网络安全策略求精一致性检测和冲突消解机制的研究

倪 俊[1,2] 陈晓苏[1] 刘辉宇[1] 李 劲[3]
(华中科技大学计算机科学与技术学院 武汉 430074)[1] (恩施州电力总公司 恩施 445000)[2]
(湖北民族学院信息工程学院 恩施 445000)[3]

摘 要 通过对基于策略的网络安全管理的研究,分析了现有网络安全策略冲突检测和消解方法存在的不足。基于策略求精的思想和安全策略冲突分类技术,建立基于策略的网络管理安全级模型,并用扩展的XACML语言加以描述。根据策略行为间的关系,采用知识推理技术,动态分层地对相应安全级策略进行一致性自动检测和实时冲突消解,使其具有良好的可重用性和可扩展性,以利于安全策略管理效率的提高。并通过策略求精访问控制的应用实现进行了验证。最后给出了未来的研究方向。

关键词 网络安全,策略求精,安全级模型,一致性自动检测,知识推理,冲突消解
中图法分类号 TP393 **文献标识码** A

Research on Network Security Policy Refinement Consistency of Detection and Conflict Resolution Mechanisms

NI Jun[1,2] CHEN Xiao-su[1] LIU Hui-yu[1] LI Jing[3]
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)[1]
(Enshi Electric Power Corporation, Enshi 445000, China)[2]
(Department of Information Engineering, Hubei University for Nationalities, Enshi 445000, China)[3]

Abstract Through policy-based network security management research, this paper analyzed the existing network security policy conflict detection and resolution method shortcomings. Based on policy refinement of ideas and security policy conflicts classification technology, policy-based network management security-level model was established, with extended XACML language description. According to the relationship between policy behavior, using knowledge reasoning, dynamic layered security corresponding level of policy refinement consistency automatic detection and timely conflict resolution were made, letting it has a good reusability and scalability, and is conducive to the improvement of management efficiency. Policy-based access control refinement application implementation was verified. Finally, some of the future research directions were discussed.

Keywords Network security, Policy refinement, Security-level model, Consistency of automatic detection, Knowledge reasoning, Conflict resolution

1 引言

基于策略的网络安全管理PBNSM(Policy Based Network Security Management)是与网络系统安全管理相关的行为规则描述,它可以在一定的抽象层次上指导系统的行为管理并使其保持一致性,因而成为近年来网络安全管理研究的热点和重点。但随着大规模区域性网络中访问主体、客体数量的不断膨胀,对不同的网络环境,网络安全策略的配置(如访问控制、授权、认证等)不仅越来越复杂,而且多域间各策略之间的冲突时有发生,严重影响安全策略的执行效率。因此,在网络安全系统同时执行多条不同的安全策略时,维护管理安全策略间的一致性就显得非常重要。如何保证不同层次安全策略的一致性,以及同一层次安全策略的一致性,并有效消解冲突,优化策略,成为应用安全策略的关键[1-3]。针对当前大部分安全策略冲突检测与消解算法缺少灵活性和扩展性等缺点[4-6],本文在文献[7]的基础上,提出了一个基于本体和规则相统一的一般性安全策略形式化描述方法,给出了基于策略的安全级模型及安全策略间的冲突分类描述,并定义了安全策略描述要素间的逻辑关系。在此基础上针对不同的冲突类型,设计了对应的可扩展的安全策略冲突检测与消解算法,以优化策略求精的流程结构,减少策略求精的计算成本,有利于安全策略管理效率的提高。

2 相关工作

近年来,国内外学者对策略冲突检测与消解作了大量的研究:一是基于IETF策略核心信息模型的关于管理策略的

到稿日期:2010-03-24 返修日期:2010-07-24 本文受国家863计划项目(2007AA10Z309)资助。
倪 俊(1966-),男,博士生,高级工程师,主要研究方向为网络安全和网络应用技术,E-mail:esznj@163.com;陈晓苏(1953-),男,教授,主要研究方向为网络安全和网络应用技术;刘辉宇(1978-),男,博士生,讲师,主要研究方向为网络安全和网络应用技术;李 劲(1973-),男,副教授,主要研究方向为网络应用技术。

