

文章编号:1001-9081(2010)10-2632-04

开放网络环境中基于属性的通用访问控制框架

钟 将,侯素娟

(重庆大学 计算机学院,重庆 400030)

(20081402070@cqu.edu.cn)

摘 要:针对传统访问控制模型在新一代可信互联网环境应用中存在用户角色赋值效率不高、跨域访问控制实现困难等局限性,提出了基于属性的通用访问控制框架。该框架对用户、资源、操作和上下文四类对象的属性信息进行统一的描述和处理,简化了传统RBAC及其他访问控制系统复杂的权限判定方式,从而增强了访问控制系统的通用性和灵活性;同时,对于跨域的访问应用了基于属性证书的验证方式并给出了相应的策略评估方案和评估算法,能够针对不同应用域中用户的访问需求动态实施资源管理和访问控制;另外,框架中引入的运行上下文对象机制,进一步提升了该框架对复杂、动态互联网环境的适应能力。

关键词:开放网络环境;访问控制;属性;运行上下文;规则

中图分类号: TP393.08 **文献标志码:** A

Attribute-based universal access control framework in open network environment

ZHONG Jiang, HOU Su-juan

(College of Computer Science, Chongqing University, Chongqing 400030, China)

Abstract: Concerning the limitations of the application of traditional access control model in new generation credible Internet environment, such as the inefficiency in user-role assignment and the difficulty in cross-domain access control, a universal attribute-based access control framework was proposed. It took a unified method to dispose the attributes of users, resources, operations and running context, simplified the complex way of permissions determination in traditional RBAC and other access control modes, thus enhancing the versatility and flexibility of access control system. At the same time, authentication based on attribute certificates was applied in cross-domain access, policy evaluation and evaluation algorithm were also discussed, which could dynamically realize resource management and access control for users from different domains. In addition, the mechanism of the running context makes the framework more suitable to be applied in complex and dynamic Internet environment.

Key words: open network environment; access control; attribute; running context; rule

0 引言

在传统的访问控制中,基于角色的访问控制(Role-Based Access Control, RBAC)^[1-4]以其突出的优点使得系统管理员能够根据部门、企业安全政策的不同划分不同的角色,执行特定的任务,因此得到了广泛的应用。然而随着用户数目的膨胀,RBAC模型的角色分配和管理使得角色权限管理工作变得庞大且繁琐。另外,随着网络资源应用域范围的扩大,不同应用域之间的交互以及应用域内不同客户端和服务端端的交互愈加频繁,对于跨域的访问控制技术要求越来越高,有时候不同的安全域之间相互只知道对方的部分信息,从而使跨域的安全访问显得越来越迫切,传统的基于角色的访问控制已不能适应这种环境^[5-9]。为了便于管理,同时实现跨域的访问和资源共享,又可以最大限度保护用户的隐私,需要设计一种通用的基于对象属性的访问控制模型。

借鉴现有的RBAC访问控制模型,并在此基础上进行扩展,实现了基于属性的跨域的访问控制(Attribute-based Access Control, ABAC)。同其他基于属性的访问控制的描述方式相比,本模型所有的实体均采用统一的方式——属性来描述,从而使得访问控制判定功能在进行判定时,能够对访问控制判定依据采取统一方式处理。

访问控制实际上是当前用户对特定资源的某种操作合法性的判断问题。因此,建立支持开放网络环境和跨管理域应用的访问控制系统的核心问题是:访问控制系统如何识别来自其他应用系统中的用户,然后根据系统内的访问控制策略来判断用户的操作是否合法。为此本文提出将访问模型中的用户、资源和操作的属性集合都基于统一语义的属性集合来描述和定义。这样,用户的角色和所属的管理域都统一为用户的某种属性,方便系统管理员建立更为细致的访问控制策略。同时,通过引入运行上下文对象,使得匿名用户的访问控制成为可能。此外,为了便于用户实现跨管理域的资源查询、定位和操作,系统中的资源和操作也基于统一语义的属性进行描述。在系统运行过程中,属性是动态可变的量,而策略比较稳定,因此基于属性的访问控制策略方式可以很好地将属性管理和访问判定相分离^[10]。

1 ABAC模型

本文提出的ABAC模型中,访问控制系统是一个六元组:〈用户集,资源集,操作集,访问控制规则,属性集,运行上下文〉。ABAC模型是在RBAC的基础上扩展而来的,它更加符合人类的认知,能够在此基础上制定更加灵活的访问控制策略,为应用系统提供多粒度的权限管理。

