信息安全模型研究及安全操作系统设计①

Research on Information Secur ity Model and the Design of Security OS

蔡 勉 金 怡 （北京工业大学 电子信息与控制工程学院 北京 100022）

摘要：本文介绍信息流的格模型、DTE 模型和 RBAC 模型，在这三种安全策略模型的基础上，设计了一个达到结构化保护级要求的安全操作系统方案，提供安全系统的组成和功能，使 Linux 操作系统达到高安全保护级别。

关键词：Linux 操作系统 安全操作系统 安全模型

1 引言

本文基于原有 Linux 操作系统，设计一个满足结构化保护级（相当于《TCSEC》标准的 B2 级，我国《计算机信息系统安全保护等级划分准则》的第四级）安全功能的实用操作系统。该安全操作系统结合了国内外的相关安全标准和已有的先进技术，具有强大存取控制机制，能更好的满足“保密性”、“完整性”、“可用性”和“隐私性”等要求，可以作为高度安全的服务器的操作系统使用。

2 信息安全模型分析

本文所设计的安全模型是从安全策略和访问控制的角度对安全系统进行设计。访问控制要从强制访问控制和自主访问控制两方面去实现。

2.1 信息流的格模型

信息流模型是访问控制模型的一种变形。这类模型不检查主体对客体的存取，而是试图控制从一个客体到另一个客体的信息传输过程，根据两个客体的安全属性决定操作是否进行。信息流模型与访问控制模型之间的差异很小，但这很小的差别，却能完成访问控制模型无能为力的工作—识别隐蔽信道。隐蔽信道的核心是间接存取。信息流分析能保证操作系统模型在对敏感信息存取时不会将数据泄露给所调用的模块。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。信息流模型需要遵守的安全规则是：当且仅当一个操作序列的执行不会产生违反两个安全类之间的信息流动所规定的方向时，那么流模型是安全的。

Denning 把一个信息流模型 FM 定义为：

$$FM = (N, P, SC, \oplus, \rightarrow)$$

其中 N,P 分别为客体和进程的有限集，SC 是安全类的有限集，类运算符“⊕”为一个满足结合律和交换律的二进制运算符。对任意两个操作数的类，“⊕”指定了它们之间任意二进制函数产生的结果操作数所属的类。流关系“→”指定了两个安全类之间的信息流动。当且仅当类型 A 中的信息允许流入类 B，记为 $A \rightarrow B$。

信息流模型的安全策略可简单表述为：当且仅当一个操作序列的执行不会产生违反关系“→”所规定的信息流，那么信息流模型 FM 是安全的。这一策略在下面假设下，$(SC, \oplus, \rightarrow)$ 形成了一个有限格：

（1）$(SC, \rightarrow)$ 是一个偏序集；

（2）SC 是有限集；

（3）SC 有一个下界 L，$\forall A \in SC, L \rightarrow A$；

① 基金项目：北京市教委（KM200610005028）

(4) ⊕是在 SC 上的最小上界运算符。

最小上界运算符⊕有以下性质:对于所有 A,B, C ∈SC

(1) A →A⊕B 和 B →A⊕B ;

(2) A → C 且 B →C⇒A⊕B → C[3,4]。

2.2 DTE

DTE 模型最初由 Boebert 和 Kain 提出,经修改后在 LOCK 系统中得到实现。与其他的访问控制机制一样,DTE 将系统视为一个主动实体(主体)的集合和一个被动实体(客体)的集合。每个主体有一个属性—域,每个客体有一个属性—类型,这样,所有的主体被划分到若干个域中,所有的客体被划分到若跟个类型中。DTE 再建立一个表"域定义表"(DDT Domain Definition Table),描述各个域对不同类型客体的操作权限,系统按照规则,限制域中的用户对类型中的资源的访问,同时限制域中的用户对其他域中的用户的访问。同时建立另一种表"域交互表"(DTT Domain Interaction Table),描述各个域之间的许可访问模式(如:创建、发信号、切换)。系统运行时,依据访问的主体域和客体类型,查找域定义表,决定是否允许访问。基于 DTE 的安全策略配置实现了安全的可实现机制与安全策略的定义相分离,有效地简化了安全控制,并提供了与原有系统的向后兼容性。

