一种面向 SSL VPN 的新型应用层访问控制模型*)

夏 涛 周敬利 余胜生 欧阳凯
(华中科技大学计算机学院 武汉 430074)

摘 要 利用虚拟私有网(VPN: Virtual Private Network)来实现安全跨越 Internet 访问远端服务群的技术是目前网络安全研究的一个重要组成部分。但是,由于虚拟私有网的隧道技术能够绕过防火墙,使得基于 VPN 服务器攻击内部服务群成为可能。因此,本文提出了一种面向 VPN 的新型访问控制模型——应用层集中式信息访问控制模型。它综合了目前主流访问控制模型的控制特点、反病毒和入侵检测的工作机制。并针对 VPN 通信流的特点,将访问控制与 VPN 隧道、转发机制紧耦合,从而增强网络安全性。同时,本文给出了该模型的一个实现原型。

关键词 虚拟私有网,访问控制,应用层,隧道

A Novel Application-layer Based Access Control Model for SSL VPN

XIA Tao ZHOU Jing-Li YU Sheng-Sheng OUYANG Kai
(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract The use of VPN to securely access the remote servers through Internet is one important technology in the current network security research. However, the tunneling technology of VPN makes it possible to bypass the control of firewall and compromise interior servers based on VPN server. Thus, this paper puts forth the Application-layer based Centralized Information Access Control Model, a new access control model for VPN. It integrates the features of the current mainstream access control models and the working mechanism of anti-virus and intrusion detection. On the basis of VPN communication stream, it also tightly couples access control with VPN tunnel and transmission mechanism to enhance network security. This paper also provides a prototype for the model.

Keywords Virtual private network, Access control, Application-layer, Tunneling

1 引言

由于 Internet 已成为主流的低成本通讯构架,大量研究机构和公司利用公用网建立安全可靠的私有网(VPN: Virtual Private Network)[1]。VPN 技术通过安全协议(比如 IPSec[2]、TLS/SSL: Transport Layer Security /Secure Socket Layer[3])实现了数据的保密性、消息完整性和端点认证。但是,由于 VPN 服务器的隧道技术(tunneling),使得恶意用户能够利用 VPN 服务器绕过防火墙的访问控制,达到攻击 VPN 内部服务群的目的。因此,面向 VPN 服务器的访问控制是实现高安全性 VPN 的关键技术之一。

访问控制技术广泛地应用于操作系统、数据库、网络和分布式系统中。传统而言,访问控制技术的研究可以归为:访问控制模型/策略的研究和安全策略框架体系的研究。针对访问控制模型,其经典模型包括:B-LP 模型[4]、Biba 模型[4]、RBAC(Role Based Access Control)模型[5]以及 CDAC(Content Dependent Access Control)模型[6]等。对于安全策略框架而言,1999 年,Ray Spencer 等人提出的 Flask 安全体系揭开了现代动态安全策略框架研究的序幕[7];2002 年,Elisa Bertino 等人提出了一种用于可扩展访问控制系统的语言 MACS (Multipolicy Access Control System)[8],从语言规范的角度描述了多种安全策略共存的解决方案。

近年来,针对 VPN 的访问控制策略研究越来越为人们所重视。J. Jason 等人提出了基于 IP 层的策略信息配置模型 (Configuration Policy Information Model)[9]。但是,目前面向 VPN 的安全访问控制策略主要是基于 IP 层的研究,并没有基于应用层的访问控制研究。相对于 IP 层的研究,基于应用层的访问控制具有如下特点。

细粒度:不仅能够完成现有的 IP 层访问控制功能,更能做到对多种信息的综合控制(比如:同时考虑用户名、来源、目的端以及时间等因素),以及对特定应用层协议的解析控制(比如:文本和上下文分析)。

防止网络病毒和入侵攻击:通过对网络病毒特征和 IDS (Intrusion Detection System)规则建立相应约束,可以防止其对内部服务群造成伤害(比如:根据 HTTP 协议的文本分析,通过切断传输路径来防止恶意用户利用内部 Web 服务器的弱点进行攻击。如果是基于 IP 层,一旦恶意用户能够访问内部 Web 服务器,则 IP 层的访问控制无法阻止其对服务器的入侵行为)。

因此,本文提出基于应用层的集中式信息访问控制(ACIAC: Application-layer based Centralized Information Access Control)模型。ACIAC 模型是基于对 VPN 通信的分析,通过建立控制信息中心,对访问控制所需资源进行集中式管理,利用约束和事件来实现访问控制。

2 ACIAC 模型要素

本节结合集合和关系理论,通过对 ACIAC 模型的术语、规则以及控制等方面的论述,阐明 ACIAC 模型的组织要素。

*)国家自然科学基金(批准号:60373088)资助项目。夏 涛 博士,讲师,主要研究领域为网络安全技术、计算机存储;周敬利 教授,博士生导师,研究方向为高性能网络存储技术、多媒体计算技术;余胜生 教授,博士生导师,研究方向为数字信号处理、网络存储技术;欧阳凯 博士研究生,研究方向为安全操作系统、网络安全技术。