收稿日期:2010-04-26;修回日期:2010-06-17。 基金项目:国家科技支撑计划项目(2008BAH37B04)。

作者简介:钟将(1974-),男,重庆江津人,副教授,主要研究方向:知识发现与知识管理、网络安全;侯素娟(1985-),女,山东武城人,硕士,主要研究方向:访问控制、网络安全。

同时该模型能有效解决 RBAC 模型中关于复杂角色条件下用户—角色—权限赋值的效率问题。本模型中角色是用户的一个属性,因此 RBAC 可以看成是本文所提 ABAC 模型的一个单属性特例。

1.1 模型的定义

定义 1 属性。将用户、资源、操作和运行上下文视为系统中的四类对象,分别用来表示访问控制的访问主体、访问客体、访问类型和资源请求时的情景信息,四类对象自身的特性被定义为属性 $A = \{Ua, Ra, Oa, RCa\}$, 可为访问控制提供细粒度的控制信息。关系如图 1 所示。

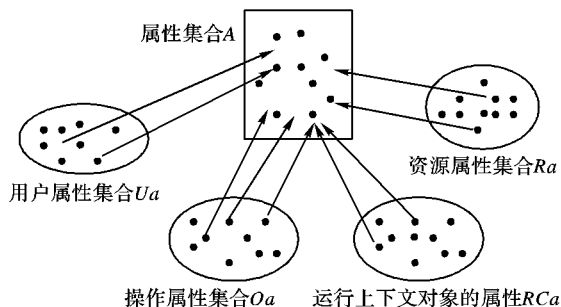


图 1 属性集合与四类对象的属性之间的关系

其中属性 A 由属性编号 A_{ID} , 属性名称 A_N , 属性类型 A_T , 属性值类型 A_{VT} , 属性值 A_V 组成。对于 $\forall A_i$, 满足 $A_i \in A$, 被分配唯一的属性编号 ($A_{idi} \in A_{ID}$), 同时有各自的属性名称 ($A_{ni} \in A_N$) 和属性类型 ($A_{ti} \in A_T$), 而每个属性值类型 ($A_{vti} \in A_{VT}$) 对应着一组属性值 ($A_{vi} \in A_V$), 可以表示为 $A_i ::= A_{idi} : A_{ni} : A_{ti} : A_{vi}$, 其中: $A_{vi} = \{A_{vi1}, A_{vi2}, \dots, A_{vij}, \dots, A_{vin}\}$ ($1 \leq j \leq n, n$ 为对应于 A_{vti} 的值的个数, 不同的 A_{vti} 对应的 n 的个数不同)。

对于某个实体 $e \in Ua \cup Ra \cup Oa \cup RCa$, 若 $\exists A_{idi} \in A_{ID}$, 那么 $\exists A_{ni} \in A_N, A_{ti} \in A_T, A_{vti} \in A_{VT}, A_{vi} \in A_V$, 此时称为对属性赋值, 对属性赋值完成的过程也是一个对象生成的过程。

各个对象的属性根据具体系统的应用预先定义, 由各个属性值集合 A_{vi} 的笛卡尔集的有意义的子集构成, 这四种属性的形式化定义如下:

$$\begin{aligned} A_{Ua} &\subseteq A_{Uav1} \times A_{Uav2} \times \dots \times A_{Uavi} \times \dots \times A_{Uavk} \\ A_{Ra} &\subseteq A_{Rav1} \times A_{Rav2} \times \dots \times A_{Ravi} \times \dots \times A_{Ravk} \\ A_{Oa} &\subseteq A_{Oav1} \times A_{Oav2} \times \dots \times A_{Oavi} \times \dots \times A_{Oavk} \\ A_{RCa} &\subseteq A_{RCav1} \times A_{RCav2} \times \dots \times A_{RCavi} \times \dots \times A_{RCavk} \end{aligned}$$

用户属性 A_{Ua} 用来描述访问主体 (即用户) 的安全特性, 如年龄、部门、职称等, 自然地, 可将角色视为用户的单一属性; 资源属性 A_{Ra} 用来对访问客体 (资源) 的特征进行描述, 如资源的类型、资源的数据结构、访问该资源所需费用等; 操作属性 A_{Oa} 用来描述访问类型, 如下载、删除、上传等; 上下文对象的属性 A_{RCa} 用于客户端执行访问请求时参与各方的上下文对象的具体描述信息, 包括客户端和服务端端的相关信息。

