第 35 卷　第 1 期　　计　算　机　工　程　　2009 年 1 月
Vol.35　No.1　　Computer Engineering　　January 2009
·软件技术与数据库·　文章编号：1000—3428(2009)01—0040—03　文献标识码：A　中图分类号：TP311

智能安全防护软件策略构件的设计与实现

帅　飞，王晋东，王　坤，张恒巍
(解放军信息工程大学电子技术学院，郑州　450004)

摘　要：针对智能安全防护软件中策略的定制和管理问题，设计并实现软件的策略构件，采用基于用户最大满意度的策略选择算法解决策略选择时的策略缺失和策略冗余冲突。采用基于构件关联检索的一致性检测算法解决定制策略中的系统交互一致性冲突。结果证明策略构件能有效解决软件策略的定制和管理中的冲突问题。

关键词：智能安全防护软件；策略管理；冲突检测；冲突解决

Design and Implementation of Intelligent Security Defendable Software Policy Component

SHUAI Fei, WANG Jin-dong, WANG Kun, ZHANG Hen-wei
(Institute of Electronic Technology, PLA University of Information Engineering, Zhengzhou 450004)

【Abstract】 This paper presents a kind of security defendable software policy component, aiming at the problems in the customization and administration. The conflict between the policy lack and redundancy in the policy selection is solved by the algorithm based on the users' max satisfaction, while the consistency conflict in the policy customization is settled by the detecting algorithm of the associated component searching. It is proved that this policy component can solve the problems both in the customization and administration effectively.

【Key words】 intelligent security defendable software; policy management; conflict detection; conflict resolution

现代网络技术和信息技术的发展日新月异，窃密攻击方式和攻击手段越来越先进和多样化，对信息安全构成了新的威胁。智能安全防护软件可以通过对行为策略的控制，对软件进行重构，赋予其新的功能，从而增加软件部署的灵活性，提高了软件对环境的适应性。依据信息安全保障的需求，针对安全防护方面存在的薄弱环节开展智能软件理论及工程技术研究，探索新的安全防护模式和防护技术与信息系统的有机融合，对提高安全防护部署的灵活性和有效性、不断增强信息安全的综合保障能力、构建完善的信息安全保障体系有着重要的意义。

1　智能安全防护软件工作原理

智能安全防护软件采用了构件化的软件体系结构，因此，可以快速适应环境的变化。在最大化的安全软件结构框架下可以根据不同的目的，按不同的策略组装成不同的目标系统。另外，智能安全防护软件可以作为一个复合构件，集成到更大的系统中。其系统模型见图 1。

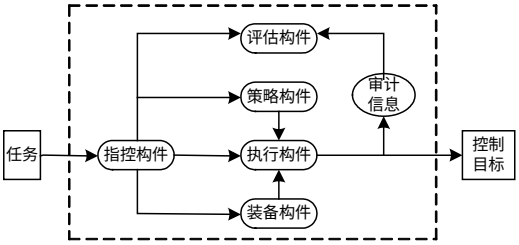

图 1　智能安全防护软件系统模型

如图 1 所示，软件系统主要包括指挥控制构件、执行构件、策略构件、装备构件、评估构件 5 部分。各部分功能描述如下：

(1)指控构件是软件的核心部件，负责与控制台进行信息交互，又可以在脱机时自主发出控制命令。

(2)执行构件是软件的关键部件。执行构件被激活后根据控制策略执行相关的任务。

(3)策略构件是软件的智能部件，能够根据指挥命令进行自主学习、策略更新，还可以向执行构件注入相应的控制策略。执行构件则根据相应的策略执行相关的任务。

(4)装备构件是软件的功能资源部件，特殊情况下在被指挥控制构件、装备构件激活后，向执行构件注入新的功能，实现执行构件的重构。

(5)评估构件是软件的分析部件，可以根据审计信息、主动风险巡逻和漏洞发现信息等，对系统资源进行机密性、完整性、可用性等的风险评估，按照标准确定相应风险等级，由指挥控制构件根据风险等级启动策略构件的策略更新。

在构件的组成上，按不同的任务角色划分成不同的构件，除指控构件外，其他每个构件都可以被激活和挂起。构件可以被重新构造，并且每个构件的重构不会对软件的结构产生影响，实现了软件结构稳定性和灵活性的有机结合。在指控构件的控制下，构件可以处于睡眠、唤醒、活动等不同状态，完成相关的安全防护任务，实现安全巡逻、信息监控、痕迹清除、端口封闭、事件审计、风险评估等功能。