DTE 模型有政策中立的优势。但是在实用中,他的两张访问控制表会迅速膨胀,变得很复杂,很难验证其安全性。

2.3 RBAC

基于角色的访问控制属于访问控制系统化中的一种。所谓角色是指拥有一个权限和责任集的某一特定职位。基于角色的访问控制在用户和资源之间加入了角色,把对资源的使用权赋给角色,然后让用户属于某一角色,从而使用户具有角色的权限。这样,整个访问控制过程就分成两个部分,即访问权限与角色相关联,角色再与用户关联,从而实现了用户与访问权限的逻辑分离。在 RBAC 中还可以包括角色层次和角色约束等机制。

RBAC 的基本思想是:授权给用户的访问权限,通常由用户在一个组织中担当的角色来确定。例如,一个公司内可以包含的角色可以有会计师,部门经理和采购员等。由于他们的职能不同,所拥有的访问权限显然也各不相同。RBAC 根据用户在组织内所处的角色进行访问授权与控制。也就是说,传统的访问控制直接将访问主体(发出访问操作、存取要求的主动方)和客体(被调用的程序或欲存取数据访问)相联系,而 RBAC 在中间加入了角色,通过角色沟通主体与客体。在 RBAC 中,用户标识对于身份认证以及审计记录是十分有用的,但真正决定访问权限的是用户对应的角色标识。RBAC 的基本思想如图 1 所示。

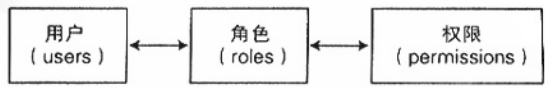

图 1　RBAC 的基本思想

与基于安全级别和类别纵向划分的安全控制机制相比,RBAC 显示了较多的机动灵活的优点。特别显著的特点是,RBAC 在不同的系统配置下可以显示不同的安全控制功能,既可以构造具备自主访问控制类型的系统,也可以构造成为强制访问控制类型的系统,甚至可以构造同时具备这两种访问控制的系统。

Sandhu 教授已经指出:RBAC 模型是一个开放式的概念模型,可以依据不同的应用进行扩展。基于此,本论文就把 RBAC 策略用在域 – 类型加强型系统中。

3　安全体系结构

Flask 模型是由美国 NSA(National Security Agency)、SCC(Secure Computing Corporation)和 Utah 大学在 Fluke 操作系统项目中提出的安全模型。与一般的的安全模型

比较,Flask 模型较好地支持了动态安全策略。Flask 安全模型体系结构如图 2 所示。

Flask 安全结构,如下图所示,描述了执行安全策略判定的子系统,作判定的子系统,每个子系统组件的需求之间的相互操作。

Flask 安全结构为对象管理器提供了三个主要功能。第一，结构提供从一个安全服务器重新访问、标记和多实例判定的接口。访问判定规定在两个实体间一个特定的许可权是否被允许，特别是一个主体和对象之间。标记判定规定分配给对象的安全属性。多实例判定规定多实例资源集的哪个成员被特定的请求访问。第二，结构提供一个访问向量缓存（AVC）模块允许对象管理缓存访问判定来减小性能损耗。第三，结构提供对象管理器注册接受安全策略的改变通知的能力。

4 安全系统的设计

形式化的安全模型是设计开发高级别安全操作系统的前提。一个操作系统是安全的，是指它满足某一给定的安全策略，安全模型则是对安全策略所表达的安全需求简单、抽象和无歧义的描述。目前，人们只公认少数几个安全模型，用于实际系统的就更少了。本系统采用 Flask 安全体系结构，系统内核分为安全策略判定和具体的实施机制两部分，安全策略逻辑与通用