定义 2 用户。用户是指可以独立访问被保护数据或资源的一类对象, 它往往是提出请求或要求的发起者, 可以是用户, 也可以是任何发出访问请求的智能体, 包括进程、服务、程序等, 此处简化为用户, 用 $USER$ 表示一个用户集合, 用 $user$ 表示用户集合 $USER$ 中的一个用户, 即 $\exists user \in USER$ 。每个用户在系统中由一个唯一的 $user_ID$ 来标识。

定义 3 资源。资源是需要接受用户访问的一类对象, 包括所有受访问控制机制所保护的系统资源, 在不同应用场景下可以有着不同的具体定义: 比如在操作系统中可以是一段内存空间、磁盘上面的某个文件, 在数据库里可以是一个表中的某些记录, 在 Web 上可以是一个特定的页面, 也可以使网络结构中的某个广义上的数据包结构等。用 $resource$ 表示一个

资源集合, res 表示资源集合 $resource$ 中的某个资源, 即 $\exists res \in resource$, 每个资源在系统中有一个唯一的 res_ID 来标识。

定义 4 操作。操作是用来定义用户行为的一类对象, 它具体定义了用户对资源进行何种类型的访问。用 $operate$ 表示一个操作集合, $oper$ 表示操作集合中某个操作, 表示为 $\exists oper \in operate$, 同样的, 每个操作在系统中有一个唯一的 $oper_ID$ 来标识。

定义 5 运行上下文对象。为了使访问控制系统能够适应开放网络环境, 访问控制策略还需要根据系统运行上下文设定不同的访问控制策略。运行上下文对象记录了当前操作参与各方的一些动态属性, 例如当前用户的 IP、服务器当前的访问量和时间、网络的安全级别、CPU 的利用率等。它不依赖于某个特殊的用户和资源, 但往往会应用在访问控制策略中。

定义 6 属性表达式。属性表达式是某个一个属性变量的取值满足某种条件的运算符的三元组, 它不仅包含了属性与属性值的关系, 而且包含属性与属性之间的关系, 同时包含了属性值与属性值的关系。定义如下:

$$\begin{aligned} ae &::= \{Ua \mid Ra \mid Oa \mid RCa \mid Uav \mid Rav \mid Oav \mid RCav\} \text{ ropt} \\ &\quad \{Ua \mid Ra \mid Oa \mid RCa \mid Uav \mid Rav \mid Oav \mid RCav\} \\ ua &\in \{U_{a1}, U_{a2}, \dots, U_{ai}, \dots\} \\ ra &\in \{R_{a1}, R_{a2}, \dots, R_{ai}, \dots\} \\ oa &\in \{O_{a1}, O_{a2}, \dots, O_{ai}, \dots\} \\ rcav &\in \{>, <, =, >=, <=, \neq, \text{in}, \text{not in}, \text{between}\} \\ Uav &\subseteq A_{Uav1} \times A_{Uav2} \times \dots \times A_{Uavi} \times \dots \times A_{Uavk} \\ Rav &\subseteq A_{Rav1} \times A_{Rav2} \times \dots \times A_{Ravi} \times \dots \times A_{Ravk} \\ Oav &\subseteq A_{Oav1} \times A_{Oav2} \times \dots \times A_{Oavi} \times \dots \times A_{Oavk} \\ RCav &\subseteq A_{RCav1} \times A_{RCav2} \times \dots \times A_{RCavi} \times \dots \times A_{RCavk} \end{aligned}$$

其中: Ua 为用户属性变量; Ra 为资源属性变量; Oa 为操作属性变量; $RCav$ 为上下文对象的属性变量; $ropt$ 为关系表达式运算符, 其中运算符可以有多种: $>$, $<$, $=$, $>=$, $<=$, \neq , in , not in , between 等; Uav 为用户属性值或属性值的集合; Rav 为资源属性值或属性值的集合; Oav 为操作属性值或属性值的集合; $RCav$ 为上下文对象的具体属性值或属性值的集合。如在某一用户属性表达式 $\{\text{age} > 30\}$ 中, 用户属性变量为 age , “ $>$ ”是关系表达式运算符, 属性值为 30; 客户端执行请求时运行上下文对象客户端的 IP 地址为 202.202.1.3, 即 Client.IP = 202.202.1.3。

本模型的优势在于: 区别于一般访问控制模型中只有用户和资源才拥有属性表达式, 本模型中用户、资源、操作、运行上下文对象都具有属性表达式, 分别表示各自的属性满足一定条件的运算。

