一种基于对象线索化RBAC的设计与实现

王建军[1,2],李新国[1],赵晋琴[1]
(1.湖南第一师范专科学校 信息系,湖南 长沙 410002;
2.国防科技大学 计算机学院,湖南 长沙 410002)

摘 要:安全内核中通常是使用一个模块实现所有的安全策略,对应用程序的独立性不强,不利于安全策略的改变。信息安全工具包的安全内核采用了线索化的正交软件体系结构,对每个子系统使用对象技术实现,也对象化了访问安全工具包内部信息的应用程序,使安全内核独立于外部应用程序。安全内核内部对象也相互独立,便于功能扩展和维护。安全内核实施了RBAC作为对象的安全访问控制机制,保证了对象化应用程序的安全性,也保证了工具包内部信息的安全性。

关键词:信息安全;安全内核;体系结构;对象技术;RBAC
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2006)08-0222-03

Design and Implementation of RBAC Based on Threaded Object

WANG Jian-jun[1,2], LI Xin-guo[1], ZHAO Jin-qin[1]
(1. Information Science and Technology Department, Hunan First Normal College, Changsha 410002, China;
2. College of Computer Science, National University of Defense Technology, Changsha 410002, China)

Abstract: Security policy was implemented by using a module in security kernel, and it was not benefit to change of security policy. Used threaded software architecture in security kernel of information security toolkit, used object technology to implement every sub-system, and objected the application which accessing information in security toolkit. Security kernel can not depend to exterior application, and has independence among the inside objects, convenient function to expand and maintain.

Key words: information security; security kernel; architecture; object technology; RBAC

1 信息安全工具包

信息安全工具包(Chinese Information Security Toolkit,简称CIST)是一个国产的开发平台,主要用于开发公共密钥基础设施(PKI)、CA或加解密应用等信息安全产品。CIST给外界提供一组API函数。用户通过API函数,能够简单、方便地完成PKI和加解密应用产品的开发。例如,用户要产生一个简单证书,则直接调用API函数cryptCreateCert (CRYPT_CERTTYPE_CERTIFICATE, &cryptCert, cryptUser),并设置参数:证书地址cryptCert,用户名cryptUser,以及证书类型CRYPT_CERTTYPE_CERTIFICATE等,就可以由CIST自动生成一个证书。CIST的主要功能是产生密钥和管理密钥(证书)、执行加解密算法以及网络安全连接等。对于用户,CIST内部犹如一个黑箱,不需要用户详细了解其内部执行任务的细节。CIST中有7个资源对象协调完成具体的工作,它们分别是:

收稿日期:2005-12-26
基金项目:湖南省教育厅研究项目(05C046);军队基础研究项目(JC03-06-008)
作者简介:王建军(1969-),男,湖南衡阳人,副教授,硕士研究生,研究方向为信息安全。

1)系统对象(System objects):是CIST系统中的一个核心对象,在CIST初始化时就要生成。其主要功能是产生随机数。随机数是为对象管理器对对象进行安全管理和上下文对象产生密钥服务的。

2)用户对象(User objects):在系统中创建的对象应该有一个属主(owner)——创建该对象的线程、进程或其它用户对象(该用户不同于API用户),对象的属主拥有对象的控制权。

3)上下文行为对象(Context objects):用来完成产生密钥、加解密和签名等工作。上下文对象经常附加于其它对象之后,上下文对象对用户来说是不可见的。当使用一个证书校验一个签名时,实际工作是附加在证书对象后的加密公钥上下文完成的,证书对象控制整个过程的完成。

4)信封对象(Envelop objects):是数据容器对象,它的行为可以由压入到它里面去的控制信息进行修改。例如,如果一个数字签名行为对象被作为控制信息加入到数据容器中,这时被压入到容器中的数据将被数字签名;如果一个口令属性被压入到一个容器中,这时被压入的数据将被加密。

5)会话对象(Session objects):是数据容器对象。与会话对象相关联的安全上下文对象经常在使用一个对等系统交换信息时建立,并且会话对象能够处理多个数据对象

