第5章 访问控制

主要内容

5.1 概述

5.2 访问控制模型

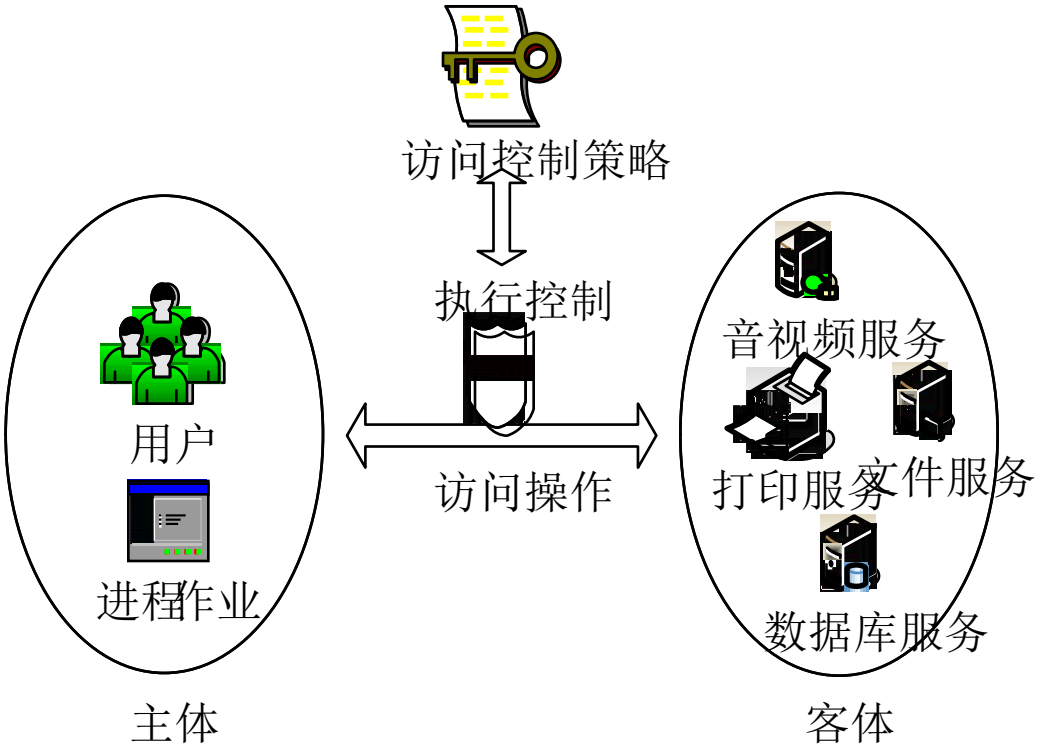
- 5.2.1 自主访问控制
- 5.2.2 强制访问控制
- 5.2.3 基于角色的访问控制

5.3 Windows系统的安全管理

- 5.3.1 Windows系统安全体系结构
- 5.3.2 Windows系统的访问控制
- 5.3.3 活动目录与组策略

5.1 概述

- 身份认证：识别“用户是谁”的问题
- 访问控制：管理用户对资源的访问



访问控制的基本组成元素

- 主体(Subject)：是指提出访问请求的实体，是动作的发起者，但不一定是动作的执行者。主体可以是用户或其它代理用户行为的实体（如进程、作业和程序等）。
- 客体(Object)：是指可以接受主体访问的被动实体。客体的内涵很广泛，凡是可以被操作的信息、资源、对象都可以认为是客体。
- 访问控制策略（Access Control Policy）：是指主体对客体的操作行为和约束条件的关联集合。简单地讲，访问控制策略是主体对客体的访问规则集合，这个规则集合可以直接决定主体是否可以对客体实施的特定的操作。

