同时企业内部的职能划分越来越细，独立运作能力也越来越强，可以独立和别的企业的相应或相关职能部门进行交易，所以在现实的商业运作中企业内部的多级管理，和企业间的无级别贸易是并存的。这种关系必然反映在信任管理中，怎么来实现和约束正确的信任关系来访问资源和进行交易，建立相应的信任关系。目前，层次信任模型的建立和管理在一定的信任域内建立是正常的，但在信任域间的交叉认证和混和多级信任模型方面，还没有就信任管理达成一致。

信任管理包含了两个方面，一个是对于信任链的维护与管理，一个是对信任域间信任关系的管理与维护。用户是信任的主要参与者，因此用户有必要对信任链加以管理，也就是说应该由他自己来判断是否该相信谁和该相信什么。信任域的管理通常由认证机构来负责。

6.6 访问控制与审计

6.6.1审计跟踪概述

审计是对访问控制的必要补充，是访问控制的一个重要内容。审计会对用户使用何种信息资源、使用的时间、以及如何使用（执行何种操作）进行记录与监控。审计和监控是实现系统安全的最后一道防线，处于系统的最高层。审计与监控能够再现原有的进程和问题，这对于责任追查和数据恢复非常有必要。

审计跟踪是系统活动的流水记录。该记录按事件从始至终的途径，顺序检查、审查和检验每个事件的环境及活动。审计跟踪通过书面方式提供应负责任人员的活动证据以支持访问控制职能的实现（职能是指记录系统活动并可以跟踪到对这些活动应负责任人员的能力）。审计跟踪记录系统活动和用户活动。系统活动包括操作系统和应用程序进程的活动；用户活动包括用户在操作系统中和应用程序中的活动。通过借助适当的工具和规程，审计跟踪可以发现违反安全策略的活动、影响运行效率的问题以及程序中的错误。审计跟踪不但有助于帮助系统管理员确保系统及其资源免遭非法授权用户的侵害，同时还能提供对数据恢复的帮助。

6.6.2审计内容

审计跟踪可以实现多种安全相关目标，包括个人职能、事件重建、入侵检测和故障分析。

个人职能（individual accountability）：审计跟踪是管理人员用来维护个人职能的技术手段。如果用户被知道他们的行为活动被记录在审计日志中，相应的人员需要为自己的行为负责，他们就不太会违反安全策略和绕过安全控制措施。例如审计跟踪可以记录改动前和改动后的记录，以确定是哪个操作者在什么时候做了哪些实际的改动，这可以帮助管理层确定错误到底是由用户、操作系统、应用软件还是由其它因素造成的。允许用户访问特定资源意味着用户要通过访问控制和授权实现他们的访问，被授权的访问有可能会被滥用，导致敏感信息的扩散，当无法阻止用户通过其合法身份访问资源时，审计跟踪就能发挥作用。审计跟踪可以用于检查和检测他们的活动。

事件重建（reconstruction of events）：在发生故障后，审计跟踪可以用于重建事件和数据恢复。通过审查系统活动的审计跟踪可以比较容易地评估故障损失，确定故障发生的时间、原因和过程。通过对审计跟踪的分析就可以重建系统和协助恢复数据文件；同时，还有可能避免下次发生此类故障的情况。

入侵检测（intrusion detection）：审计跟踪记录可以用来协助入侵检测工作。如果将审计的每一笔记录都进行上下文分析，就可以实时发现或是过后预防入侵检测活动。实时入侵检测可以及时发现非法授权者对系统的非法访问，也可以探测到病毒扩散和网络攻击。

故障分析（problem analysis）：审计跟踪可以用于实时审计或监控。