(例如网络数据包)。

6)密钥集对象(KeySet objects):密钥集对象是密钥和证书的容器。包含一个或多个密钥或证书,也可能是证书撤消列表(CRL's)等。密钥集对象与用来保存密钥的机制紧密相联,这些机制有 PKCS#12、PKCS#15、包含证书的关系数据库、CRL's、LDAP 目录和 HTTP 证书链接等。

7)证书对象(Certificate objects):是一个安全属性容器对象。在证书对象中包含了一个属性集合,这些属性用于存放一个公钥或私钥或其它与证书相关的信息。安全属性容器的最基本的类型是公钥证书,它包含了公钥属性信息、证书主体与发布者的私钥信息。

因为 CIST 内部的工作与密钥有关,内部信息需要很高的安全强度,防止密钥被篡改或泄漏。为了满足 CIST 的安全需求,在其内部设置了一个安全内核。安全内核的功能主要有:

(1) 防止用户对 CIST 内部进行非法访问;

(2) 对资源对象之间的访问进行安全检查。

图 1 是安全内核执行安全监控的示意图。外部用户(USER)访问内部对象,而内部对象之间相互访问时,安全内核要执行与安全策略相关的检查。

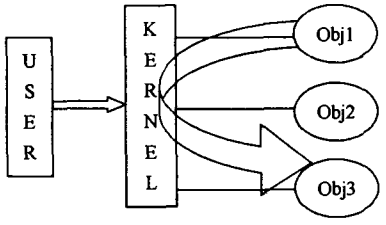

图 1　安全内核执行安全监控

2　RBAC

安全内核要实现对系统对象、用户对象、密钥集对象、上下文对象、证书对象、信封对象和安全会话对象的访问控制,以及对每一类对象的属性访问控制。访问控制的实现依赖于访问控制安全模型。

传统访问控制模型的缺点是:直接为每一个用户赋予一组许可权,在该过程中,通常还将具有相同职能的用户分成组,然后为每个组分配许可权。一旦主、客体的组织结构或系统的安全需求有所改变,可能导致进行大量烦琐的授权变动,使系统管理员的工作非常繁重[1]。

RBAC 是目前流行的一种访问控制机制。在 RBAC 中,每个角色与一个操作集合相关联,只有获得该角色授权的用户才能够执行相关联的操作集合。实现 RBAC 的一个优势是,当与管理的灵活性或应用行为不相冲突时,允许与角色相关联的操作尽可能的任意[2]。

与定义和修改一个角色相关联的可能行为有:

(1) 增加一个角色与其相关联的操作;

(2) 移动一个角色与其相关联的操作;

(3) 修改一个存在的角色;增加一个操作;移去一个操作;修改一个存在的操作。

RBAC 将访问控制权分配给角色,用户通过赋予不同的角色获得对象的访问权。优点:灵活、方便、安全。

在 RBAC 中,信息经常有基于一个固定的操作集合的应用访问。如果修改对一个应用有效的操作,可能会对一个已经存在了的应用产生冲突。移去一个操作或修改一个操作的语意会严重影响应用的功能和可能产生不可预期的结果。对于传统方式实现的 RBAC,要改变角色的访问控制权限,必须直接修改该角色的操作集合中的操作。根据以上分析,这种修改是不能随意进行的[3]。

3　正交软件体系结构

正交软件体系结构由组织层和线索的构件构成。层是由一组具有相同抽象级别的构件构成。线索是子系统的特例,它是由完成不同层次功能的构件组成(通过相互调用来关联),每一条线索完成整个系统中相对独立的一部分功能。每一条线索的实现与其他线索的实现无关或关联很少,在同一层中的构件之间是不存在相互调用的[4]。

正交软件体系结构的主要特征如下:

(1) 正交软件体系结构由完成不同功能的 $n(n>1)$ 个线索组成;

(2) 系统具有 $m(m>1)$ 个不同抽象级别的层;

(3) 线索之间是相互独立的(正交的);