5.2 访问控制模型

- 1985年美国军方提出可信计算机系统评估准则TCSEC，其中描述了两种著名的访问控制模型：
  - 自主访问控制DAC(Discretionary Access Control)
  - 强制访问控制MAC(Mandatory Access Control)
- 1992年美国国家标准与技术研究所(NIST)的David Ferraiolo和Rick Kuhn提出一个模型
  - 基于角色的访问控制RBAC（Role Based Access Control）模型

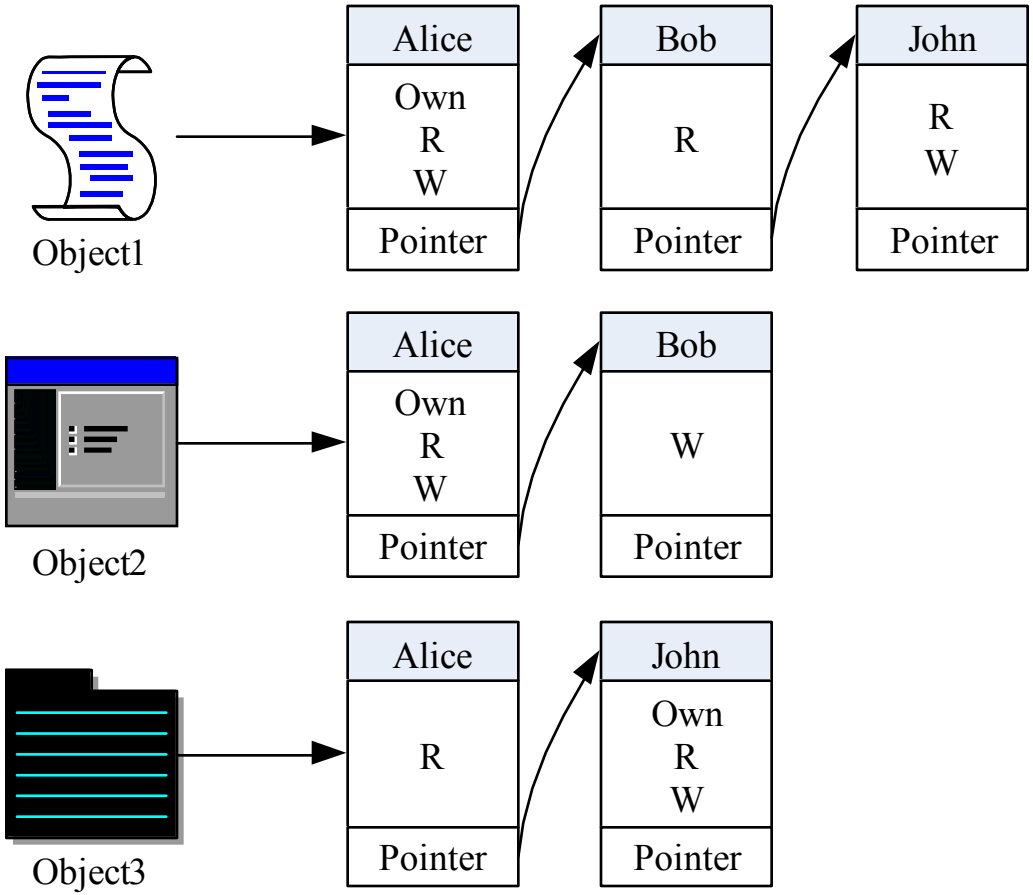
5.2.1 自主访问控制

- 自主访问控制DAC模型
  - 根据自主访问控制策略建立的一种模型，
  - 允许合法用户以用户或用户组的身份来访问系统控制策略许可的客体，同时阻止非授权用户访问客体。
  - 某些用户还可以自主地把自己所拥有的客体的访问权限授予其它用户。
- UNIX、LINUX以及Windows NT等操作系统都提供自主访问控制的功能。