定义 7 访问控制规则。访问控制规则, 它是由用户、资源、操作和运行上下文四类基本对象的属性表达式 (ae) 通过 and , or 或 not 三种逻辑运算连接起来的表达式。定义如下:

$$\begin{aligned} Rule_i &::= [\text{not}] \{ae_1\} \text{ or } [\text{and}] [\text{not}] \{ae_i\} \dots \\ &\quad \text{or } [\text{and}] [\text{not}] \{ae_n\} \end{aligned}$$

其中, ae 是上述定义中涉及的属性表达式, not 为逻辑“非运算”, or 为逻辑“或运算”, and 为逻辑“与运算”。

定义 8 访问控制策略。本文将访问控制策略定义为一个访问控制规则的非空有限集合, 没有任何访问控制策略的空策略集合是没有意义的, 而由无限个访问控制规则组成的策略集合由于在判定上的不可终止性, 同样不予考虑。

访问控制策略是由一个或若干个访问控制规则构成的集合, 用 $Policy$ 表示, $\exists p_i \in Policy (i \in \mathbf{N})$ 。

$$p_i ::= \{Rule_1, Rule_2, \dots, Rule_i, \dots, Rule_n\}; n > 0$$

上述定义中任何一个访问控制规则都是访问控制策略集合中的一个元素,即 $\forall Rule_i, Rule_i \in p_i$ 。

定义9 允许算子 \oplus 。

$p_i \oplus p_j = p_i$ 或 p_j ; $i \in \mathbf{N}, j \in \mathbf{N}, p_i \in Policy, p_j \in Policy$ 。
 $(p_i \oplus p_j) \in Policy$, 它表示一条新的访问控制策略, 规定如果一个访问授权被 p_i 允许或被 p_j 允许, 那么它被 $p_i \oplus p_j$ 允许。

定义10 不允许算子 \odot 。

$\hat{p}_i \odot \hat{p}_j = \hat{p}_i$ 且 \hat{p}_j ; $i \in \mathbf{N}, j \in \mathbf{N}, p_i \in Policy, p_j \in Policy$ 。
 $(\hat{p}_i \odot \hat{p}_j)$ 表示一条新的访问控制策略, 该算子规则规定对于任意两条不被允许的策路 \hat{p}_i, \hat{p}_j , 应用了该算子后的策略仍然不被允许。

1.2 访问控制策略定义示例

例如, 在一个网上组播系统中, 包含了四类对象: 用户、资源、操作和运行上下文。其中用户的基本属性集合为 {用户类型, 年龄, 费用余额, 管理域}, 资源的属性有 {资源类型, 所需费用, 应用域}, 操作的属性类型有 {操作类型}, 运行上下文对象由具体的策略给出。前三类对象的属性及描述如图2所示。

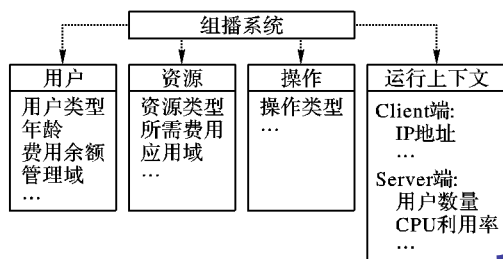


图2 网上组播系统中各类对象及其属性

其中以下属性为枚举值:

用户类型 = {管理员, 高级用户, 普通用户}

管理域 = {清华大学, 重庆大学, 西安交通大学}

资源类型 = { L_1, L_2, L_3 }

资源应用域 = {清华大学, 重庆大学, 西安交通大学}

操作类型 = {在线观看, 下载, 上传, 编辑, 删除}

系统为了实现以下的访问控制策略:

1) 普通会员通过付费的方式或者年龄大于13小于21的高级会员可以收看电影类型 L_2 的节目;

2) 年龄大于21的高级会员可以下载或在线观看所有类型的电影;

3) 管理员具有对所有的资源进行所有操作的权限;

4) 对于未注册用户, 只有访问客户端的IP为202开头且服务器的访问用户数量不足50人时才能在线观看 L_1 的资源;

5) 清华大学的管理员可以对清华大学、重庆大学和西安交通大学三所高校的 L_3 类型的资源进行删除。

则在系统中可以根据系统中各类对象的属性定义由以下访问控制规则构成的访问控制策略:

$Policy_1 ::= \{Rule_1, Rule_2, Rule_3, Rule_5\}$

$Policy_2 ::= \{Rule_4\}$

$Rule_1 ::= (user. 用户类型 = \text{“普通会员”} \text{ and } user. 费用余额 > res. 所需费用) \text{ or } (user. 用户类型 = \text{“高级会员”} \text{ and } user. 年龄 > 13 \text{ and } user. 年龄 < 21)) \text{ and } res. 资源类型 = \text{“}L_2\text{”} \text{ and } oper. 操作类型 = \text{“在线观看”}$

$Rule_2 ::= (user. 用户类型 = \text{“高级会员”} \text{ and } user. 年龄 > 21) \text{ and } res. 资源类型 \text{ in } (\text{“}L_1\text{”}, \text{“}L_2\text{”}, \text{“}L_3\text{”}) \text{ and } oper. 操作类型 \text{ in } (\text{“在线观看”}, \text{“下载”})$

$Rule_3 ::= (user. 用户类型 = \text{“管理员”} \text{ and } res. 资源类型 \text{ in } (\text{“}L_1\text{”}, \text{“}L_2\text{”}, \text{“}L_3\text{”}) \text{ and } oper. 操作类型 \text{ in } (\text{“在线观看”}, \text{“下载”}, \text{“上传”}, \text{“编辑”}, \text{“删除”})$

$Rule_4 ::= (Client. IP \text{ between } \{202. 0. 0. 0, 202. 255. 255\} \text{ and } Server. 用户数量 < 50) \text{ and } 资源类型 = \text{“}L_1\text{”} \text{ and } 操作 = \text{“在线观看”}$

$Rule_5 ::= (user. 管理域 = \text{“清华大学”} \text{ and } user. 用户类型 = \text{“管理员”}) \text{ and } (res. 应用域 \text{ in } \{\text{“清华大学”}, \text{“重庆大学”}, \text{“西安交通大学”}\} \text{ and } res. 资源类型 = \text{“}L_3\text{”}) \text{ and } oper. 操作类型 = \text{“删除”}$

$Policy_1, Policy_2$ 是分别针对注册用户和未注册用户定义的策略, 两者皆为允许规则集, 可以应用在不同的场合和情景中。

已经注册的用户在进行服务访问时, 系统根据 $Policy_1$ 作授权判定。对于未注册用户进行除 L_1 之外的资源访问请求, 系统将根据 $Policy_2$ 匹配访问控制规则中的 $Rule_4$, 由于只对 L_1 级别的资源系统在规则指定条件下将在线观看的权限授权给指定的开放用户, 系统将拒绝未注册用户对于其他类型的资源进行访问。

1.3 授权决策过程

由于在模型中将操作作为一类对象同用户、资源等其他对象统一地基于属性进行描述和策略授权, 因而不存在权限冲突的问题。但如果系统对同一个资源同时定义“允许策略集”和“不允许策略集”, 这就可能导致基于某个资源进行访问时存在策略冲突的问题。为此, 参照 XACML 组合算法^[11], 提供以下解决方法。

1) 允许覆盖: 依照允许算子 \oplus 对相关策略进行计算, 遇到有策略的返回值为 True 时结束。

2) 当1) 结束仍得不到返回值时, 按照不允许算子规则 \odot 的语义返回 False 值。

以上方法实质上依照了“允许策略”优先的原则, 只要访问请求符合一个相关的策略时允许访问, 只有所有相关策略均不满足时拒绝访问。用以下过程予以说明。

```

Input:  $\langle user, resc, oper, RunningContext \rangle$ 
/* 分别为用户标识、资源标识、操作标识和运行上下文对象 */
Output: True or False /* 请求被允许或拒绝 */
Begin:
    {Uav1, Uav2, ...} ← user
    {Rav1, Rav2, ...} ← resc
    {Oav1, Oav2, ...} ← oper
    {RCav1, RCav2, ...} ← RunningContext
    /* 根据输入获取相关属性值 */
    {p1, p2, ..., pn} ←  $\langle user, resc, oper, RunningContext \rangle$ 
    /* 获取相关策略 */
    /* 其中 Pi = {Rule1, Rule2, ..., Rulej, ..., Rulem} */
For i = 1 to n - 1
    k = 0
    For j = 1 to m - 1
        If (Rulej ⊕ Rulej+1) = True then
            Pi = True
        Else
            Rulej+1 = False
            k = k + 1
        If k = m - 1 then
            Pi = False
    If (Pi ⊕ Pi+1) = True then
        Return True
    Else
        Pi+1 = False