冲突检测，主要代表为美国 IBM 公司研究员 Dinesh Verma 提出的多维空间思想和意大利都灵大学提出的代数方法；二是关于安全策略的冲突检测，主要有澳大利亚 Queenland 大学提出的基于逻辑描述语言的冲突检测方法[4]，英国 Imperial College 的 Lupu 等人在面向对象的策略表示语言 ponder[5] 和基于策略的分布系统管理框架的基础上提出的基于角色的访问控制方法[8]；三是使用逻辑编程语言进行冲突检测的方法[9]；四是英国 Imperial College 的 Bardar 等人针对说明性语言 ponder 不便于形式化推理的缺点，提出采用形式语言 EC (Event Calculus) 对其翻译，描述系统行为，再运用形式化的方法表现策略冲突[10]；五是 York 大学的 Schaad 研究了基于角色授权委托的冲突检测[11]，并使用 Prolog 作为分析工具；六是 George Mason 大学的 Jajodia 等人设计了一种名为 ASL (authorization specification language) 的语言[12]，研究了主体间的关系及由此产生的冲突与消解方法。本文侧重安全策略求精的冲突检测与消解。

3 安全策略表示和冲突分类

3.1 安全策略的定义及表示

根据 IETF/DMTF 的定义，策略由条件和行为组成。当条件满足时，规则中的动作能被触发，此时角色中的主体或客体就要去实现这个动作。简单的安全策略规则定义可描述为：

If {condition} then {action} role {role}

定义 1(安全策略 P)　$P=\{P[i]\ 1\leqslant i\leqslant |P|\}$，规定了安全策略的一般表达式。其中 $P[i]$为安全策略的属性特性。根据安全策略属性的多少和属性的性质，将安全策略分为形式安全策略和语义安全策略。当 $i\leqslant 3$ 时为形式安全策略；当 $i\geqslant 4$ 时为语义安全策略。

当 $i=3$ 时，$P=\{P[1],P[2],P[3]\}=\{$Subject, Object, Action$\}$，属形式安全策略，其策略检测分析是基于主体 Subject、客体 Object 和操作 Action 这三元组进行的。

当 $i=5$ 时，$P=\{P[1],P[2],P[3],P[4],P[5]\}=\{$Subject, Object, Action, Condition, Sign$\}$，属语义安全策略，其策略检测分析是基于主体 Subject、客体 Object、操作 Action、条件域 Condition 和策略授权符号 Sign 这五元组进行的。其中 subject 指主体域，可能由用户 U、用户组 UG 或角色 R 组成，通过主体的授权关系表示策略之间的委托关系；object 指客体域，主要表示目标资源对象；操作域 action 一般是动作序列，action$\in\{$read, write, …$\}$，这里假设 action 是原子的，相互无关；条件域 condition 为策略对系统资源进行作用时的外部限制，可用 s_scope 和 t_ scope 表示结构，其 s_scope$\in\{$'L', 'P'$\}$，表示主体授权范围，s_scope='L'称为主体本地授权 (local)，即授权仅作用于域的直接成员；s_ scope='P'称为主体传递授权 (propagate)，即授权作用于域的直接成员和间接成员。t_ scope 表示客体授权范围，其意义与 s_scope 一致；sign$\in\{$'+','-'$\}$，表示策略授权符号。

3.2 XACML 语言

XACML (Extensible Access Control Markup Language) 是 OASIS 2003 年制定的基于 XML 标准的一种通用的用于保护资源的策略语言和访问决策语言，其主要思想是围绕一个四元组 (subject, resource, action, condition) 来定义访问控制授权策略，具备可扩展性，支持参数化的策略描述和多样化的策略组合[13]，支持属性层次操作关联带来的多种冲突检测和消解[14]。XACML 规范定义了编码规则、策略绑定规则以及多规则或策略的选择和组合算法。扩展的 XACML 语言使得策略控制的描述制订更加灵活，制定的访问策略可以方便地应用于各种不同的场合。

3.3 安全策略冲突分类

根据冲突的来源和安全策略分层属性，将安全策略冲突分为形式冲突和语义冲突。

3.3.1 形式冲突

形式冲突是策略规范的不一致性引起的。当两条或多条策略在相同的主体、行为和目标上定义了相反的表示形式时，就会发生这种冲突。形式冲突的特征是策略的属性之间存在交集，因此可通过计算策略的特征属性之间是否存在交集来检测出来。最简单的形式冲突的情况是二条策略 $P_i(s_i,o_i,a_i)$和 $P_j(s_j,o_j,a_j)$，其中 $s_i\cap s_j=s_c$，$o_i\cap o_j=o_c$，$a_i\cap a_j=a_c$，如果 P_i 和 P_j 存在主体、客体和动作三元组的非空交集$\langle s_c,o_c,a_c\rangle\neq\emptyset$，且 P_i 为肯定授权、P_j 为否定授权或者 P_i 为否定授权、P_j 为肯定授权，则策略 P_i 和策略 P_j 存在冲突。形式冲突是由于域的相互覆盖引起的，但是阻止这些覆盖是不现实的，因为需要多个策略应用于同一个域，以反映职责的区别和管理功能的多样性。使用策略优先权关系能大量减少策略间冲突的数量，而且能允许不一致的策略定义显示的存在。