(4) 系统有一个公共驱动层(一般为最高层)和公共数据结构(一般为最低层)。

在软件进化过程中,系统需求会不断发生变化。在正交软件体系结构中,因线索的正交性,每一个需求变动仅影响某一条线索,而不会涉及到其他线索。从而把软件需求的变动局部化,产生的影响也被限制在一定范围内。

4　RBAC 的实现

4.1　体系结构的设计

为了满足安全内核的需求,把它的具体结构确定为正交软件体系结构,把每个访问控制程序的执行权限分配给一个角色,再把角色与正交结构中的线索关联起来,既可以实现对访问控制程序的安全保护,实现对对象和属性的安全检查,又利于以后对安全内核的维护与改进。如果使用分层对象技术来设计 RBAC,那么既能够保留对角色管理的灵活性,又能够在改变角色的权限时,使对应用程序的冲突和影响最小化[5]。

使用分层对象技术设计 RBAC 的模型如图 2 所示。

方法对象中的方法是由访问对象信息的全体方法构成,并且保持固定。这些方法对应用程序是有效的。对方法对象的访问控制是由角色对象提供的,每一个角色对象提供一种访问控制。方法对象中的方法和角色对象中的方法形成一一对应的关系,而且它们具有相同的名字、类型和参数。对被方法对象访问的资源对象信息的访问控制,是专门在角色对象中定义的,而不包含在应用程序中。

角色对象中方法的主要部分被限制为如下内容：

(1)确定对该类角色对象访问的条件；

(2)作为接口对象和方法对象之间的过滤器。

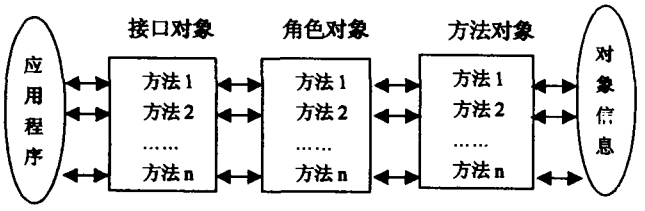

图 2 使用分层的对象技术实现 RBAC

如果允许对一个角色进行访问，则角色对象中的方法就调用方法对象中相应的方法。

接口对象中的方法与角色对象中的方法具有相同的名字、参数和类型。该对象中的方法给出与应用相关联的当前角色，调用角色对象中相应的方法。

由上述可知，在安全内核中，如果使用分层的对象技术实现 RBAC，既能够实现由访问控制程序执行的访问控制，又能够满足正交软件体系结构的思想，如果以后对安全内核进行扩充和修改，对已经使用了该安全内核的外部应用几乎不会产生影响。

图 3 表示 CIST 的安全内核是一个四层八线索的正交体系结构。

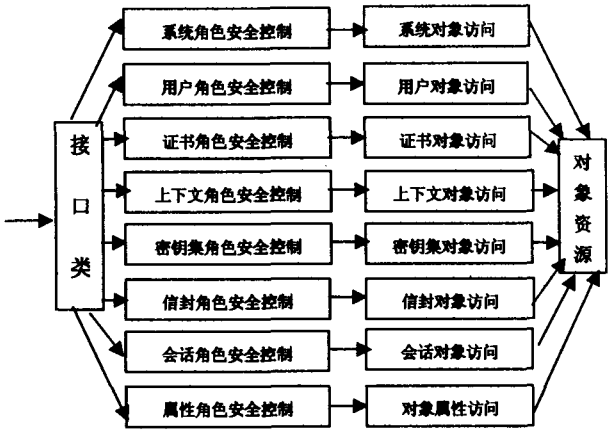

图 3 安全内核的正交体系结构图

4.2 对象的实现

在安全内核中，7 个对象资源和对象属性的基本访问方法如表 1 所示。

表 1 对象资源的访问方法

方法类型 / 对象资源	GetParamAcl	GetAttributeAcl	SecurityControl
系统对象	F	F	T
用户对象	F	F	T
证书对象	T	F	T
上下文对象	F	F	T
密钥集对象	F	F	T
信封对象	F	F	T
会话对象	F	F	T
对象属性	F	T	T