```



```

K = k + 1
If k = n - 1 then
    Return False
End

```

算法的主要思想基于用户、属性和策略存储的分布式性质,根据属性证书对属于不同域中的策略和属性进行查询,访问决策过程见图 3。

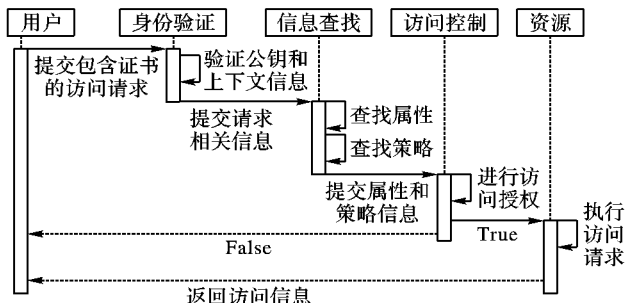


图3 ABAC的授权决策

2 系统的设计与实现

2.1 系统结构及基本工作过程

为了实现基于属性的访问控制系统,设计了如图 4 所示的系统框架,在框架中主要由证书权威机构、访问控制服务器、应用服务器、属性库和访问控制策略库等构成。

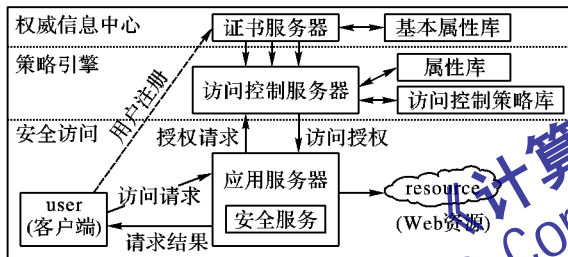


图4 基于属性的访问控制整体结构

其中证书服务器是存放用户基本信息的权威中心,是全局可信任的机构,负责签署公钥证书和用户属性证书的生成,并对证书的有效性负责,这些证书包含了用户的静态属性信息。考虑到用户的某些属性可能会发生动态的改变,例如用户的费用余额、信用等级等,因此在访问控制服务器中需要存储用户的动态属性值,访问控制服务器同样作为全局信任的机构,同样对属性证书的有效性负责。当用户向应用服务器请求资源时,应用服务器将用户标识、资源标识、操作标识以及系统运行的上下文发送给访问控制服务器,访问控制服务器根据访问控制策略来判断该访问是否合法,并将决策结果(允许或拒绝)返回,最终由应用服务器进行访问控制的实施。

在本系统的实施中,当用户提出资源访问请求,应用程序服务器在将用户标识、资源标识、操作标识以及系统运行的上下文发送给访问控制服务器时,访问控制服务器能够获取所有需要的属性信息(如果仍有需要的属性信息获取不到,则表示该属性在当前环境下不能获取),并且获取的每个属性的值都是真实、可信的,从而与用户的属性证书进行匹配,其中属性证书是由证书机构创建并签名的数据结构,包含了用户实体的属性信息,用于确定用户的访问请求是否合法。

系统基于以下全局公认的信任关系:

1)能够生成和验证属性证书的证书服务器是全局信任的机构,因为伪造的证书服务器可能生成伪造的属性证书和允许验证非法的用户属性和相关条件。

2)访问控制服务器同样是全局信任的,当基本访问策略

发生改变、新的资源建立或用户的信息发生动态改变时需要访问控制服务器作动态改变。

2.2 与应用系统的集成方式

由于不同的应用系统中对用户的属性、资源的描述,操作类型以及安全策略各不相同,因此上述数据通常由应用系统维护和管理。为了保证不同管理域中对用户的属性语义理解的一致性,建议各应用系统都从统一的属性集合中选择所需的属性来描述系统中的用户,为此本文建立了基本的用户属性集合供不同的应用系统选择。尽管描述资源的属性可以不同,但是考虑到跨管理域应用中方便用户查询和定位资源,建议不同的应用系统采用统一语义的属性集合的子集来描述资源。当应用系统中上述属性数据和访问控制策略建立之后,应用系统中对用户行为授权判断的操作原语来进行判断:

```
bool can_access(user, resc, oper, RunningContext)
```

其中参数分别表示用户的属性信息、资源属性信息、用户操作信息和当前系统的运行上下文对象。应用系统通过接口将上述参数发送给访问控制服务器,访问控制服务器将根据访问控制策略对当前用户行为的合法性进行判断。由于应用系统中保存了用户的属性信息,因此对域内用户进行判断是只需提供用户唯一标识即可。而对跨管理域的用户,需要包含用户的属性证书,同时系统运用了基于 X509 协议的密码机制来保证属性证书的真实性。