3.3.2 语义冲突

语义冲突指策略规则与系统逻辑规则或资源条件限制不符而引起的冲突。从冲突产生的方式来看，可分为多条策略共同存在导致与系统规则冲突和单条策略中的某一个或多个与系统规则冲突两种。语义冲突包含本体的一致性问题，这些本体涉及策略的属性(主体、目标和行为)以及外部标准(临界资源、全局原则以及特定的应用要求，如有限的资源或组织的总体策略)。因此大多数语义冲突不能通过策略的定义来预测。由于其与策略执行时的资源占用情况或策略属性的状态值有关，因此需用动态检测的方法。如根据策略具体的冲突种类，先判别出其关联的对象，然后在关联对象上附加属性标签来加以检测，因此评估和检测冲突时必须借助附加的元策略[15,16]限制条件。这种限制和基本策略类型中的限制在概念上是有区别的，基本策略类型中的约束是限制策略的使用性，而元策略是在策略服务中对被允许策略的一种约束。元策略解决语义冲突时有静态和动态两种方案。静态方案是指存在冲突的策略不能一同存在于一个系统中，动态方案是指存在冲突的策略不能在运行时同时激活。

4 网络安全策略求精的安全级模型

安全策略求精是一个包含安全策略一致性检测与冲突消解的细化过程。根据安全策略定义和求精管理需要，结合网络安全防护的逻辑结构特点，将网络安全管理分为不同的安全级，可以从抽象到具体在不同安全级对安全目标加以描述，以更好地控制和管理好网络，快速响应网络环境的动态变化，增强网络安全自动化防御能力。网络安全策略求精从抽象到具体的安全级一般可分为事务安全策略、服务安全策略、行为安全策略、操作安全策略等 4 级。不同级安全策略定义如下。

事务安全策略：是网络安全防护用多种不同的高级语言

或 GUI 方式定义的高级安全策略，描述网络应用的总体安全规范。事务安全策略可用自然语言编写，此级发生的安全策略冲突属语义冲突。

服务安全策略：是对网络安全防护需求的细化。对低层策略的抽象，采用形式化的安全策略规范语言描述。描述针对事务安全策略采用的安全服务规则，具有严格的表示形式。服务安全策略可用本体描述语言如扩展的 XACML 或 OWL 表示，此级策略冲突属语义冲突。

行为安全策略：对网络安全防护实现一致性检查、信任管理等功能，描述实现服务策略采用的安全行为。用 XACML 语言表示，使之适用于不同类型的安全设备，并保存到规则库。行为安全策略级发生冲突属形式冲突。

操作安全策略：安全策略的执行实体。对网络安全防护在行为策略的基础上进一步明确行为的具体执行环境，如行为的发起者、作用对象、执行条件、触发事件等。操作安全策略将 XACML 语言表示的规则、属性、授权角色等映射到网络安全管理的安全代理或者访问控制列表 ACL(Access Control List)，通过安全代理或 ACL 安全规则分发给网络中的相关安全设备。操作安全策略级发生冲突属形式冲突。