由表 1 可得，对对象资源进行访问的完整的方法集合为{GetParamAcl, getAttributeAcl, SecurityControl}，T 表示该方法对对应的对象是有效的，F 表示该方法在对象中被屏蔽，是无效的。其中 GetParamAcl()是在对证书进行签名和签名检查时，对参数 ACL 进行定位。GetAttributeAcl()是在对对象的属性进行访问时，用于对对象的属性进行定位。SecurityControl()用于对对象和对象属性进行访问时进行安全检查。每个对象都有一个 SecurityControl()，当安全检查通过后，就会调用与该对象相对应的对象处理函数，完成对对象的访问，如果是对属性进行访问，则在安全检查通过后，通过属性读、写函数完成对属性的操作。

方法对象的定义如图 4 所示。

```
class CMethod
{ public:
      SecurityControl ( ){调用对应的对象处理函数};
      GetAttributeAcl ( ){调用属性 ACL 定位函数};
      GetParamAcl ( ){调用参数 ACL 定位函数};
};
CMethod method;
```

图 4 方法对象的定义

角色对象共有 8 个。首先定义一个基本角色对象，8 个角色对象均由基本角色对象派生而来。

基本角色对象的定义如图 5 所示。

```
class CBaseRole
{public:
      virtual SecurityControl ( )=0;
      virtual GetAttributeAcl ( ) =0;
      virtual GetParamAcl ( )=0;
};
```

图 5 基本角色对象的定义

再定义 8 个具体的角色对象。以系统角色对象为例，如图 6 所示。

```
class CSystemRole:public CBaseRole
{public:
SecurityControl ( ){method. SecurityControl ( );};
GetAttributeAcl ( ){return("Don't permit access.")};
GetParamAcl ( ){return("Don't permit access.")};
};
```

图 6 系统角色对象的定义

其它角色对象的定义类似于系统角色对象的定义。

接口对象要完成两个任务：

(1)对需要的角色对象的定位；

(2)调用被选择的角色对象相应的方法；

接口对象的定义如图 7 所示。

```
class CInterface
{ public:
    SecurityControl ( ){1. 对角色进行定位；2. 调用角色对象中的 SecurityControl ( )};GetAttributeAcl ( ){ 1. 对角色进行定位；2. 调用角色对象中的 GetAttributeAcl ( ) };GetParamAcl ( ){ 1. 对角色进行定位；2. 调用角色对象中的 GetParamAcl ( )};
};
```

图 7 接口对象的定义

(下转第 227 页)

(CN)让认证中心AC验证。AC利用共享密钥Kac解密,确认CN的身份后,发送Kac(Ks)给CN让它得到临时会话密钥Ks。同样,AC还会发送Kah(CN,Ks)给HA,告诉HA通信节点CN已经通过认证,让HA发送MH的COA给CN。HA用Kah解密后,使用刚接收到的Ks发送Ks(COA,Km(CN))。CN用Ks解密后,向CA请求绑定路由表,把Km(CN)发送给MH,让MH知道这个CN的身份是经过认证的。至此认证工作结束,CN可以按照先前的优化路由方案发送数据包给MH了。

具体步骤如下所示:

(1) CN: Kac(CN)→AC

(2) AC: Kah(CN,Ks)→HA, Kac(Ks)→CN

(3) HA: Ks(COA,Km(CN))→CN

(4) CN: Km(CN)→MH

3.2 认证的安全性及相关讨论

当CN的身份被AC确认后,AC会让HA发送MH的COA给CN(HA对于MH的认证已经由移动IP协议所确保),CN完全可以相信这个MH的转交地址COA是真实可信的,从而相信这个MH的身份是已经确认的。而MH收到了用自己和HA之间的共享密钥加密的CN的身份信息,也可以据此相信CN的身份已经被家乡代理确认了,进而相信这是一个经过认证的CN。此外,通过时间戳可以防止不怀好意者通过截获Km(CN)来进行重发攻击;通过使用可信的AC分发的会话密钥Ks,使得CN可以确认HA的真实身份,防止了一个假冒的HA把伪造的MH的转交地址COA发给CN。