2.1 术语定义

定义 1 流(Stream) 是指经过 VPN 服务器的所有通信信息,分为接入流(AS: Access Stream)和数据流(DS: Data Stream)。AS 是指在用户接入 VPN 或者建立隧道过程中,VPN 服务器所处理的信息;DS 是指一个隧道建立后,在该隧道内传输的数据信息。任意流都隶属于一个特定的用户,通过 $u(S/AS/DS)$ 表示。通过对 AS 和 DS 的分类控制,能够做到细粒度的访问控制,既能通过 AS 对内部服务器接入进行控制,也能通过 DS 对内部服务器的内容进行控制。

定义 2 集/链

(1)信息集$\{Info\}$:是任意一次 VPN 流所涉及的所有属性集合。

(2)约束集$\{Cond\}$:是任意一次 VPN 流中涉及的所有静态访问控制集合。

(3)事件集$\{Event\}$:是特指 ACIAC 范畴的访问控制集,通过该控制集实现动态管理机制,比如:约束集的创建、更新以及销毁。

(4)任一约束所涉及的属性:分为主体集$\{Sub\}$和客体集$\{Obj\}$,$\{Sub\}$代表流传输的动态信息,$\{Obj\}$代表流传输的静态信息。

(5)链 $List(x)$:是 ACIAC 模型管理具有特定逻辑关系的信息、约束和事件等对象集的数据结构抽象描述。

(6)用户集 U:描述了向 VPN 注册的所有用户,而原始角色集 R 描述了可以执行相同等级权限操作的一群用户。经过授权的用户以 $A(u)$ 表示。此外,ACIAC 中定义了一个特殊的用户,即管理员 $Admin$,管理员能够对信息做管理操作。

(7)动作集$\{Action\}$:是指 ACIAC 依据约束执行的结果采取的控制动作集合;在 ACIAC 模型里,定义了允许 *allow*、丢弃 *discard*、过滤 *filter* 和悬滞 *pend*。

allow 表示允许流经过 VPN 服务器到达内部服务器或者客户端。

discard 是指由于访问权限使得流不能通过 VPN 服务器,从而导致中断该次隧道或接入。

filter 是指由于流携带了非法内容或者特定服务内部需求限制,使得流不能通过 VPN 服务器,但是对该次隧道或接入并没有影响。

pend 特指在某次流中约束不具备执行的条件,而被调度器忽略执行的情况;该值不具备控制的依据。

定义 3 信息元素的形式表达 采用$\langle name, value, List(R, Access), List(U, Access)\rangle$对描述。*name* 指明信息元素名;*value* 指明了信息元素值的抽象描述,其具体含义由 *name* 定义的时候指定;*Access* 指明了对信息元素的无权限/读权限/写权限/拥有权限的控制权限集 $Access=\{none, read, write, own\}$;$List(R, Access)$指明了该元素的角色群以及访问权限;$List(U, Access)$指明了该元素的用户群以及访问权限。没有显示指明的角色或是用户(管理员除外)对该元素都没有任何访问权限。

定义 4 等级 $level(x)$的判定$\{r\in R\}\times\{u\in U\}$ ACIAC 模型采用用户/角色混合模式界定等级。假设属于角色 $r1$ 的用户 $u1$ 想访问信息元素 $info1$,在满足下述两种情况时,则允许该次访问:

a)$r1$ 有足够的权限访问 $info1$,并且 $info1$ 并没有显式地指明 $u1$ 的权限是 *none*;

b)$r1$ 没有足够的权限访问 $info1$,但是 $info1$ 显式地指明 $u1$ 的权限,并且不是 *none*。

否则,其它任意情况,ACIAC 将拒绝该次访问。

定义 5 一次流传输过程的形式表达式是:

$$\{\forall u\in A(u), u(List(s))\subset\{Sub\}\}\xrightarrow{\substack{u(List(c))\subset\{Cond\}\\ u(List(e))\subset\{Event\}}}\{u(List(o))\subset\{Obj\}\}$$

当一个约束/事件注册时,依据其所需的信息登记为主体链 $List(s)$和客体链 $List(o)$以及约束/事件路径。在一次流传输过程中,ACIAC 根据流 $u(S)$属性,调用其约束/事件链 $List(c)/List(e)$中每一个约束/事件的路径。

2.2 规则定义

规则 1 所有流的控制(约束/事件)都必须在授权用户状态下进行。

$$\{\forall u\in A(u), \forall s\in u(S), \forall c\in u(List(c))\cup\forall e\in u(List(e))\}\Rightarrow Exec(c\cup e)$$

规则 2 变更任意流 $u(S)$的信息集或信息元素值,必须满足 $u(S)$的权限不小于信息元素的初始等级。

$$\{\forall s\in u(S), \forall i\in\{Info\}, level(s)\geqslant level(i)\}\Rightarrow Modify(i)$$

规则 3 同一个约束可以有多个实例,但是一个用户只能有一个约束的一个实例。

$$\{\forall u\in A(u), \forall c\in u(List(c))\}\Rightarrow\{u(Entity(c))\equiv 1, Entity(c)\geqslant 1\}$$