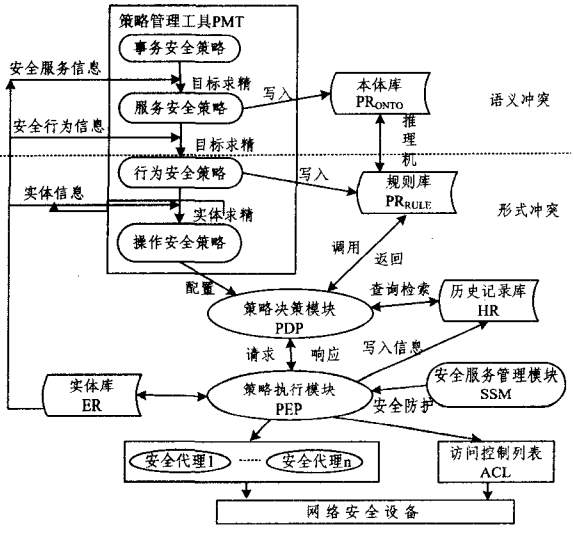

图 1　基于分层策略的安全级管理模型

基于策略的安全级管理模型采用分层不同的逻辑描述形式，扩展了 IETF/DMTF 定义的 PBNM，其实现单元除了 PBNM 定义的策略管理工具 PMT(Policy Management Tool)、策略仓库 PR(Policy Repository)、策略决策点 PDP(Policy Decision Point)、策略执行点 PEP(Policy Inforcement Point)和策略通信协议 PCP(Policy Communication Protocol)之外，还增加了安全服务管理模块 SSM(Security Service Management)、实体信息库 ER(Entity Repository)、安全代理平台 SAP(Security Agent Platform)和访问控制列表 ACL，其结构如图 1 所示。其中，PMT 负责完成安全策略创建及求精等；PR_{ONTO} 和 PR_{RULE} 分别存放和检索 PMT 生成的本体安全策略和规则安全策略；PDP 负责读取 PR_{ONTO} 和 PR_{RULE} 中的策略，并根据 ER 中的实体信息和 PEP 监控的系统运行时的状态信息进行策略决策；PEP 执行策略同时负责收集系统的事件、行为和对象等信息并保存到 ER 中，负责监控系统运行时的事件和状态等信息并发送到 PDP；SSM 负责安全服务管理，并按功能将安全服务划分为访问控制模块、认证服务模块、数据机密性服务模块等；ER 存储 PEP 可监控的实体信息，其中实体为所有 PEP 可监控收集的系统事件、状态、行为、对象及 SSM 提供的安全服务和安全行为等的统称；SAP 和 ACL 负责网络安全设备的具体策略或规则执行。

本文基于策略的安全级管理模型，提出一个集语义策略和形式策略为一体[17]的安全策略逐级求精方法。本方法首先通过语义目标求精，将抽象策略转化为系统应执行的安全行为；然后通过实体求精，确定安全行为的执行环境，从而将抽象策略转化为系统可理解和可执行的操作规则，解决了策略求精问题。本文提出的安全级管理模型是供大范围的网络管理系统用的，这些大系统包括很多自治管理的域，却没有一个集中的管理权威。在安全策略的逐级求精过程中，通过结合各安全级的认知情况，系统能评估各级安全策略行为之间的动态信任关系，有效完成安全策略的一致性检测和冲突消解，从而保护各级安全策略的有效执行。行为分析方法具有良好的可重用性和可扩展性[18]，只要预先定义好策略行为间的关系，就能自动地对策略进行冲突消解。

5　安全策略求精一致性自动检测

5.1　安全策略求精的流程结构

安全策略管理工具 SPMT(Security Policy Management Tool)为管理员提供了丰富的策略定义接口，可以接收管理员制定的高级事务安全策略。经过有效性和一致性检测后，将其转化为能够部署到网络中去的低级安全策略。安全策略求精一致性检测与冲突消解流程结构如图 2 所示，主要包含以下 5 个基本组成部分。