系统中各个模块以及模块之间的组合关系如图 5 所示。

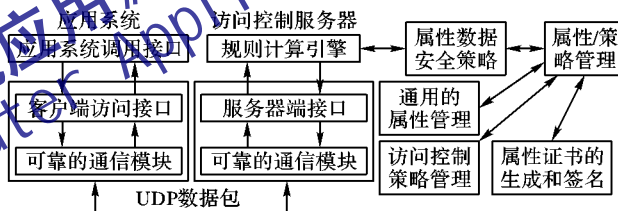


图5 各个模块之间的功能组合关系

3 与传统模型的比较

传统的访问控制的解决方案中应用最多的是基于角色的,本章给出本文模型与 RBAC 为代表的传统模型的比较^[12]。

1)ABAC 是 RBAC 的超集。ABAC 可以提供基于各类对象属性的授权策略,同样支持基于用户角色的授权和访问控制,角色在 ABAC 中仅仅是用户的一个单一属性。由此, RBAC 可以看做 ABAC 的单属性特例, ABAC 是 RBAC 的扩展。

2)ABAC 的应用范围更广。统一的语义描述使得模型的定义和策略控制更加方便和灵活,引入上下文运行对象的机制不仅使得匿名的访问成为可能,而且能够有效解决诸如“适当限制在上网高峰期的资源访问”问题。此外 ABAC 能够利用组合逻辑来组合规则和策略定义更复杂的策略为组织提供更灵活和强大的访问规则。RBAC 完全基于用户的角色进行授权和判定,没有考虑到用户的其他特性以及运行上下文和资源因素。

3)ABAC 支持动态属性的授权决策。RBAC 完全基于静态信息进行授权决策(因为角色本身是静态的),而且对于规模稍大的系统,用户—角色设置和角色—权限分配无疑对管理员是一个繁重的任务。而 ABAC 在授权决策中所基于的各类对象的属性可以是静态的,也可以是动态的,同时从上述给出的访问控制规则定义的例子中可以看出它能够消除静态角

(下转第 2640 页)

进行感知,并以此为依据对链路质量进行推理,将其作为路由选择的依据,实现对路由的优化选择,提高网络的吞吐量,达到负载均衡。仿真结果表明,在不对网络引入更多额外开销情况下,在分组成功递交率、分组传递平均端到端时延性能方面均有较大幅度的提升,能够从一定程度上达到负载均衡的路由效果。

下一步将针对多信道应用环境,结合信道分配策略,研究 LR-OLSR 协议的跨层联合优化问题,通过降低节点之间的相互干扰,提高无线资源的利用率。

参考文献:

- [1] WHITEHEAD P. Mesh networks: A new architecture for broadband wireless access systems [C]// RAWCON 2000: IEEE International Conference on Radio and Wireless. Washington, DC: IEEE, 2000: 43-46.
- [2] AKYILDIZ I F, WANG XUDONG, WANG WEILIN. Wireless mesh networks: a survey [J]. Computer Networks, 2005, 47(4): 445-487.
- [3] WAHARTE S, BOUTABA R, IRAQI Y, *et al.* Routing protocols in wireless mesh networks: challenges and design considerations [J]. Multimedia Tools and Applications, 2006, 29(3): 285-303.
- [4] LIU T, LIAO W. Capacity-aware routing in multi-channel multi-rate wireless mesh networks [C]// IEEE International Conference on Communications. Washington, DC: IEEE, 2006, 5: 1971-1976.
- [5] SONG WEN, FANG XUMING. Routing with congestion control and load balancing in wireless mesh networks [C]// ITST 2006: 6th International Conference on ITS Telecommunications Proceedings. Washington, DC: IEEE, 2006: 719-724.