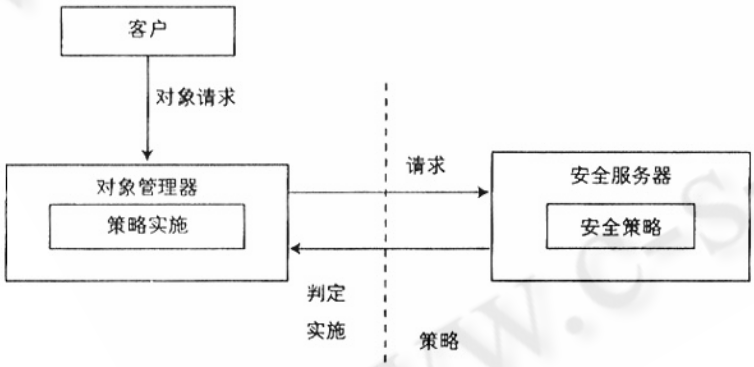

图 2　Flask 结构

接口一起封装在一个独立的组件中，通用接口用于获取安全策略决策。在系统实现中，这个组件被称为安全服务器（security server），它是内核中的一个子系统，在内部设立相互独立的政策支持机制，每个安全政策对应一个政策支持机制，这样做的目的是希望在安全政策的支持方面获得一定的灵活性；而负责具体的安全策略实施的组件称为对象管理器（object manager），对象管理器从安全服务器取得具体的安全策略决策，并将其作用于安全性标签以控制对客体的访问。执行代码从安全服务器和 AVC 缓存模块得到安全策略，并应用这些安全策略来为进程分配安全性标签及控制基于这些安全标签的操作。系统中的其他内核子系统都是对象管理器。本文对上述的几种安全模型作深入的研究，并面向自己的安全策略作修改，以满足安全策略的需求，本系统中的安全服务器将遵循改进的信息流模型、DTE 模型、RBAC 模型来实现系统的安全策略。

信息流的格模型只是片面强调保密性，DTE 模型片面强调完整性，而在当今社会，保密性和完整性不管对国家和企业都有着非常重要的作用。针对现在的系统保密性和完整性不能同时兼顾的弱点，本系统的设计采用了信息流的格模型和 DTE 模型实现了保密性和完整性的双重要求，并用 RBAC 模型管理信息流的格模型和 DTE 模型来实现安全管理。其中，采用信息流的格模型和 DTE 完整性模型来实现强制访问控制，采用 RBAC 模型实现安全机制管理。

本系统中 DTE 定义了多个域（Domain）和型（Type），并将系统中的主体分配到不同的域中，不同的客体分配到不同的型中，通过定义不同的域对不同的型的访问权限，以及主体在不同的域中进行转换的规则来达到保护信息完整性的目的。对用户应用系统的控制，主要采用角色模型与 DTE 模型的结合，每个进程都有一个与之相关的角色：系统进程以 system_r 角色运行，而用户可以是 user_r 或 sysadim_r；而对 Linux 内核的控制，则通过权能访问控制、信息流的格模型来实现。

4.1 安全系统的总体结构

Linux 结构化保护级系统结构如图 3 所示。用户请求的系统操作进入内核空间后，首先经过安全策略检查点调用相应的安全策略，例如，实现强制访问控制策略时，在检查点被触发时，向 MAC 决策模块发出请求，MAC 访问决策模块根据相应的安全策略来决定对请求授权还是拒绝。如果授权此次访问，则访问往下