本方案并没有直接让移动主机MH对CN进行认证,原因在于:对于一个常规密钥系统,每一对MH和CN的双向认证都需要MH和AC有共享密钥,这使得密钥分发变得十分繁琐。而在本方案中,AC和MH之间分发密钥变成了AC和家乡代理HA之间分发密钥,而只要有这样一个密钥,HA就可以利用移动IP固有的HA和MH之间的共享密钥发送CN的身份认证信息给MH。换句话说,AC和HA之间的密钥代替了AC与HA所属的多个MH之间的密钥,一个密钥分发达到了多个密钥分发的认证效果,从而简化了密钥的分发复杂度,同时也有利于密钥的统一管理。

4 结束语

文中介绍了移动IP基本思想,根据使用CA的移动IP路由优化方法,提出了一种新的身份认证方案并对其进行了讨论。此方案利用了移动IP协议中HA对MH进行认证的特性,同时结合了引入CA的优化路由,简化了双向认证的复杂性,且具有较高安全性和可行性。当然,这种方案也存在一些问题,比如:方案的实施很大程度依赖家乡代理HA的可靠性,如果HA作假或者受到攻击变得不可靠,认证将面临新的挑战,这也是今后需要解决的问题。

参考文献:

[1] Perkins C. IP Mobility Support[S]. RFC2002. 1996.

[2] Perkins C. IP Encapsulation within IP[S]. RFC2003. 1996.

[3] Johnson D B, Perkins C. Route Optimization in Mobile IP[EB/OL]. draft-ietf-mobileip-optim-05.txt, 1996-10.

[4] Solomon J D. 移动IP[M]. 北京:机械工业出版社,2000.

[5] Stallings W. 密码编码学与网络安全:原理与实践(第2版)[M]. 北京:电子工业出版社,2001.

[6] 岑　巍. 关于移动IP的进一步研究[J]. 电信科学,1992(2):15-17.

[7] 朱　健,杨　庚. 移动IP路由方案研究[J]. 中国数据通信,2003(4):82-85.

[8] 蒋海林,张丽娟,谈振辉. Mobile IP路由优化协议中存在的一些问题[J]. 电力系统信息,2002(2):49-51.

(上接第224页)

对角色进行定位是先把主体(用户、线程、进程)的行为类型(消息类型)与角色对应起来,形成一个角色表,即,只有请求执行某种类型行为的主体才能够被分派相应的角色,此时,角色的分派不是直接与主体相关联,而是与主体的请求行为有关。事实上,在CIST中,一个主体能够请求执行的行为,已经在主体形成的时候与主体关联起来了。这是一个简单灵活的RBAC实现方式。

5 结束语

在CIST中,资源对象和对象属性只能由指定的程序访问,通过控制程序的访问权限,就可以控制对对象及其属性的安全访问。把安全内核的体系结构定义为正交体系结构,便于安全内核的功能扩展。把每个子系统定义为一个对象,便于子系统的改进。通过RBAC保证安全内核中每个对象的安全访问,既提高了系统的安全性,也加强了系统的独立性。由于没有增加程序执行过程的复杂度,所以不会降低系统效率。

参考文献:

[1] 訾小超,张绍莲,茅　兵,等. 访问控制技术的研究与进展[J]. 计算机科学,2001(7):26-28.

[2] Ferraiolo D F, Kuhn D R. Role-Based Access Controls[A]. Proceedings of the 15th NIST-National Computer Secuity Conference[C]. Baltimore, Maryland:[s.n.],1992.

[3] Ferraiolol D F, Cugini J A. Role-Based Access Control(RBAC): Features and Motivations[A]. 11th Annual Computer Security Appilcations Procceedings[C]. Maryland:[s.n.], 1995.

[4] 张友生. 几种新型软件体系结构[EB/OL]. URL: http://www.opentest.51.net/softeng/newstru.htm, 1999.

[5] Barkley J. Implementing Role-Based Access Control using Object Technology[A]. First ACM Workshop on Role-Based Access Control[C]. Gaithersburg, Maryland:[s.n.],1995.