规则 4 事件集由 ACIAC 模型集中管理,在 VPN 授权用户(即验证用户成功)后,ACIAC 将满足该用户等级条件的事件集关联,形成该用户的事件链;事件链的事件会被属于该用户的流 $u(S)$触发。用户并不拥有事件实例,只是引用事件。

$$\{\forall u\in A(u), \forall s\in u(S)\}\xrightarrow{triger}\{\forall e\in u(List(e))\}$$

$$\{\forall e\in\{Event\}\}\Rightarrow\{Entity(e)\equiv 1\}$$

规则 5 在用户授权的状态下,其约束链由通过该用户的事件链管理。

$$\{\forall u\in A(u), \forall e\in u(List(e))\}\xrightarrow{e}Manage(u(List(c)))$$

规则 6 在 VPN 授权用户后,用户为约束链和事件链的主体集维护了一个信息副本,以$\langle name, value\rangle$对表示。在不同用户空间,同样的主体元素名的副本值 *value* 也不尽相同。所有用户的客体集均由 ACIAC 全局维护。在不同用户空间,客体集的值是一样的。

$$\{\forall u1\in A(u), \forall u2\in A(u), \forall s1\in u1(List(s)), \forall s2\in u2(List(s)), name(s1)=name(s2)\}\Rightarrow value(s1)=value(s2)$$

$$\{\forall u1\in A(u), \forall u2\in A(u), \forall o1\in u1(List(o)), \forall o2\in u2(List(o)), name(o1)=name(o2)\}\Rightarrow value(o1)\equiv value(o2)$$

规则 7 在执行授权用户约束链的过程中,任意约束的结果为 *discard* 或 *filter*,则无需执行后续约束;否则必须执行完约束链。只有在约束链中所有约束都不为空的情况下,流传输才被允许。

$$\{\forall u\in A(u)\xrightarrow{\forall c\in u(List(c))}\forall action\notin\{discard, filter\}\}\Rightarrow Pass(u(S))$$

2.3 控制描述

结合 VPN 实际工作的流程,以用户一次使用 VPN 为例,模型管理从逻辑上基本可以分为:用户接入、隧道建立、数据流控制、用户退出。

用户接入:在 VPN 验证用户成功后,触发 ACIAC 生成有效 IDA(u)(有效 ID 是由 ACIAC 动态产生的全局唯一且不可逆的值),并建立事件的引用关系(即 $List(e)$),对 $List(e)$中所有事件的引用计数器加 1;最后试图执行该事件链。通常在用户接入的时候,ACIAC 至少执行初始化事件(一般是第

一事件):构建该用户的约束链 $List(c)$,建立约束和事件的主体元素副本,并重新确定对象副本元素的值。

隧道建立:在用户接入 VPN 后,用户在对 VPN 服务群中某一服务访问时,需要建立隧道。通常 ACIAC 首先依据隧道建立请求携带的有效 ID,鉴别请求的合法性;其次遍历执行该用户的 $List(e)$,并依据规则 6 执行 $List(c)$。

数据流控制:在一个隧道建立后,ACIAC 对随后客户端和内部服务端之间的通信进行控制。对数据流的控制在于对数据文本以及上下文关系的分析,这和典型的 CDAC 模型以及 CBAC 模型类似。我们的目标是针对 Web 服务构建数据流的基本框架(比如:对特定 URL 的过滤、对敏感文本的分析);通过 Web 数据流控制框架,可以方便地扩展到其他类型应用层服务的数据流控制(比如病毒防范、攻击分析)。显而易见,数据流控制是通过对每个数据包的分析提高安全性,其代价是 VPN 服务器的负载增大。

用户退出:在用户退出 VPN 系统时,ACIAC 遍历 $List(e)$,每执行一次事件,对该事件的引用计数器减 1。$List(e)$ 中的最后一个事件通常是销毁事件:清理该用户相关信息以及相应的各种资源。