(1) 图形用户接口 GUI：实现安全策略编辑功能，由系统管理员或高级策略管理员通过其编辑高级安全事务策略。

(2) 安全策略转化逻辑和已存标准数据库：用于安全策略的转化，将高层安全策略转化为不同级别的低级安全策略。在网络安全防护系统内可以部署多级技术级安全策略。

(3) 策略分类算法：用来分解和区分不同级别的安全策略，形成多个安全策略集，如语义策略集和形式策略集。

(4) 安全策略一致性检测算法：负责保证较低级别安全策略在当前的网络拓扑和容量下是正确的、一致的、可行的。

(5) 安全策略分配器：提供安全策略部署接口，通过 LDAP 协议将各级安全策略写入相应的规则库和本体库，并按规定分发给安全代理或 ACL。

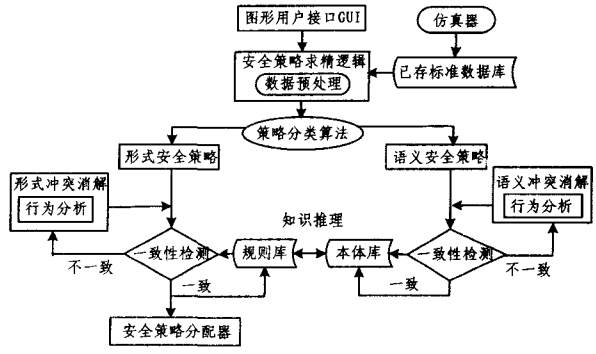

图 2　安全策略求精一致性检测与冲突消解流程结构

5.2 形式冲突一致性自动检测

安全策略一致性检查的目标是用最少代价检查整个安全策略库的一致性。由于形式冲突中的主体、客体、行为或环境条件存在某种逻辑关系，因此冲突检测的基本思路是判定安全策略 $P=\{P[1],P[2],P[3]\}=\{P[\text{subject}],P[\text{object}],P[\text{action}]\}$ 具有逻辑关系的两条策略中的行为标记是否相反。如相反，则构成冲突，也就是达成了 $P[\text{action}]+$、$P[\text{action}]-$ 冲突或 $P[\text{object}]+$、$P[\text{object}]-$ 冲突。还有一类冲突判定是如果肯定标记策略为 $P[\text{object}]+$ 而否定标记策略为 $P[\text{action}]-$，则构成 $P[\text{object}]+$、$P[\text{action}]-$ 策略冲突。判断 $P[\text{subject}]$ 冲突的算法如下。

输入：p_i，P，其中 p_i 为待检测的任一单条策略，P 为所有形式策略集合；

输出：P'，表示所有与 p_i 冲突的策略集合；

(1) get p_i[flag]
(2) for(j=0;j<P. count();j++)
(3) if P. p_j[flag]=! P. p_i[flag] and $p_i \in P$[action], $p_i \in P$[action] or $p_j \in P$[object], $p_i \in P$[object] then
(4) if P. p_j[subject]$\cap$P. p_i[subject]$\neq\varnothing$ then
(5) P'. add(P. p_j)
(6) endif
(7) endif
(8) if $p_i \in P$[action] and P. $p_j \in P$[object] and P. p_j[flag]=! p_i[flag] and p_i[flag]=false then
(9) if P. p_j[subject]$\cap p_i$[subject]$\neq\varnothing$ then
(10) P'. add(P. p_j)
(11) endif
(12) endif
(13) if $p_i \in P$[object] and $p_j \in P$[action] and P. p_j[flag]=! p_i[flag] and p_i[flag]=true then
(14) if P. p_j[subject]$\cap$P. p_i[subject]$\neq\varnothing$ then
(15) P'. add(P. p_j)
(16) endif
(17) endif
(18) endfor
(19) return P'

形式检测算法的核心就是判定具有实体重叠关系的两条策略中是否构成($P[\text{action}]+$，$P[\text{action}]-$)、($P[\text{object}]+$、$P[\text{object}]-$)或($P[\text{action}]-$，$P[\text{object}]+$)关系。如成立，则构成冲突。在算法中，只要遍历整个形式策略集合的规则库，就将所有策略规则与待定策略进行匹配判定。从复杂度来看，其计算时间与策略集合中元素个数相关，复杂度为 $O(n)$。实施判定的前提是要求安全策略级间的逻辑关系有明确的定义和行为关系。

5.3 语义冲突一致性自动检测

语义冲突规则根据具体的应用环境而定，难以用统一的规则进行描述，必须由人工干预进行检测，但可通过基于描述逻辑的发现无关规则间冲突的方法和在授权规则集合中检测不同类型冲突的方法来处理[19]。对语义安全策略 $P=\{P[1],P[2],P[3],P[4],P[5]\}=\{P[\text{subject}],P[\text{object}],P[\text{action}],P[\text{condition}],P[\text{sign}]\}$，其策略检测分析是基于主体 Subject、客体 Object、操作 Action、条件域 Condition 和策略授权符号 Sign 这五元组进行的。语义冲突通常指策略和策略的外部约束之间发生的冲突。根据各属性之间相互关系可划分为主体关联冲突、客体关联冲突、主客体关联冲突和主客体自关联冲突。针对其检测的方法有两种：基于 Tbox 推理的静态方法和基于 Abox 推理的动态检测方法。静态方法是指对策略的表示进行语法分析，以期发现冲突，与个体无关。动态方法是针对策略的外部条件对给出的 Abox 实例进行一致性检查，从而判断是否存在冲突。

5.3.1 静态冲突检测方法

输入：p_i，$p_i[]'$，P，其中 p_i 为待检测的任一单条策略，$p_i[]'$为主体、客体及主客体关联的任意一条策略，P 为所有语义策略集合；

输出：P_s'，表示所有与 p_i 冲突的主体关联策略集合；P_o'，表示所有与 p_i 冲突的客体关联策略集合；P_{so}'，表示所有与 p_i 冲突的主客体关联策略集合；

(1) get p_i[flag]
(2) for(j=0;j<P. count();j++)
(3) if P. p_j[flag]=! P. p_i[flag] and $p_j \in P$[action], $p_i \in P$[action] or $p_j \in P$[object], $p_i \in P$[object] then
(4) if P. $p_j[\,]' \cap$P. $p_i[\,]' \neq\varnothing$ then
(5) if P. p_j[subject]$\cap$P. p_i[subject]$\neq\varnothing$ then
(6) if P. p_j[object]$\cap$P. p_i[object]$\neq\varnothing$ then
(7) P'_{so}. add(P. p_j)
(8) else P_s'. add(P. p_j)
(9) endif
(10) else P_o'. add(P. p_j)
(11) endif
(12) endif
(13) endif
(14) endfor
(15) return P'