(上接第2635页)

色的定义和管理,同时也能够消除用户的角色分配和角色的权限分配的需要和管理。

4) 复杂性对比。①某个应用域中用户有 m 个属性,每个属性有 k_i 个属性值,在 RBAC 模型中需要定义 $\prod_{i=1}^m k_i$ 个角色,而在本模型中只需定义 m 个属性,这将极大地降低系统管理员的工作量。②随着用户和资源数目的增长, RBAC 的规则数目呈指数级增长,而 ABAC 的规则呈线性增长。

4 结语

本文给出了一种基于统一属性描述的访问控制模型,本模型已经在国家科技支撑计划可信互联网项目中成功实现,为该项目中的安全组播、安全网络测量以及安全 BBS 系统提供了信任协商和动态访问控制的支持。

本文提出并实现的 ABAC 模型主要实现对用户某项具体操作的合法性判定的问题,该过程具有较高的效率。但是由于属性定义的灵活性和属性取值动态变化,因此查询用户具有的操作权限的效率可能比 RBAC 模型低。此外在开放的网络环境下使访问控制策略能够自动调整和演化也是下一步需要研究的问题。

参考文献:

- [1] SANDHU R S, COYNE E J, FEINSTEIN H L, *et al.* Role-based access control models [J]. IEEE Computer, 1996, 29(2): 38-47.
- [2] SANDHU R S, COYNE E J, FEINSTEIN H L, *et al.* Role-based access control: a multi-dimensional view [C]// Proceedings of 10th Annual Computer Security Applications Conference. Washington,

- [6] 魏翼飞,张勇,宋梅,等.一种适用于 WiFi Mesh 网络的 AODV 改进路由协议[J].北京邮电大学学报,2007,30(4):128-132.
- [7] 沈强,方旭明.无线 Mesh 网中一种基于综合准则的 DSR 扩展路由方法[J].电子学报,2007,35(4):785-790.
- [8] JACQUET P, MÜHLETHALER P, CLAUENSEN T, *et al.* Optimized link state routing protocol for Ad hoc networks [C]// IEEE INMIC 2001: IEEE International Proceedings in Multi Topic Conference. Washington, DC: IEEE, 2001: 62-68.
- [9] SHARMA S. P-OLSR: Position-based optimized link state [C]// 2009 IEEE 34th Conference on Local Computer Networks. Washington, DC: IEEE, 2009: 237-240.
- [10] CHRIQI A, OTROK H, ROBERT J M. SC-OLSR: Secure clustering-based OLSR model for Ad hoc networks [C]// 2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. Washington, DC: IEEE, 2009: 245-239.
- [11] YANG YANGLING, WANG JUN, KRAVETS R. Designing routing metrics for mesh networks [C]// Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh). Washington, DC: IEEE, 2005: 675-679.
- [12] SOBRINHO J L. Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet [J]. IEEE-ACM Transactions on Networking, 2002, 10(4): 541-550.
- [13] SOBRINHO J L. Network routing with path vector protocols: theory and applications [C]// Proceedings of ACM SIGCOMM 2003 Conference on Computer Communications. New York: ACM, 2003: 49-69.
- [14] The network simulator: NS-2 [EB/OL]. [2010-01-16]. <http://www.isi.edu/nsnam/ns>.

DC: IEEE, 1994: 54-62.

- [3] 黄益民,平玲娣,潘雪增.一种基于角色的访问控制扩展模型及其实现[J].计算机研究与发展,2003,40(10):1521-1528.
- [4] 严悍,张宏,许满武.基于角色的访问控制对象建模及实现[J].计算机学报,2000,23(10):1064-1071.
- [5] YUAN E, TONG J. Attributed Based Access Control (ABAC) for Web Services [C]// ICWS'05: IEEE International Conference on Web Services. Washington, DC: IEEE Computer Society, 2005: 561-569.
- [6] JOHNSTON W, MUDUMBAI S, THOMPSON M. Authorization and attribute certificates for widely distributed access control [C]// WETICE'98: Proceedings of the 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. Washington, DC: IEEE Computer Society, 1998: 340-345.
- [7] ZHANG XINWEN, LI YINGJIU, NALLA D. An attribute-based access matrix model [C]// Proceedings of the 2005 ACM Symposium on Applied Computing. New York: ACM, 2005: 359-363.
- [8] YE CHUNXIAO, ZHONG JIANG, FENG YONG. Attribute-based access control policy specification language [J]. Journal of South-east University: English Edition, 2008, 24(3): 260-263.
- [9] 叶春晓,吴中福,符云清,等.基于属性的扩展委托模型[J].计算机研究与发展,2006,43(6):1050-1055.
- [10] 李晓峰,冯登国.基于属性的访问控制模型[J].通信学报,2008,29(4):91-93.
- [11] MAZZOLENI P, CRISPO B, SIVASUBRAMANIAN S, *et al.* XAC-ML policy integration algorithms [J]. ACM Transactions on Information and Systems Security, 2008, 11(1): 1-26.
- [12] 沈海波,洪帆.基于策略的 Web 服务访问控制研究[J].计算机科学,2007,34(5):107-110.