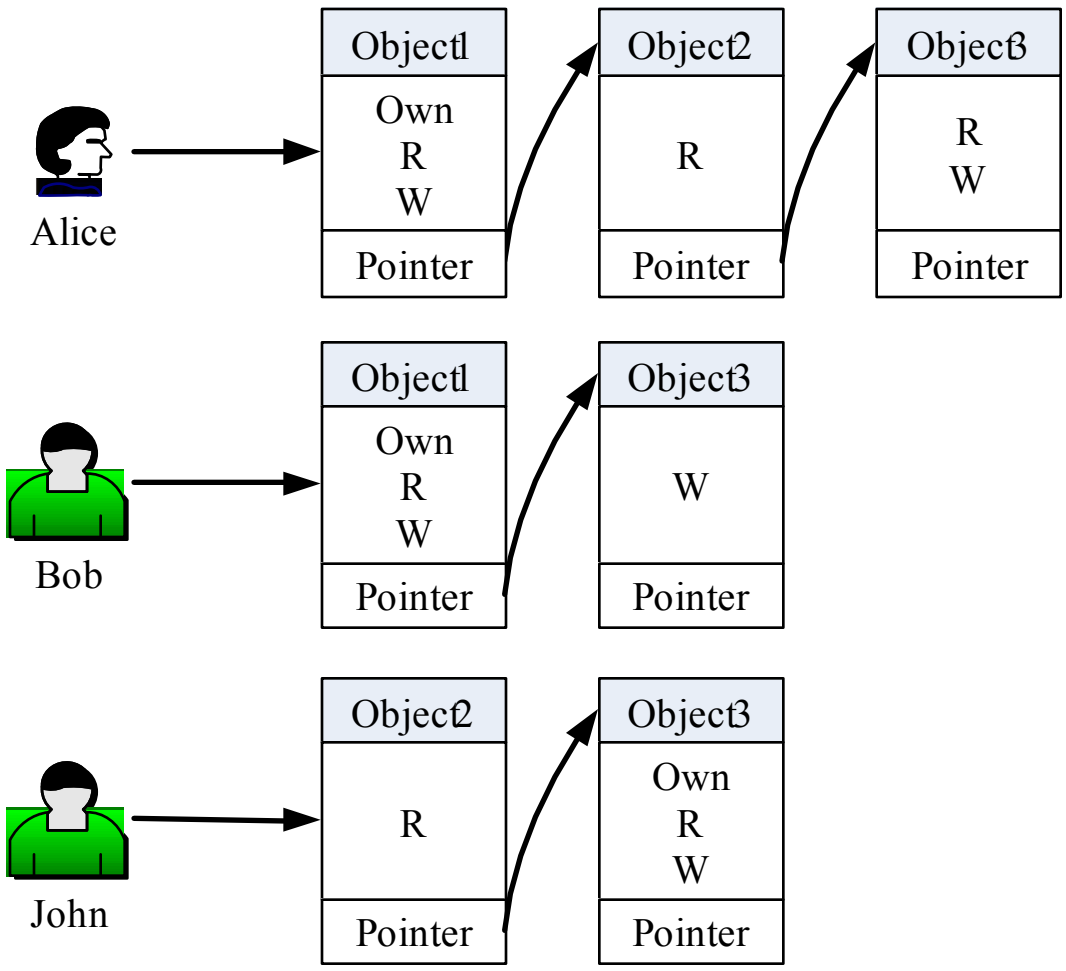
访问权限信息存储

- 特权用户为普通用户分配的访问权限信息的形式
  - 访问控制表ACL（Access Control Lists）
  - 访问控制能力表ACCL（Access Control Capability Lists）
  - 访问控制矩阵ACM（Access Control Matrix）

访问控制表ACL


Alice
Own
R
W
Pointer
Bob
R
Pointer
John
R
W
Pointer
Object1
Alice
Own
R
W
Pointer
Bob
W
Pointer
Object2
Alice
R
Pointer
John
Own
R
W
Pointer
Object3

访问控制能力表ACCL



访问控制矩阵ACM

主体 \ 客体	Object1	Object2	Object3
Alice	Own，R，W	R	R，W
Bob	R	Own，R，W	
John	R，W		Own，R，W