5.3.2 动态冲突检测方法

以描述逻辑为基础的动态冲突检测方法是基于 Abox 一致性推理的。在 Abox 中存在策略概念及其相关的个体。具体是在 Tbox 中加入约束条件结合 Abox 的一致性检测，从而查找是否存在应用相关策略冲突。如对于主客体自关联冲突，可以在描述策略时加入约束条件，在其中标明“不允许自操作”限制。引入 Abox 后，通过约束条件的不可满足，判断出 Abox 的不一致性，从而检测出存在自关联冲突。具体算法略。

6 基于安全级的策略冲突混合消解机制

6.1 一般冲突消解方法

目前，可用多种方法来解决网管系统中的策略冲突。其中最简单的方法就是改变策略的条件、动作等属性，使其不再产生冲突。但这种方法只能是在制定策略的时候使用，即预先检测冲突，这往往比较困难。而当冲突已经发生时，需要专门的消解方法，亦即对策略的信任和协商过程。根据产生式系统的控制策略法则，主要有“替代”、“最先匹配”、“优先级”、“优先权+匹配优先”、“多约束优先”、“元策略”等多种策略冲突消解方法。

6.2 基于安全级的行为分析和知识推理策略冲突消解方法

在策略求精冲突消解过程中，根据不同的冲突情况可以

采取多种解决措施。在各级策略及策略之间因为策略冲突的解决而存在有序关系，可以构成策略有向图。当大型网络系统安全策略数量很多时，基于有向图的策略冲突解决方法由于需要对大量策略进行计算处理而带来了很高的计算成本，本文的解决方法是在一定的条件下将策略大矩阵划分为不同安全级的策略小矩阵，再对各安全级小矩阵策略进行计算处理。

安全策略求精后各级分层安全策略都可能检测出冲突。本文将不同安全级的冲突分为语义冲突和形式冲突，提出语义冲突在层次的顶端，形式冲突在层次的底端。当发现冲突存在时，先从冲突层次底端开始寻求策略的冲突消解。如果低端不能有效解决冲突，则使用较高层消解策略冲突，直到顶端。

针对形式冲突：根据检测的结果，可以设置否定优先法、按策略规则优先权设置、按实体偏序关系优先、按系统规则优先等；也可用新行为重新解决相关的策略之间的冲突，就是抽象出一个新的动作代替以前发生冲突的动作。基于行为分析和知识推理的安全策略冲突消解方法如下：当检测到两个策略规则的条件集合有交集而它们指定的动作集合不相同时就发生了冲突，检测到策略冲突后可通过选择一个新的执行动作来解决。

针对语义冲突：最简单的方法是在创建策略时不对策略做检测和判定，而是在系统运行时，根据策略冲突的表现，对冲突进行处理。当系统检测到语义冲突时，应根据元策略判定策略的优先权，从而确定策略的执行，如中止系统运行或重新启动系统。

同时，形式策略和语义策略之间采用知识推理机制完成策略冲突的协商解决。

7 应用实现

根据策略求精安全级模型，设计出图 3 所示的访问控制策略冲突消解结构，其主要由冲突检测器、中央处理器和冲突消解器 3 部分组成。

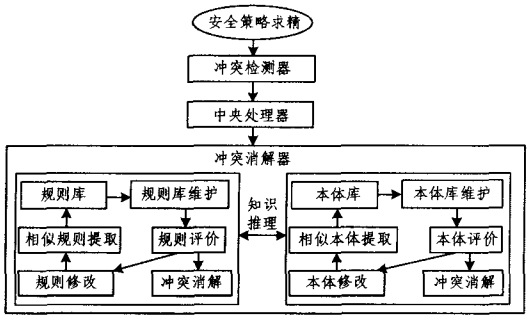

图 3 基于知识推理冲突消解系统设计

7.1 冲突检测器

冲突检测可通过以目标求精的方式自上而下分层管理，进行阶段检测的方法来实现。每个冲突对象描述为规则(本体)属性、方法。属性描述冲突发生时的环境状态，每个属性值都有相应的影响消解结果的权值。方法是针对相应属性的解决方案，其中包括该方法产生的原因。实现对象描述的代码如下：

```
Class CConflictmanagemnet
{
Public:
int Id_conflict;//冲突标志
int Type_conflict;//冲突类型
void ConflictDetect();//冲突检测
void ConflictAnalysis();//冲突分析
void ConflictResolution();//冲突消解
}
```