2　策略构件的设计与实现

当前，在基于策略的管理方面有很多研究成果，除了由IETF提出的策略通用框架外，在Strongman系统中定义了一种

作者简介：帅　飞(1982－)，男，硕士研究生，主研方向：信息安全；王晋东，副教授；王　坤，博士；张恒巍，讲师、硕士
收稿日期：2008-06-07
E-mail：sf4018@163.com

可扩展的安全策略体系结构[1]，还有文献[2]提出的基于Ponder语言的策略部署框架等。Strongman系统的特点在于将多种安全机制通过构建相应的翻译器包容在一个信任管理框架下，并在统一的策略表示基础上实现一致性检验等功能。而Ponder策略部署框架是对IETF策略管理框架的扩展，采用统一的Ponder策略语言，策略描述明确，能方便地进行策略自动分发。其中，应用较广泛的是IETF的策略通用框架。

2.1 IETF 策略管理框架

IETF 提出了如图 2 所示的基于策略的管理的通用框架，包括 4 个组件：策略服务器(也称策略决策点)，策略管理工具，目录服务器(也称策略库)和策略执行点。策略管理工具提供图形用户接口，是管理员制订和管理策略的窗口；制订后的策略存储在策略库中；PDP 按照一定的要求从策略库中搜索、翻译并将策略送到指定位置。PDP 的功能还包括响应 PEP 的请求。PEP 根据所处环境具体执行策略。IETF 还为该框架建立了 PCIM 信息模型，并映射为 LDAP 模式，指定 LDAP 作为策略存储与访问的协议，为 PDP 与 PEP 之间的通信设计了 COPS 协议和 SNMP 协议。

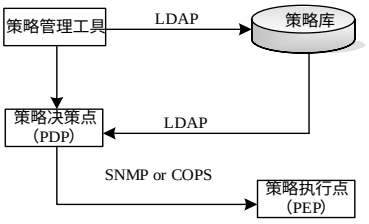

图 2 IETF 策略管理框架

2.2 策略描述

智能安全防护软件的策略按功能可分为安全策略、网络策略、设备控制策略、媒体文件策略、评估策略 5 大类。

(1)安全策略。包括软件的访问控制、根据用户身份分配权限、加解密构件、对文件进行透明加密。

(2)网络策略。对非法计算机或未安装客户端的计算机进行非法接入阻断，防止外来计算机随意接入内部网络，防止外界的恶意攻击对内部网络造成安全隐患。通过对受控终端进行 IP 地址和 MAC 地址的绑定，防止用户随意更改网络配置，及时发现未知(未登记)IP 地址、MAC 地址列表，自动发现有未知受控终端接入的交换机端口，方便系统管理员发现非法入侵者。

(3)设备控制策略。对移动存储设备、网络共享、软驱、光驱、打印机进行禁用/只读/自由使用的监控操作，并根据需要自动生成安全日志记录。

(4)媒体文件策略。媒体文件策略对各类文件的操作(新建、修改、删除)形成记录，以备审计。另外还可对文件的读写进行访问控制。

(5)评估策略。可以对整个系统进行安全评估，根据系统存在的风险等级确定策略的实施效果。

软件的执行策略可以是单个功能策略，也可由多个功能策略共同生成，分为静态策略和动态策略 2 类。静态策略根据专家的经验事先定义，存储在策略库中；动态策略是由用户通过策略控制台进行功能定制，然后动态生成的策略。动态生成的策略经过评估构件评估合格后可以添加进策略库，生成新的静态策略，完成策略构件的策略自学习。

采用构件化的软件结构使智能安全防护软件的执行策略定义更加简单灵活，执行策略的描述采用自然描述语言，更加通俗易懂，其策略描述如下：

Policy=<PolicyId><策略名><策略描述><策略类型><构件集合>

其中，PolicyId 是策略在策略库中的编号，是动态生成的，一般不做修改；策略名、策略描述、策略类型都由用户定义完成；构件集合即软件功能的集合，是策略的关键部分。例如，以下是 2 条静态执行策略实例：

<1><文件系统策略><对主机文件的操作进行记录><媒体文件策略><c2,c3 >

<2><主机安全防护><对主机的媒体文件、网路、设备进行监控，防止信息泄露><执行策略><c1,c2,c3,c4,c5>

其中，文件系统策略是一个单一的功能策略，主机安全防护策略则由多个功能策略组成。在策略库中，由多个功能策略组合而成的策略类型名用执行策略来命名。

2.3 策略构件的设计

策略构件是智能安全构件的智能部件，能够根据评估结果和指挥命令进行自主学习、策略更新。在指挥控制构件的控制下，向执行构件注入控制策略，执行构件会根据相应的策略执行相关的任务。