5.2.2 强制访问控制

- 强制访问控制MAC是一种多级访问控制策略
  - 系统事先给访问主体和受控客体分配不同的安全级别属性。
  - 在实施访问控制时，系统先对访问主体和受控客体的安全级别属性进行比较，再决定访问主体能否访问该受控客体。
- MAC模型形式化描述
  - 主体集S和客体集O
  - 安全类SC(x) = < L， C >
    - L为有层次的安全级别Level
    - C为无层次的安全范畴Category

访问的四种形式

- 向下读（RD，Read Down）：
  - 主体安全级别高于客体信息资源的安全级别时，即SC(s)≥SC(o)，允许读操作； Bell-LaPadula
- 向上读（RU，Read Up）：
  - 主体安全级别低于客体信息资源的安全级别时，即SC(s)≤SC(o)，允许读操作； Biba
- 向下写（WD，Write Down）：
  - SC(s)≥SC(o)时，允许写操作； Biba
- 向上写（WU，Write Up）：
  - SC(s)≤SC(o)时，允许写操作。 Bell-LaPadula

MAC信息流安全控制

主体 \ 客体	TS	C	S	U
TS	R/W	R	R	R
C	W	R/W	R	R
S	W	W	R/W	R
U	W	W	W	R/W

High ↓ ↓ ↓ Low

5.2.3 基于角色的访问控制

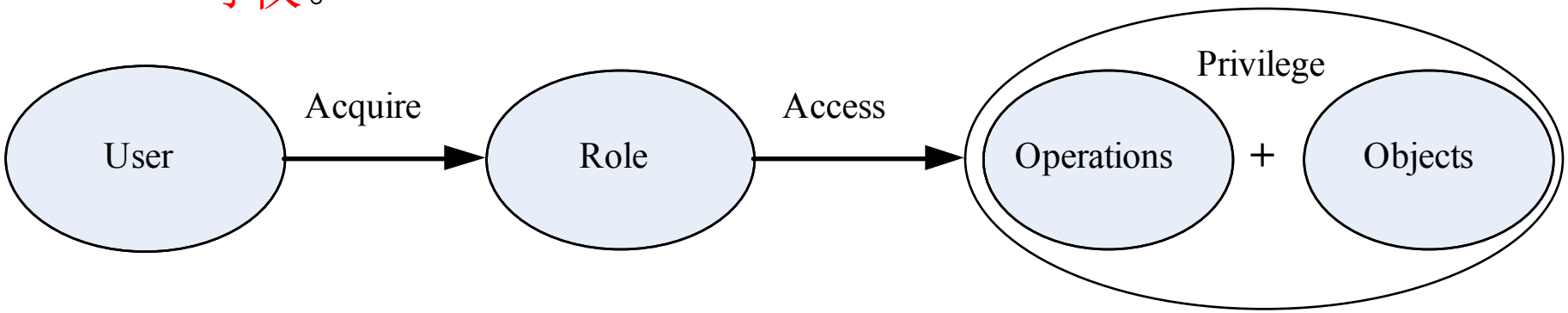
- Group的概念，一般认为Group是具有某些相同特质的用户集合。
- 在UNIX操作系统中Group可以被看成是拥有相同访问权限的用户集合，
  - 定义用户组时会为该组赋予相应的访问权限。
  - 如果一个用户加入了该组，则该用户即具有了该用户组的访问权限
  - 角色Role的概念，可以这样理解一个角色是一个与特定工作活动相关联的行为与责任的集合

角色Role的理解

- 一个角色是一个与特定工作活动相关联的行为与责任的集合。
  - Role不是用户的集合，也就与组Group不同。
  - 当将一个角色与一个组绑定，则这个组就拥有了该角色拥有的特定工作的行为能力和责任。
  - 组Group和用户User都可以看成是角色分配的单位和载体。
  - 而一个角色Role可以看成具有某种能力或某些属性的主体的一个抽象。