7.2 中央处理器

中央处理器用来计算冲突检测器中产生的冲突信息，将其进行分类处理，即求出冲突的对应规则值和本体值，用于进行冲突消解推理过程中的规则和本体匹配运算。

7.3 冲突消解器

实现冲突消解的关键之一就是建立一个合适的检索方式，检索的重点就是计算或估价冲突信息、规则库、本体库已有规则的匹配度，根据匹配度大小决定采用的规则和本体。

面向安全级的策略行为分析和知识推理的冲突消解过程可以归纳为以下步骤：

(1) 安全策略求精后进行一致性检查。如发生冲突，则产生消息，消息触发事件，将冲突信息送到中央处理器中进行计算，求出对应的规则或本体值。

(2) 从规则集或本体集中取出一条冲突解决规则。

(3) 将冲突与之匹配。由对应的规则库或本体库分析、比较和协商完成，即行为分析处理。

(4) 根据检测到的策略冲突，对其行为部分进行分析。若匹配成功，则执行这一冲突规则，进行冲突消解，否则执行(5)。

(5) 停止分析，进行知识推理，直到找不到匹配的冲突解决规则。如果是不关联行为，采用优先级方法消解策略冲突；如果有关联，则采用替代方法和元策略方法，利用知识推理、智能处理得到一个新行为，即新安全策略，再送入语义冲突检测模块，继续执行过程(1)，直到没有策略冲突为止，最终达到自动消解冲突的目的。从而避免出现网络系统安全策略执行冲突故障，提高网络系统的实时控制能力和安全可靠性操作能力。

结束语 安全策略求精一致性检测和冲突的解决，直接影响到 PBNSM 系统安全策略的正确执行以及安全管理的效率。本文在研究 PBNM 理论的基础上，提出了一种基于策略的网络安全分层管理实现方案。针对策略的一致性问题，将不同的安全策略冲突归为形式冲突和语义冲突两大类。并结合不同安全级的策略冲突给出了相应的检测算法和解决方法，较好地解决了安全策略求精的一致性问题。并根据不同安全级策略提出事件驱动的基于行为分析和知识推理的安全策略冲突消解方法，通过调整或重组安全策略动态改变系统运行时行为，大大增强了网络系统安全防护的灵活性和扩展性。

今后的工作主要集中在以下 3 个方面：一是研究非集中式的协作模型，对求精冲突进行分布式动态检测和智能消解；二是冲突检测和消解算法的进一步优化实现及性能提升；三是研究与 QoS 相关的策略冲突问题。

参 考 文 献

[1] 李祥军，孟洛明，焦利. 网管系统策略冲突解决的结果中存在的

问题及检测与解决方法[J]. 计算机研究与发展,2006,43(7):1297-1303

[2] Ono K, Nakamura Y, Satoh F, et al. Verifying the Consistency of Security Policies by Abstracting into Security Types[C]// 2007 IEEE International Conference on Web Services(ICWS 2007)

[3] Davy S, Jennings B, Strassner J. The policy continuum-Policy authoring and conflict analysis[J]. Computer Communications, 2008,31:2981-2995

[4] Dunlop N, Indulska J, Raymond K A. Dynamic Conflict Detection for Large Evolving Enterprises[C]// Proceedings of the Sixth International Conference on Enterprise Distributed Object Computing(EDOC 2002). Lausanne, Switzerland, 2002

[5] Damianou N, Dulay N, Lupu E C, et al. The ponder policy specification language[A]// The Workshop on Policies for Distributed Systems and Networks[C]. Bristol, UK, 2001

[6] Dulay N, Lupu E, Sloman M, et al. A policy deployment model for the ponder language[C]// IEEE/IFIP International Symposium on Integrated Network Management(IM'2001). Seattle, 2001

[7] 陈晓苏,林植,冯向东. 基于分层模型的网络安全策略逐级求精算法[J]. 小型微型计算机系统,2007,28(6):998-1002

[8] Lupu E, Sloman M. Conflicts in Policy-based Distributed Systems Management[J]. IEEE Transactions on Software Engineering-Special Issue on Inconsistency Management, 1999, 25(6):852-869

[9] Chomicki J, Lobo J, Naqvi S. A Logic Programming Approach to Conflict Resolution in Policy Management[C]// 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2000). Breckenridge, Colorado, USA, 2000