本文借鉴 IETF 策略管理框架，结合智能安全防护软件的特点，设计了如图 3 所示的策略构件结构。

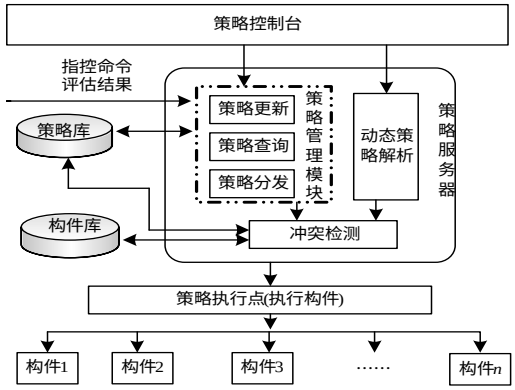

图 3 策略构件结构

策略构件主要由以下 4 个部分组成：

(1)策略控制台。为管理员提供操作接口。管理员可以通过该接口创建或配置策略。

(2)策略决策点。包含静态策略管理模块、动态策略解析、冲突检测 3 个模块。

1)动态策略解析。把动态生成的执行策略解析成面向机器的执行策略。

2)静态策略管理模块。实现执行策略的翻译、查询、更新、撤销和存储等管理功能，对软件策略进行有效的管理。主要包括策略更新、策略查询、策略撤销、策略解析 4 个模块。策略更新模块接收指控构件发送的命令，决定是否把现有策略添加到策略库，进行策略库的更新；策略查询按照不同的 PEP 对安全策略进行查询，以便管理中心及时了解各个 PEP 上的策略实施情况，方便管理；策略撤销是指当内网的状态发生变化、某些用户的身份或访问权限被撤销时，需要撤消相应用户的执行策略，也可根据指控命令撤销当前执行策略；策略解析把执行策略翻译成面向机器的执行策略。

3)冲突监测。常见的策略冲突主要分为策略冗余和策略缺失 2 种。当发生策略冗余时采用基于用户满意度的冲突消解算法，选择最优的策略。在产生策略缺失时，可以通过策略控制台定制新的执行策略。

(3)策略库。存储策略的数据库。

(4)策略执行点(执行构件)。接收策略服务器的策略，调用相应的构件执行软件策略。

2.4 策略冲突消解方法

当软件选择执行策略时会发生 3 种情况：(1)符合要求的策略只有一个；(2)符合要求的策略有多个；(3)没有符合要求的策略。当发生后 2 种情况时，系统就会产生策略冲突。产生策略冲突的原因是策略冗余和策略缺失。本文分 2 个方面来讨论。

(1)策略冗余冲突消解法(基于用户最大满意度的算法[3])

采用改进的暗标拍卖法[4]，引入用户关注度和应用重视度权值对策略标值的计算体系加以改进。针对不同用户的实际情况，通过设置用户关注度(*uai*)和应用重视度(*WQi*和*WEi*)权值，可以灵活地调整策略标值的计算体系。在该机制中，应用重视度权值*WQi*与策略提供的服务质量和应用需求的服务质量的差值相乘，其结果就是应用为此项服务质量付出的标值。当策略提供的服务质量大于应用需求的服务质量时，标值大于 0，反之则小于 0。各项服务质量指标的标值之和作为应用*A*为使用此策略愿付出的标值，记为*Q*。不同策略的执行代价不同，即执行时占用的系统资源不同，而不同应用对资源的开销要求也不同，所以，策略的执行代价也是策略选择时应该考虑的问题。与服务质量相反，希望执行代价尽可能小，所以，执行代价对策略选择的影响应与服务质量对策略选择的影响相反。因此，应用对资源消耗的重视度权值*WEi*与策略执行的资源消耗相乘的值为策略的执行代价标值，记为*E*。

不同的用户对各种应用的关注度也是不同的。用一个用户重视度函数来定义用户对不同应用的关注，记为权值*uai*($0<uai\leq 1$)，*ua* 与应用 *A* 为使用该策略愿付出的标值 *Q* 相乘，其值作为某用户 *U* 为应用 *A* 使用此策略愿付出的标值，记为 *B*($B=ua\times Q$)。系统计算某用户的所有应用为各个策略的投标值之和 *P*($M=B-E$)，获得最大 *P* 值的策略为选用策略。

(2)策略缺失冲突消解方法(基于构件关联检索算法)