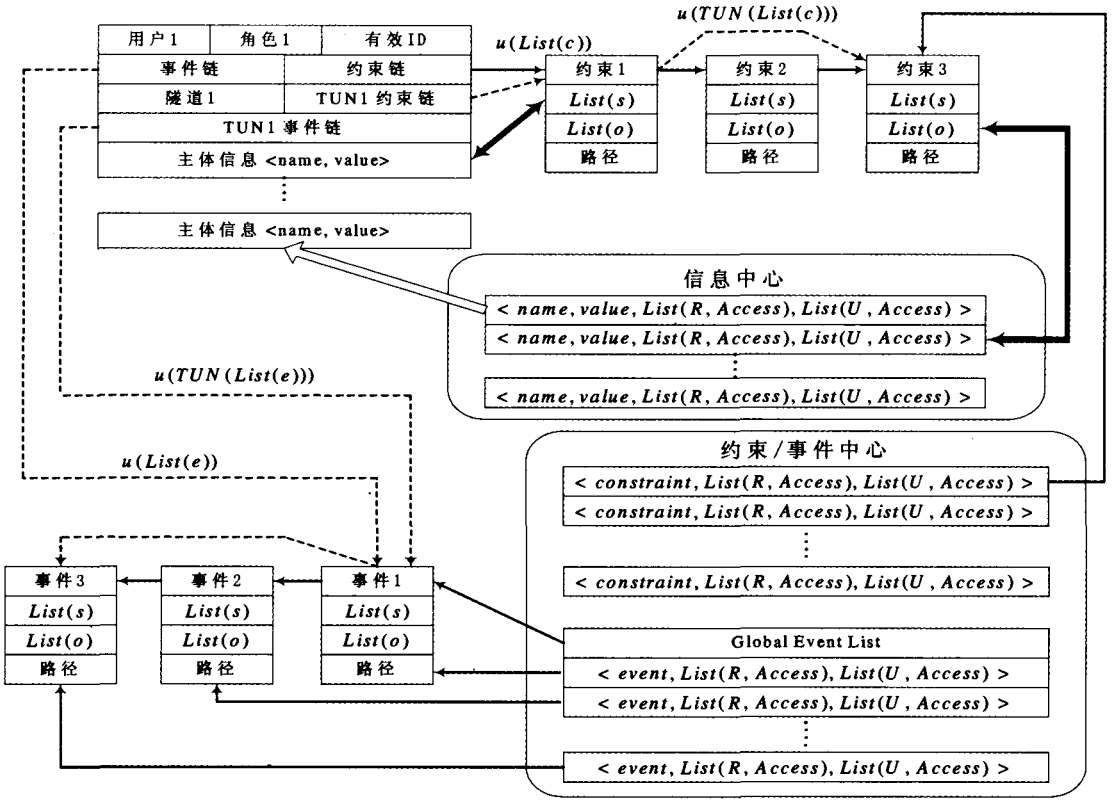


图 1 ACIAC 模型示例

图 1 展示了信息、主体链、客体链、约束链、事件链与用户/角色交互的关系(所有的操作都必须在前面讨论的规则控制下进行)。显而易见,在 ACIAC 中有两个中心:信息中心和约束/事件中心。无论主体元素还是客体元素都来自信息中心。在 ACIAC 为认证授权用户 $u1$ 创建有效 ID 后,$u1$ 将其所需的主体对象从信息中心复制到自己的私有空间,并重新设置值。当一个约束或事件需要客体元素时,ACIAC 从信息中心中获取该元素,并赋给相应的约束/事件。在约束/事件中心,ACIAC 以〈$constraint$, $List(R, Access)$, $List(U, Access)$〉和〈$event$, $List(R, Access)$, $List(U, Access)$〉管理约束/事件,同时 ACIAC 也为事件集维护一个全局链。在 $u1$ 初始化过程中,ACIAC 为 $u1$ 创建一个引用事件链 $u(List(e))$($Event1$ 和 $Event2$);然后 $u1$ 通过初始化事件创建自己的约束链 $u(List(c))$($Cons1$、$Cons2$ 和 $Cons3$)。

当用户 $u1$ 创建一个隧道时,ACIAC 能够根据约束/事件的属性优化该用户的约束/事件链,以引用链的形式存在,如图 1 的 $u(TUN(List(c)))$和 $u(TUN(List(e)))$所示。其目的是避免不必要的约束/事件控制,节约控制开销。

2.4 模型论述

ACIAC 模型是结合 VPN 通信特点和当前常用的网络访问控制模型各自优点的产物。通过前面的论述,我们不难看出,ACIAC 模型既具有 RBAC/UBAC 模型的控制特点,也具有 CDAC/CBAC 模型的控制特点。

我们将 ACIAC 模型的特性归纳如下:

a)集中式对象管理。通过将现实世界资源形式化为信息元素以及将访问控制抽象为约束/事件的模式,ACIAC 利用两个中心来对以上资源进行集中式管理,从而达到管理的便捷性以及避免对象的二义性。比如:两个约束 $Cons1$ 和 $Cons2$ 都需要客体信息元素 $Info1$ 来决定访问路径;当管理员修改了 $Info1$,由于 $Cons1$ 和 $Cons2$ 都是通过信息中心获得该信息的值,因此对 $Info1$ 的一次修改能同时反馈给上述两个约束,从而避免了 $Info1$ 的二义性。

b)基于流的控制。根据定义 1,我们知道 ACIAC 将 VPN 通信信息分为控制流 AS 和数据流 DS。这样划分的目的是实现细粒度控制,避免不必要的控制操作。显然,任何访问控制的操作都会降低 VPN 服务器的性能。假设:在一个隧道里,一次流传输的平均耗时是 T_{trans},一次访问控制操作(一次事件或约束)的平均耗时是 $T_{control}$,而目前访问控制的数目是 Num。如果没有访问控制,则一次传输的总耗时是:

$$T_{total} = T_{trans}$$

如果访问控制模型没有区别流的属性，则一次传输的总耗时是：

$$T_{total}=T_{trans}+Num\times T_{control}$$

通过我们对 VPN 访问控制的分析，访问控制可以分为面向 *AS* 和面向 *DS* 两类。因此，将访问控制归类可以避免不必要的访问控制操作开销。我们可以得 *AS* 出的控制数目是 $Num(AS(List(e))+AS(List(c)))$，*DS* 的控制数目是 $Num(DS(List(e))+DS(List(c)))$，而两者之和才与 *Num* 相同。ACIAC 中一次传输的总耗时是：