[10] Bandara A K, Lupu E C, Russo A. Using event calculus to formalise policy specification and analysis[C]// The 4th Int'l Workshop on Policies for Distributed Systems and Networks (POLICY'03). Villa, Gallia, Como, Italy, 2003

[11] Schaad A. Detecting conflicts in a role-based delegation model [C]// The 17th Annual Computer Security Applications Conf (ACSAC101). New Orleans, Louisiana, 2001

[12] Jajodia S, Samarati P, Subrahmanian V S. Alogical language for expressing authorizations[C]// Proc. the 1997 IEEE Symp. Security and Privacy. Oakland, CA, USA: IEEE Press, 1997:31-42

[13] Ye Chunxiao, Zhong Jiang, Feng Yong. Attribute-based access control policy specification language[J]. Journal of Southeast University(English Edition), 2008, 24(3):260-263

[14] 王雅哲,冯登国. 一种 XACML 规则冲突及冗余分析方法[J]. 计算机学报,2009,32(03):516-530

[15] Vangheluwe H, de Lara J. Meta-models are models too[C]// Proceedings of the 2002 Winter Simulation Conference. Enver Yucesan, Insead. New York, USA, 2002:597-605

[16] Marilleau N, Lang C, Chatonnay P, et al. A meta-model of group for urban mobility modeling[C]// Proceedings of the 2005 International Conference on Active Media Technology. Kagawa, Japan, 2005:397-400

[17] 李瑞轩,赵战西,文坤梅,等. 基于本体的多域访问控制策略集成研究[J]. 小型微型计算机系统,2007,28(9):1710-1714

[18] 李凤华,王巍,马建峰,等. 基于行为的访问控制模型及其行为管理[J]. 电子学报,2008,36(10):1881-1890

[19] 于海波,车海燕,金淳兆. 基于描述逻辑的 RB-RBAC 授权规则冲突检测方法[J]. 计算机科学,2006,33(10):101-105

(上接第 8 页)

特点利用不同的数据特征进行预测,具有一定的可扩展性。

通过引入事件序列频繁情节挖掘的相关方法为时序数据预测提供了一种新的解决思路,并给出了具体地利用均值特征和趋势特征的预测方法。这两种预测方法在多步预测的情况下预测精度均优于现有算法,而利用趋势特征的预测算法特别适用于数据随机扰动较大的情形。

结束语 准确掌握网络安全态势,是有效实施网络安全监管的必要前提。现有的网络安全态势分析系统处理能力有限,难以应用于国家骨干网等大规模网络。本文介绍了一个面向大规模网络的安全态势感知系统 YHSSAS,从感知模型、数据集成、关联分析、指标体系、事件预测几个方面介绍了 YHSSAS 的关键技术。部分技术已经在国家骨干网络的若干节点试用并取得良好效果。大规模网络安全态势感知还存在若干技术难题,如大规模网络安全事件数据的有效获取、海量安全事件数据的实时关联分析、客观且可理解的网络安全指标体系的建立等。另外,跨组织、跨机构的数据获取,也涉及管理方面的问题,还需要人们认识的提高和政策法规的支持。

参 考 文 献

[1] 王慧强,赖积保,等. 网络态势感知系统研究综述[J]. 计算机科学,2006,33(10):5-10

[2] Bass T, Gruber D. A glimpse into the future of id[EB/OL]. http://www.usenix.org/publications/login/1999-9/features/future.html, 1999

[3] Endsley M R. Toward a theory of situation awareness in dynamic systems[J]. Human Factors: The Journal of the Human Factors and Ergonomics Society, 1995, 37(1):32-64

[4] 刘效武,王慧强,等. 基于异质多传感器融合的网络安全态势感知模型[J]. 计算机科学,2008,35(8):69-73

[5] 赵国生,王慧强,等. 基于灰色 Verhulst 的网络安全态势感知模型[J]. 哈尔滨工业大学学报,2008,40(5):798-801

[6] 陈秀真,郑庆华,等. 层次化网络安全威胁态势量化评估方法[J]. 软件学报,2006,17(4):885-897

[7] Halevy A, Rajaraman A, Ordille J. Data Integration: The Teenage Years[C]// Proceedings of VLDB, 2006:9-16

[8] 李辉. 多层次入侵事件检测和关联方法研究[D]. 西安:西安交通大学,2003

[9] 王娟,张凤荔,等. 网络态势感知中的指标体系研究[J]. 计算机应用,2007,27(8):1907-1909

[10] 宣霍. 网络安全定量风险评估及预测技术研究[D]. 长沙:国防科学技术大学,2007