当策略发生缺失时，用户通过策略控制台动态定制执行策略。由于装备构件库中的构件并非都是原子构件，因此一个构件的服务启动需要其他构件提供相关的服务。如果策略中只选择了构件 *A*，而当构件 *A* 启动时，需要构件 *B* 提供服务，否则构件 *A* 不能正常工作，这样系统的交互一致性产生冲突，系统不能正常工作。所以，必须在动态定制策略实施之前对其进行冲突检测，当发生冲突时，提示出错信息，用户根据信息进行策略的修改，保证系统正常执行。算法如下：

输入 执行策略的构件集合 $C = \{c1,c2,c3,\cdots,cn\}$;
For (i = 1; i <=n; i++)
// $Ci_{interface\_relationship}$ 表示构件Ci的关联构件集合接口；
If($Ci_{interface\_relationship} \supsetneq C$)
Messagebox(“构件 Ci 不满足交互一致性”);
Break;
输出 该策略不存在冲突，能正常执行。

3 结束语

本文提出的基于策略的智能安全防护软件能够方便地进行软件功能的定制，为每个用户提供个性化的产品和服务，以单件产品的制造方法满足顾客个性需求，有利于减少软件开发的时间和费用。不足之处是实现构件的粒度还比较细。下一步将研究构件组装和重构技术，进一步丰富和完善软件的功能。

参考文献

[1] Keromytis A D, Loannidis S, Greenwald M, et al. Sea1able Security Policy Mechanisms[R]. CIS Dept., University of Pennsylvania, Technical Rept: MS-CIS-01-05, 2001-01.

[2] Dulay N E, Sloman L M, Damianou N. A Policy Deployment Model for the Ponder Language[C]//Proceedings of the 7th IFIP/IEEE International Symposium on Intergrated Network Management: Intergrated Management Strategies for the New Millennium. Seattle, USA: [s. n.], 2001-05.

[3] Capra L, Emmerich W. CARISMA: Context-aware Reflective Middleware System[J]. IEEE Transactions on Software Engineering, 2003, 29(10): 929- 945.

[4] 张恒巍. 反射式信息安全中间件研究[D]. 郑州: 解放军信息工程大学, 2007.

（上接第 39 页）

参考文献

[1] Adams C, Lloyd S. Understanding Pubkic-key Infrastructure: Concepts, Standards and Deployment Considerations[M]. Indianapolis: Macmillan Technical Publishing, 1999.

[2] The Internet Society. Internet X.509, Public Key Infrastructure Certificate and CRL Profile[S]. RFC 2459, 1999.

[3] 李 新, 张振涛, 杨义先. 公钥证书撤消机制综述[J]. 通信学报, 2003, 24(9): 109-116.

[4] Micali S. Efficient Certificate Revocation[R]. MIT Laboratory for Computer Science, Technical Rept: TM-542b, 1996.

[5] Kocher P. On Certificate Revocaiton and Validation[J]. Financial Cryptography, 1998, 14(5): 172-177.

[6] Kocher P. A Quick Introduction to Certificate Revocation Trees(CRTs)[Z]. ValiCert, 1999.

[7] 范 磊, 许崇祥. 基于二叉树的证书撤销管理[J]. 计算机工程, 2002, 28(6): 33-34.

[8] Li Xin, Yang Yixian. The Analysis and Implementation of OCSP[J]. Computer Applications, 2002, 22(3): 7-9.

[9] Rivest R. Can We Eliminate Certificate Revocation Lists?[C]//Proc. of Financial Cryptography’98. [S. l.]: Springer-Verlag, 1998: 178-183.

[10] Daniel P, Rubbin A. A Response to “Can We Eliminate Certificate Revocation Lists?”[R]. AT&T Labs, Technical Rept: 99.8.1, 2000.

[11] Cooper D. A More Efficient Use of Delta-CRLs[C]//Proc. of IEEE Symposium on Security and Privacy. [S. l.]: IEEE Press, 2000: 190-202.

[12] Cooper D. A Model of Certificate Revocation[C]//Proc. of the 15th Annual Computer Security Applications Conference. [S. l.]: IEEE Press, 1999: 256-264.

[13] Adams C, Zuccherato R. A General, Flexible Approach to Certificate Revocation[EB/OL]. (1998-06-10). http://www.entrust.com/resourcecenter/pdf/certrev.pdf.

[14] 黄 曦, 邱 钊, 张 晋. 数字证书撤消列表发布机制分析比较[J]. 现代计算机: 专业版, 2007, (5): 37-39.