$$T_{AStotal}=T_{trans}+Num(AS(List(e))+AS(List(c)))\times T_{control}$$

或 $$T_{DStotal}=T_{trans}+Num(DS(List(e))+DS(List(c)))\times T_{control}$$

因此，ACIAC 可以降低访问控制对 VPN 服务器性能的影响。

c)基于用户/角色的混杂等级机制。由于用户等级单一，使得管理过于庞大，复杂度增大。而单一的角色等级又无法满足一些特殊用户的特殊要求，因此 ACIAC 采用 U/RBAC 混杂等级机制。比如：假设现有角色 *r*1 和 *r*2，属于角色 *r*1 的用户 *u*1 和 *u*2，以及两个信息元素 *info*1 和 *info*2，这两个信息元素只能被角色 *r*2 访问。现在 *u*1 需要访问 *info*1，但是我们并不希望 *r*1 的其他用户能够访问 *info*1(比如 *u*2)，也不希望 *u*1 能够访问到 *r*2 能够访问的其它元素(比如 *info*2)。显而易见，单独的角色控制或者用户控制都很难满足这种需求。而根据 ACIAC 的定义 4，我们能够显式地指明 *u*1 对 *info*1 的访问权限，从而实现上述需求。

d)基于事件的有效用户管理机制。ACIAC 模型利用事件驱动，隐式地完成授权用户的状态变迁。

$$\forall u\in A(u)\xrightarrow{\forall e\in u(List(e))}Change(u(i),u(c)),\forall i\in u(List(i)),\forall c\in u(List(c))$$

发生某事件将导致该有效用户的信息副本以及约束链发生变动。在 ACIAC 模型中，有效用户的状态相对比较简单 $status=\{init,update,terminal\}$，*init* 表示刚生成有效用户，ACIAC 将对其进行初始化操作；*update* 表示在用户使用 VPN 过程中，由于别的事件触发该有效用户的信息副本或者约束链的变动；*terminal* 表示在用户退出 VPN 时，ACIAC 对其资源的清理操作。

3 ACIAC 原型

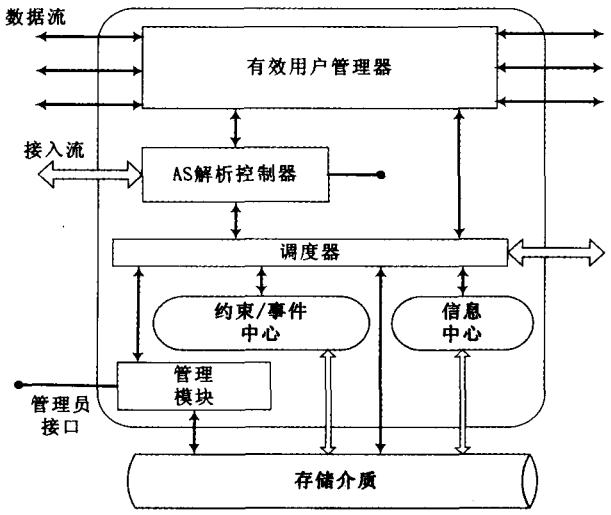


图 2 ACIAC 原型结构

基于前面对 ACIAC 模型要素的论述，本文也给出了我们的 ACIAC 实现原型。本节将详细论述该原型的组成模块及其逻辑关系。

如图 2 所示，ACIAC 原型的模块包括：调度器(Scheduler)、接入流解析控制器(ASPC：Access Stream Parse Controller)、有效用户管理器(VUM：Valid User Manager)以及管理模块(AM：Administrant Module)。以上模块协同工作，以实现 ACIAC 对 VPN 的访问控制。

调度器：

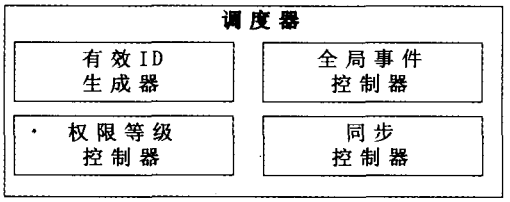


图 3 调度器的功能组织

ACIAC 调度器的核心功能组织包括：有效 ID 生成器、全局事件控制器、权限等级控制器以及同步控制器，如图 3 所示。在 VPN 验证用户成功后，ACIAC 调用有效 ID 生成器为该用户生成有效 ID。该有效 ID 是以当前时间以及一个随机数为种子，通过无冲突的散列函数生成。通过图 1，我们知道 ACIAC 有一个全局事件链，有效用户在权限等级控制下能够组成自己的事件引用链。全局事件控制器实现了全局事件链的管理，为新用户生成 $u(List(e))$。权限控制器则实现了定义 4 描述的混杂等级机制。由于 ACIAC 原型是个多线程/多进程的系统，调度器利用同步控制器实现 ACIAC 共享资源的一致性和完整性。