引入角色Role的目的

- Role的目的：
  - 为了隔离User与Privilege。
  - Role作为一个用户与权限的代理层，所有的授权应该给予Role而不是直接给User或Group。
  - RBAC模型的基本思想是将访问权限分配给一定的角色，用户通过饰演不同的角色获得角色所拥有的访问许可权。



例子

- 在一个公司里，用户角色可以定义为经理、会计、出纳员和审计员，具体的权限如下：
  - 经理：允许查询公司的经营状况和财务信息，但不允许修改具体财务信息，必要时可以根据财务凭证支付或收取现金，并编制银行账和现金帐；
  - 会计：允许根据实际情况编制各种财务凭证及账簿，但不包括银行账和现金帐；
  - 出纳员：允许根据财务凭证支付或收取现金，并编制银行账和现金帐；
  - 审计员：允许查询审查公司的经营状况和财务信息，但不允许修改任何账目。

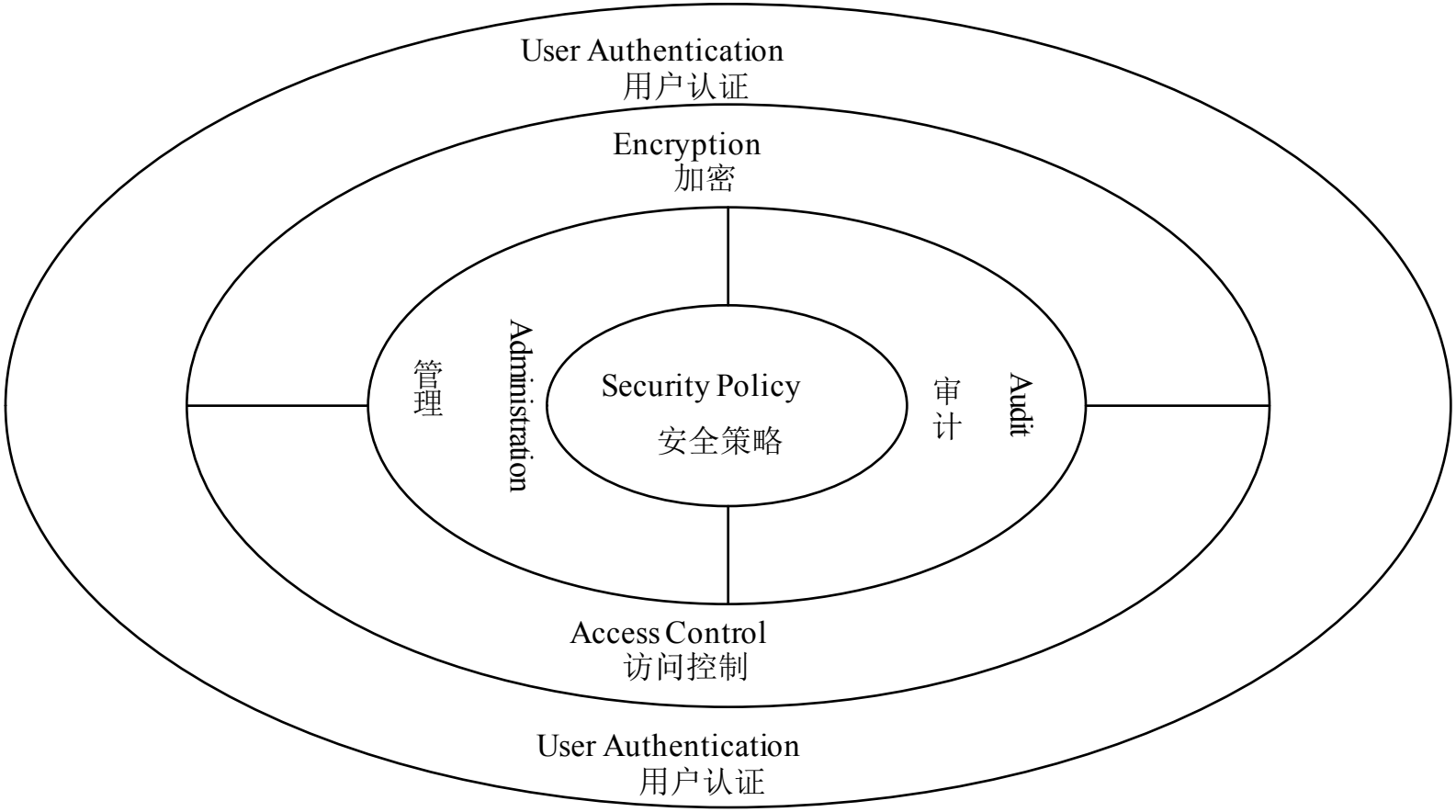
- RBAC的策略陈述易于被非技术的组织策略者理解，既具有基于身份策略的特征，也具有基于规则策略的特征。
- 在基于组或角色的访问控制中，一个用户可能不只是一个组或角色的成员，有时又可能有所限制。
- 例如经理可以充当出纳员的角色，但不能负责会计工作，即各角色之间存在相容和相斥的关系。