进行。如果拒绝此次访问请求，则此次访问失败。

安全服务器中的功能模块与原有的系统操作是相对独立的，双方通过 hook 函数进行联系。这样，可以通过改变 hook 函数的指向，可以使用不同的安全策略。

4.2 安全操作系统的功能设计

(1) 实现强制访问控制。本系统是基于信息流的格模型和 DTE 完整性模型实现强制访问控制

保密性强制访问控制

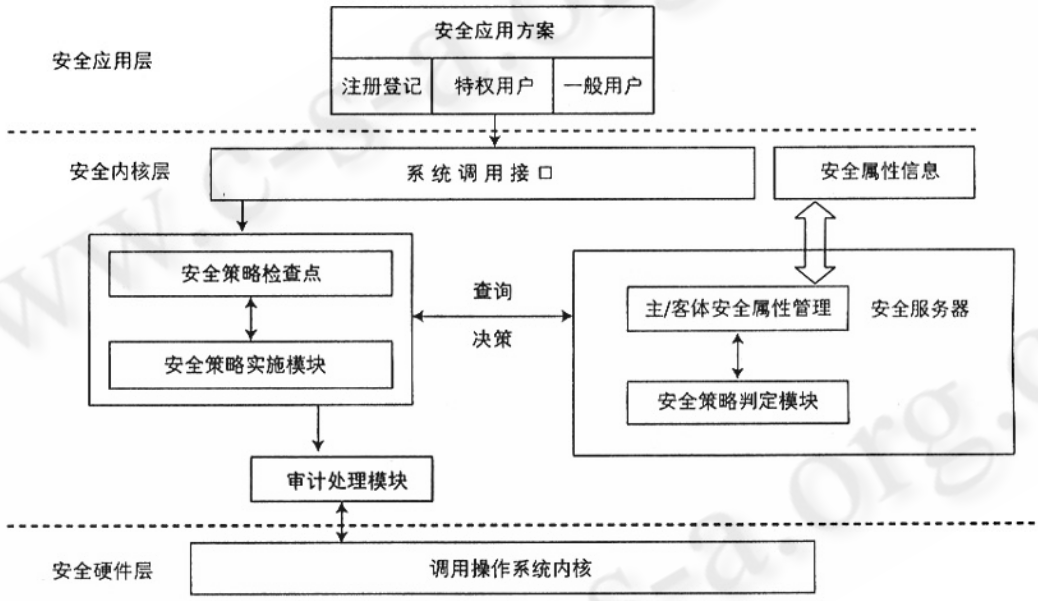

图 3　Linux 结构化保护级系统结构

以信息流模型为基础，对信息流实施强制访问控制。这类模型部检查主体对客体的存取，而是试图控制从一个客体到另一个客体的信息传输过程，根据两个客体的安全属性决定操作是否进行。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。

完整性强制访问控制

DTE 模型将系统中的主体分配到不同的域中，不同的客体分配到不同的型中，通过定义不同的域对不同的型的访问权限，以及主体在不同的域中进行转换的规则来达到保护信息完整性的目的。

(2) 自主访问控制。本系统的自主访问控制是在传统 Linux 的自主访问控制的基础之上，通过加入访问控制列表(ACL)机制，用户能够有选择的授予其他用户某些访问权限，细化了对系统资源的访问控制粒度。并且自主访问控制粒度要达到单个用户。一个典型的 ACL 机制如下所示：<Type,Id,Perm>，其中 Type 表示进程号的类型即 Id 是 uid 还是 gid，Perm 表示对应的访问权限。

(3) 增强的身份标识和鉴别。实现一个独立的文件系统来作为主、客体标志与鉴别系统。通过强化的口令管理，增强用户身份的标识与鉴别。不仅检查用户的登录名和口令、赋予用户唯一标识 uid、gid，还检查用户的安全级，赋予用户进程；根据登录角色和安全域，计算初始进程的特权集。该系统要实现对大量用户的集中管理，能够处理用户的并发查询。

(4) 实现系统特权分割。由于超级用户特权是操作系统中最重要的脆弱点之一，将超级用户特权分割为几个相互平行、相互制约的管理员角色是操作系统的发展趋势。

本系统根据最小特权原则对系统管理员的特权进

行了分割。最小特权管理的思想是指将系统原有的超级用户(如 Root)的特权划分为一组细粒度的特权,分别授给不同的系统操作员/管理员,使各种系统管理员/操作员只具有完成其任务所需的特权。