AS 解析控制器：

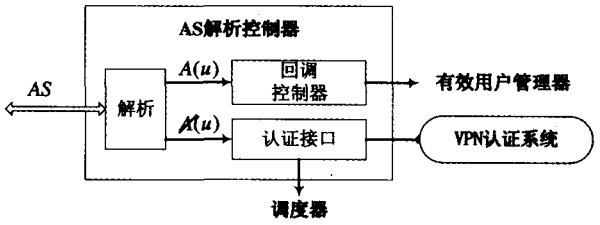


图 4 ASPC 的基本流程

ASPC 的功能相对简单，其基本流程如图 4 所示。ASPC 将 *AS* 区分为待认证 *AS* 和认知 *AS*。待认证 *AS* 是指 ACIAC 不能识 *S* 所属的用户，即 *AS* 中没有携带有效 ID(通常发生在用户接入 VPN 的时候)。因此，ASPC 通过认证接口通知 VPN 认证系统，认证该用户(认证接口同时实现了对待认证 *AS* 过滤控制的功能，以防止利用待认证 *AS* 对 VPN 认证服务的攻击)。在认证成功后，ASPC 再通知调度器为该用户分配新的有效 ID，并完成初始化工作。认知 *AS* 是指携带有效 ID 的接入流，ASPC 利用回调控制器通知 VUM 处理。VUM 定位该用户，通过该用户的 $u(List(c))$ 和 $u(List(e))$ 控制该次 *AS*。

有效用户管理器：VUM 的内部组织关系如图 5 所示，VUM 以用户实体的形式管理每个有效用户。每个用户实体拥有自己的约束链 $u(List(c))$、事件链 $u(List(e))$、主体链 $u(List(Sub))$、隧道控制器以及客体高速缓冲区。VUM 中的隧道控制器不同于 VPN 的隧道技术，该隧道控制器的目的是为数据流提供访问控制机制。VUM 能够通过该控制器优

化访问控制机制，避免不必要的访问控制操作。

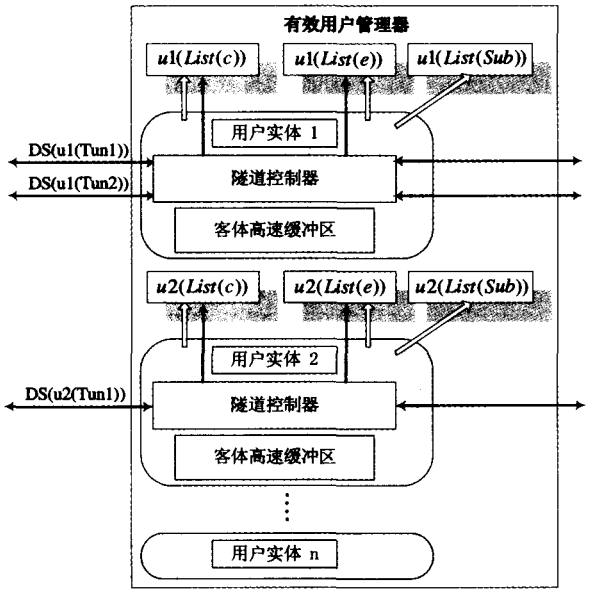


图 5　VUM 内部组织

客体高速缓冲区是 VUM 为 ACIAC 提供的另一个优化机制。我们知道，相对于 VPN 范畴的客体集{*Obj*}，属于特定用户的客体集 *u*({*Obj*})是非常有限的，并且特定用户需要访问的资源也是非常有限的。因此，VUM 利用客体高速缓冲区存放该用户最近使用的客体元素对〈*name*，*value*〉(类似于主体元素副本形式)，从而避免每次访问控制都需要从信息中心获取客体元素。

管理模块：

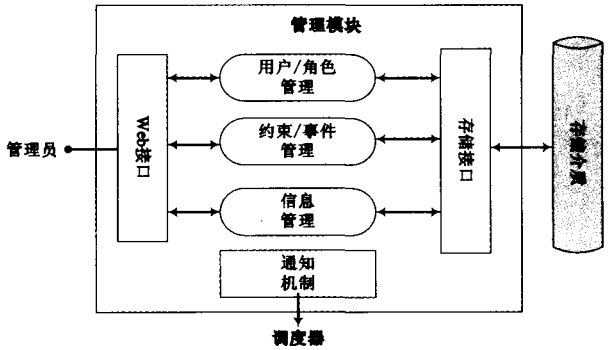


图 6　AM 内部组织

AM 的内部组织包括：Web 接口、用户/角色管理、约束/事件管理、信息管理、通知机制以及存储接口，如图 6 所示。AM 的 Web 接口为管理员提供了基于 Web 浏览器的管理机制，用于管理资源(用户/角色、约束/事件和信息)的注册/注销和修改。用户/角色、约束/事件和信息管理为 AM 提供了上述资源的内存管理结构。而存储接口则实现了上述资源在存储介质与内存存在形式上转换的功能。ACIAC 原型采用 XML (Extensible Markup Language)技术实现上述资源的管理。任何资源的变更操作会触发 AM 的通知机制通知调度器。调度器根据当前情况同步变更操作，保证所有资源在不同模块中的一致性。