制定访问控制策略的三个基本原则

- 最小特权原则
  - 是指主体执行操作时，按照主体所需权利的最小化原则分配给主体权力。
  - 最小特权原则的优点是最大限度地限制了主体实施授权行为，可以避免来自突发事件和错误操作带来的危险。
- 最小泄漏原则：
  - 是指主体执行任务时，按照主体所需要知道信息的最小化原则分配给主体访问权限。
- 多级安全策略：
  - 是指主体和客体间的数据流方向必须受到安全等级的约束。多级安全策略的优点是避免敏感信息的扩散。
  - 对于具有安全级别的信息资源，只有安全级别比它高的主体才能够对其访问。

5.3 Windows系统的安全管理

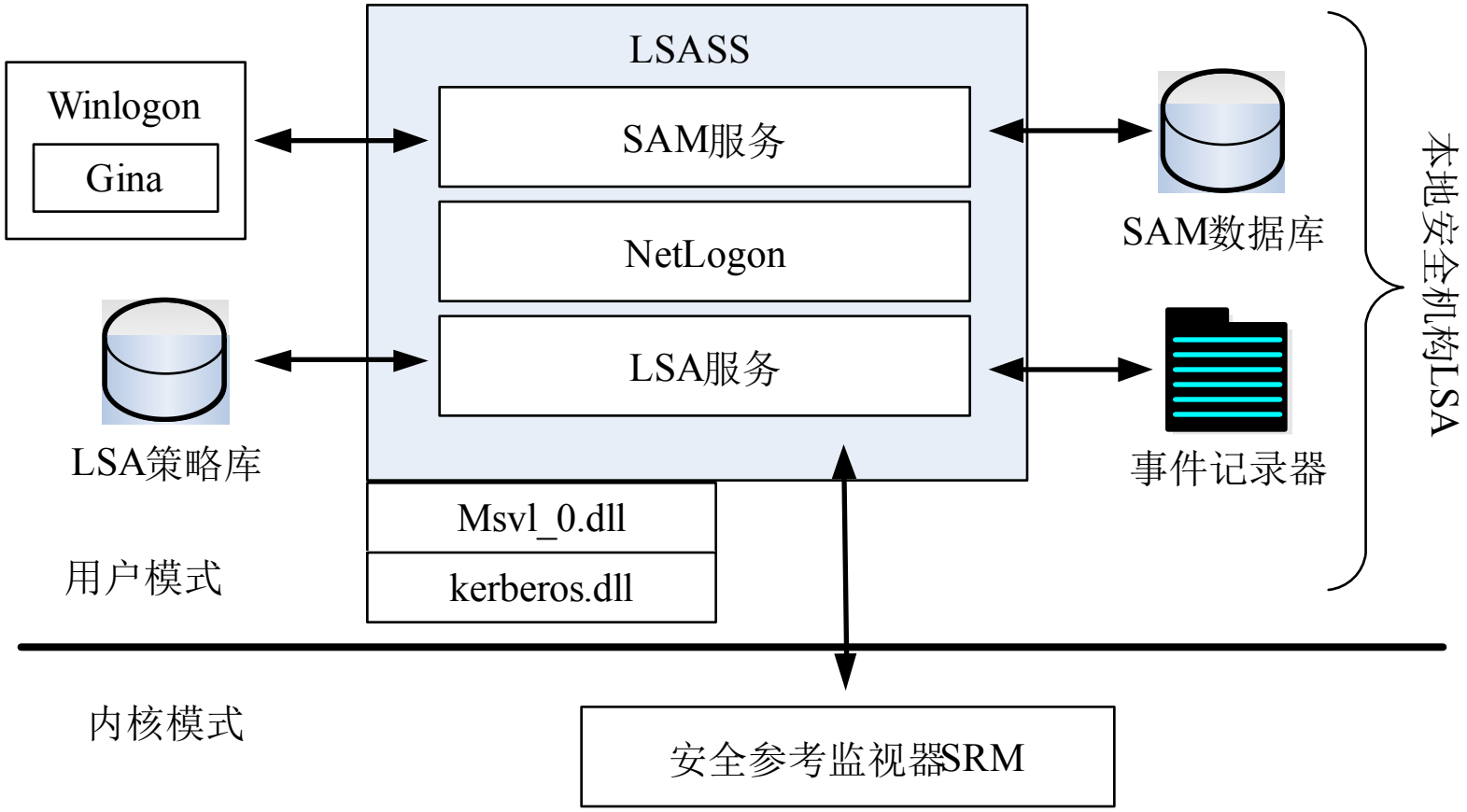
- **5.3.1 Windows系统安全体系结构**



安全主体

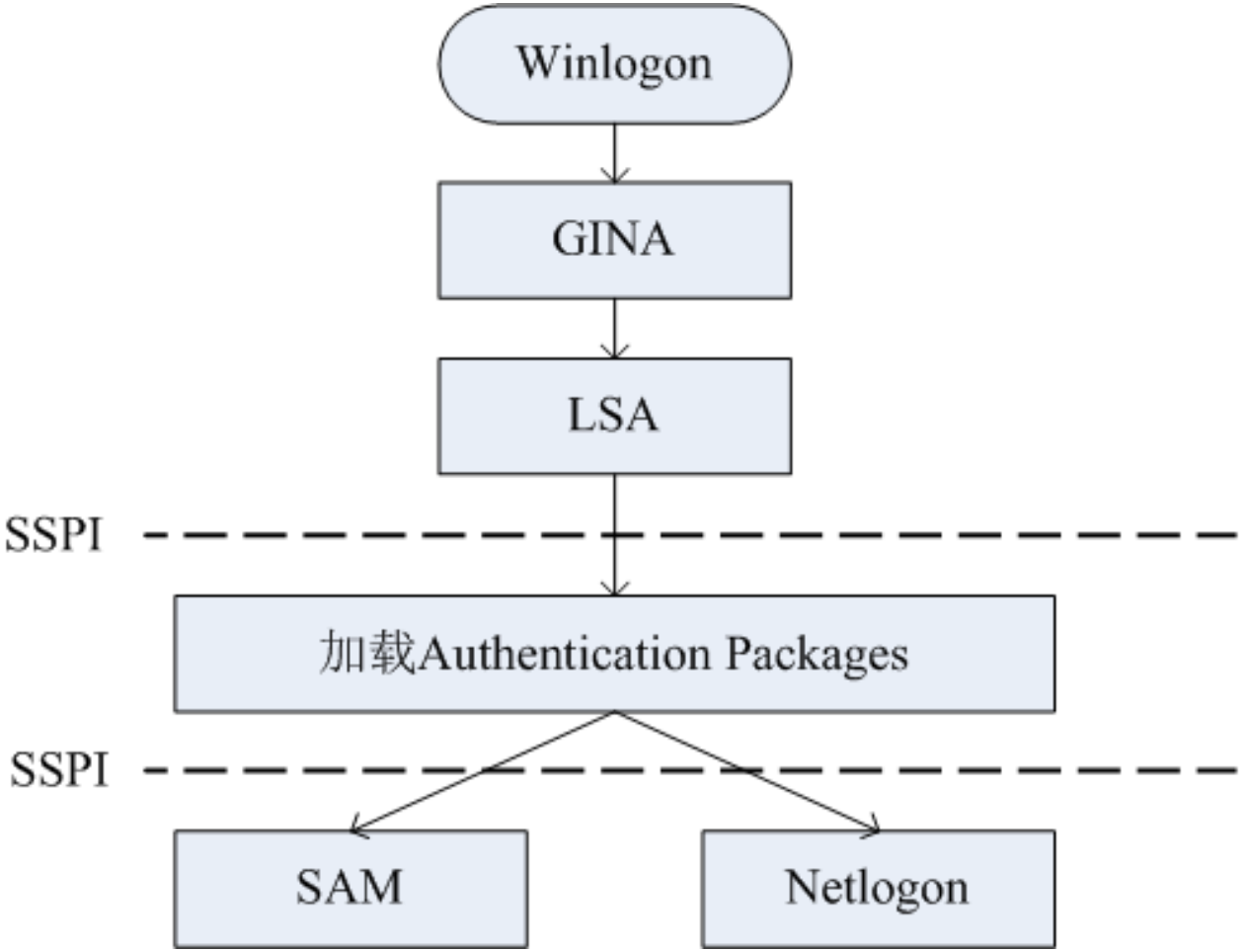
- Windows系统的安全性主要围绕安全主体展开，保护其安全性。
- 安全主体主要包括用户、组、计算机以及域等。