RBAC 模型包含了 3 个实体:用户 User、角色 Role 和许可 Privilege。其控制方式是"用户 - 角色 - 访问数据库资源的许可"。系统可为用户指定担任的角色,实现了用户与角色的分离。允许定义、配置安全角色以及角色之间的互斥关系。系统将超级用户的权限进行更细粒度划分,根据权限级并从安全管理分层角度来看,初始设置 3 个角色:系统管理员、系统审计员和系统安全管理员。其中,系统管理员具有原 Root 帐号的大部分权限,系统审计员管理系统的设计和监控等内容,系统安全管理员管理安全运行。该 3 个管理员要遵循三权分离原则,各司其职,互相监督。

(5) 实现安全审计功能。审计机制的具体实现是,在与安全性有关的函数中设置审计点,在审计点搜集到的各种审计事件信息汇到一起,调用这些操作事件的审计信息。并且要定期将这些审计事件写入所开辟缓冲区的文件中。对于发生在用户层的涉及系统安全的事件,要对应用程序提供审计接口。这些审计点分布在系统调用的出入口和有关命令中,由此调用审计进程将审计信息记录、转储和归档。并对审计配置文件、审计数据文件实施相应的强制访问策略,使得这些重要文件得到严格的保护,防止非授权查看、篡改和删除,保证只有审计管理员才能访问它们。

(6) 客体重用。客体重用是指 TSF 必须确保受保护资源内的任何信息,在资源被重用时不会被泄漏。因此,在资源被重新分配给新的主体时不能包含任何残留信息。

本系统处于效率的考虑,只在对核心数据中的残留信息进行自动清除。

(7) 隐蔽信道。信息流模型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。

(8) 可信路径。当连接用户时,计算机信息系统可信计算基提供它与用户之间的可信路径。可信路径避过应用层,在用户与内核之间开辟一条直接的可信任的交互通道,从而防止重要信息在不可信路径上传输时被盗用。

5 结论

本文通过对信息流的格模型、DTE 模型、RBAC 模型的研究,针对 linux 操作系统,设计了一个具有结构化保护级的安全操作系统。该操作系统增加了强制访问控制、最小特权等安全功能,并对隐蔽信道进行了分析。在安全性领域,系统的安全性是相对的,因此,对安全模型和操作系统方案设计的研究还有待进一步深入,而具体的安全操作系统的实现也势在必行。

参考文献

1 GB 17859 - 1999 中华人民共和国国家标准《计算机信息系统安全保护等级划分准则》,1999.

2 刘克龙、冯登国、石文昌, 安全操作系统原理与技术[M],北京 科学出版社,2004。

3 Denning D E. A lattice model of secure information flow[J],Commu ACM,1976 ,19(5) :236 ~ 243.

4 刘益和、刘嘉勇, 一个基于角色的信息流模型及应用[J], 四川大学学报,2004,36(05):94 - 97.

5 Badger L , Sterne D F , Sherman D L , Walker K M, Haghighat S A. Practical Domain and Type Enforcement for Unix [A]. Proceedings of the 1995 IEEE Symposium on Security and Privacy[C] Oakland : IEEE Computer Society Press ,1995 . 66 - 77.

6 Sandhu R S, Coyne E J , Feinstein H L , Youman C E. Role - Based Access Control Models [J]. IEEE Computer, 1996, 29(2) : 38 - 47.

7 袁春阳、梁洪亮, 高可信安全操作系统的开发及核心技术[J],信息网络安全,2005(3):47 - 49.

8 Waldhart N A. The Army Secure Operating System [A]. Proceedings of Research in Security and Privacy[C]. Oakland:IEEE,1990. 50 - 60.