4　相关研究及分析

网络访问控制技术最广泛的定义是实现所有网络安全的终极目标—— 在合适的情况下允许访问，在不合适的情况下拒绝访问。最近几年，针对 VPN 的访问控制研究越来越多。本节将对相关研究进行分析，并与 ACIAC 模型进行对比。

Xin Guo 等人在 ICCT2003 会议上提出面向 IP 层 VPN 的策略管理系统[10]，其中心思想是利用基于策略的网络管理(PBNM：Policy-Based Network Management)技术在 IP 层实现对 VPN 的访问控制。该策略管理系统包含 4 个逻辑模块：策略管理工具、策略仓库、策略决策以及策略执行。Xin Guo 等人也为该管理系统提供了一套策略定义语言，便于管理员操作。与此同时，Rongsheng Shan 等人也在 ICCT2003 会议中对大型 VPN 的网络安全策略进行了探讨，提出一种新型信任域和安全传输模式等技术来解决大型 VPN 中不同信任域之间策略冲突的问题[11]。其基本思想是：将 VPN 划分成不同的信任子域 *E*，每个子域有自己的策略；当一个目标对象从一个域 E_1 流向另一个域 E_2 时，该对象原有的策略会被 E_2 的策略代替。Rongsheng Shan 等人通过定义安全级别(security-level)、信任度(trust-degree)、网络流(net-flow)以及安全传输(security transmission)来构成信任域的基本要素，并通过规范安全传输需求以及网络安全策略来实现这种新型信任域。上述研究都是基于 IP 层对 VPN 访问控制的探讨，焦点在于 IP 层 VPN 的安全策略框架设计。

另一方面，目前任何单一的访问控制模型(比如：U/RBAC、CDAC 和 CBAC)都无法胜任 VPN 控制的所有需求。我们认为通过动态安全策略框架技术将现有访问控制模型结合起来，满足 VPN 的需求，在实际工作中至少面临两个方面的问题：

a)各种模型具体约束控制的制定方式不尽相同，必然为管理带来更多的困难。首先，每种模型都必须依据动态安全策略框架技术提供的接口注册，并且提供抽象控制函数和数据结构以便策略框架管理每种模型。其次，每种模型对信息对象的管理、存储方式不尽相同；而同一个信息对象可能被不同的模型使用，如果该信息的属性需要改动，必须对所有使用到该信息的模型进行数据更新，否则会产生信息的二义性。

b)各种模型之间如何协同工作以及访问控制模型如何与 VPN 技术有机结合的问题。由于现有访问控制模型并不是针对 VPN 设计的，无法做到与 VPN 的通信特性紧耦合，从而导致 VPN 服务器访问控制性能下降。比如：标准的访问控制模型并不知道 VPN 的通信特征，当需要对内部某一服务进行入侵检测时，势必需要对所有流入 VPN 的数据包进行分析。而事实上，如果访问控制已知 VPN 通信特征，只是对该服务的数据流进行控制，便可以极大地降低访问控制的负载。

面对上述问题，本文在分析 VPN 通信特征以及主流访问控制模型的控制特征的基础上，提出的 ACIAC 模型是面向 VPN 通信流、集中式控制信息管理的应用层访问控制模型，能够很好地解决上述问题：集中式信息管理避免了信息的二义性，简化了管理的复杂度；对 VPN 流的逻辑分类，提高了约束的针对性和专属性，提高了控制的性能。

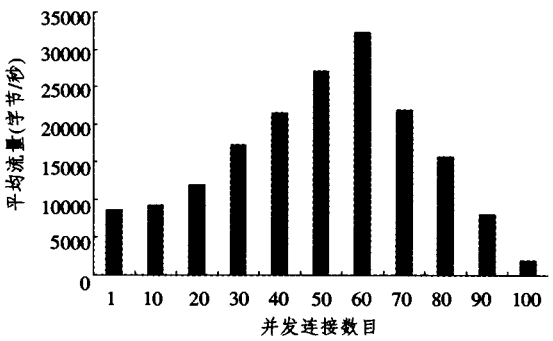
5　实验

SSL VPN 的性能可以从两个方面来分析：隧道内的数据流吞吐量和并发隧道的建立能力。在相同硬件条件下，隧道内的数据流吞吐量是由 SSL 的对称加密-信息验证码(MAC：Message Authentication Code)算法决定的。不同的加密-MAC 算法对隧道内的传输性能有显著的影响，根据 Bhatt D. V. 等人的研究，采用 RC4-MD5 算法的计算速度是采用

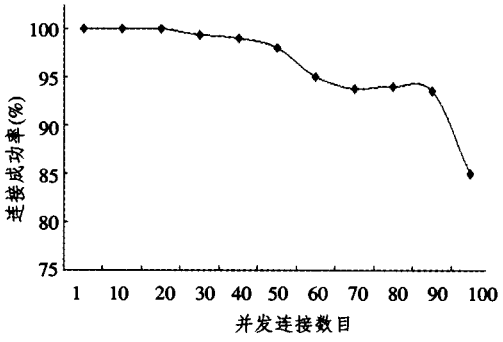
(下转封四)

(上接第 36 页)

3DES-SHA 算法的 7 倍[12]。另一方面,SSL VPN 的隧道建立涉及到一次 SSL 连接和一次 TCP 连接。我们以访问内部 HTTP 服务为例,通过性能测试软件 Siege[13] 来评估 VSB-SSL VPN 的并发连接性能。



(a) 并发流量统计



(b) 并发连接成功率并发连接数目

图 7 VSB-SSL VPN 并发测试

在我们的实验中,Siege 一次连接测试包含了一次 SSL 连接、一次 TCP 连接、一个 HTTP 页面(2kB)的加密/解密(采用 RC4-MD5 算法)以及该页面的传输。根据图 7(a)显示的 VSB-SSL VPN 并发流量统计信息,可以得知随着并发数目的增大,并发的平均流量不断增加,VPN 服务器的工作负载趋于饱和。在并发数到达 60 时,VBS-SSL 的并发流量到达峰值,这说明 VPN 服务器处于饱和工作状态(包含隧道建立操作和数据加密/解密操作)。当并发请求数超过 60 时,并发流量迅速下降,这是由于连接请求过多,VPN 服务器不能对数据及时转发导致的。另一方面,随着并发连接数目的增大,VPN 服务器来不及响应请求,使得连接请求被丢弃,从而导致隧道建立的成功率不断下降,如图 7(b)所示。

结束语 目前,国内外基于 VPN 的访问控制研究主要是基于 IP 层,还没有基于应用层的研究。本文提出的 ACIAC 模型是基于应用层的 VPN 访问控制。

ACIAC 模型是我们自行开发的应用层 VPN 结构的产物,是基于对当前主流访问控制模型的控制特点、反病毒机制、IDS 以及 VPN 通信机制的分析,构建的访问控制模型。该模型面向 VPN 通信流,通过对控制、检测需求的信息化描述以及集中式管理实现了能够同时基于用户/角色、基于内容、基于上下文的访问控制,更好地保护了 VPN 内部的服务群,从而提高了 VPN 结构的安全性。

参 考 文 献

1 Cohen R. On the Establishment of an Access VPN in Broadband Access Networks. Communications Magazine, IEEE, 2003, 41(2):156~163
2 Kent S, Atkinson R. Security Architecture for the Internet Protocol. RFC2401, November 1998
3 Dierks T, Allen C. The TLS Protocol Version 1.0. RFC2246, January 1999
4 Verschuren J, Govaerts R, Vandewalle J. Simultaneous Enforcement of the Bell-LaPadula and the Biba Security Policy Models in an OSI-distributed System. In: Singapore ICCS/ISITA '92, November, 1992. 257~263
5 Sandhu R S, Coyne E J, Feinstein H, et al. Role-Based Access Control Models. IEEE Computer, February 1996, 29(2): 38~47
6 Wolf R, Keinz T, Schneider M. A model for content-dependent access control for Web-based services with role-based approach. Database and Expert Systems Applications, 2003. 209~214
7 Spencer R, Smalley S, Loscocco P, et al. The Flask Security Architecture: System Support for Diverse Security Policies. In: Proceedings of the Eighth Security Symposium, August 1999. 123~139
8 Bertino E, Catania B, Ferrari E, et al. A System to Specify and Manage Multipolicy Access Control Models. Policies for Distributed Systems and Networks, 2002. 116~127
9 Jason J, Rafalow L, Vyncke E. IPSec Configuration Policy Information Model. RFC3585, August 2003
10 Guo Xin, Yang Kun, Galis A, et al. A Policy-based Network Management System for IP VPN. Communication Technology Proceedings, 2003. ICCT 2003, 2:1630~1633
11 Shan Rongsheng, et al. Network Security Policy for Large-scale VPN. Communication Technology Proceedings, 2003. ICCT 2003, 1:217~220
12 Bhatt D V, Schulze S, et al. Secure Internet Access to Gateway Using Secure Socket Layer. Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. 157~162
13 SIEGE-2.6. http://joedog.org/siege/

计算机科学

(1974 年 1 月创刊)
第 33 卷第 8 期 (月刊)
2006 年 8 月 25 日出版

国际标准连续出版物号 ISSN 1002-137X
国内统一连续物出版号 CN50-1075/TP

定价:30.00 元 国外定价:5 美元
邮发代号:78-68
发行范围:国内外公开

主管单位:国家科学技术部
主办单位:国家科技部西南信息中心
编辑出版:《计算机科学》杂志社
重庆市渝中区胜利路 132 号 邮政编码:400013
电话:(023) 63500828 E-mail:jsjkx@swic.ac.cn
网址:www.jsjkx.com
社 长:牟炳林
总 编:彭 丹
主 编:朱宗元
主编助理:徐书令
印 刷 者:重庆科情印务有限公司
总发行处:重庆市邮政局
订 购 处:全国各地邮政局
国外总发行:中国国际图书贸易总公司(北京 399 信箱)
国外代号:6210-MO