  - 用户是Windows系统中操作计算机资源的主体，每个用户必须先行加入Windows系统，并被指定唯一的账户，
  - 组是用户账户集合的一种容器，同时组也被赋予了一定的访问权限，放到一个组中的所有账户都会继承这些权限；
  - 计算机是指一台独立计算机的全部主体和客体资源的集合，也是Windows系统管理的独立单元；
  - 域是使用域控制器(DC, Domain Controller)进行集中管理的网络，域控制器是共享的域信息的安全存储仓库，同时也作为域用户认证的中央控制机构。

安全子系统



Windows安全子系统

Windows登录认证流程



5.3.2 Windows系统的访问控制

- 访问控制模块的组成
  - 访问令牌（Access Token）和安全描述符（Security Descriptor），它们分别被访问者和被访问者持有。通过访问令牌和安全描述符的内容，Windows可以确定持有令牌的访问者能否访问持有安全描述符的对象。
- 访问控制的基本控制单元“账户”。
  - 账户是一种参考上下文(context)，是一个具有特定约束条件的容器，也可以理解为背景环境。
  - 操作系统在这个上下文描述符上运行该账户的大部分代码。
  - 那些在登录之前就运行的代码（例如服务）运行在一个账户（特殊的本地系统账户SYSTEM）的上下文中。

安全标识符SID

- Windows中的每个账户或账户组都有一个安全标识符SID（Security Identity）
  - Administrator、Users等账户或者账户组在Windows内部均使用SID来标识的。
  - 每个SID在同一个系统中都是唯一的。
    - 例如S-1-5-21-1507001333-1204550764-1011284298-500就是一个完整的SID。
    - 第一个数字（本例中的1）是修订版本编号，
    - 第二个数字是标识符颁发机构代码（Windows 2000为5）
    - 4个子颁发机构代码
    - 相对标识符RID（Relative Identifier） RID 500代表Administrator账户，RID 501是Guest账户。从1000开始的RID代表用户账户

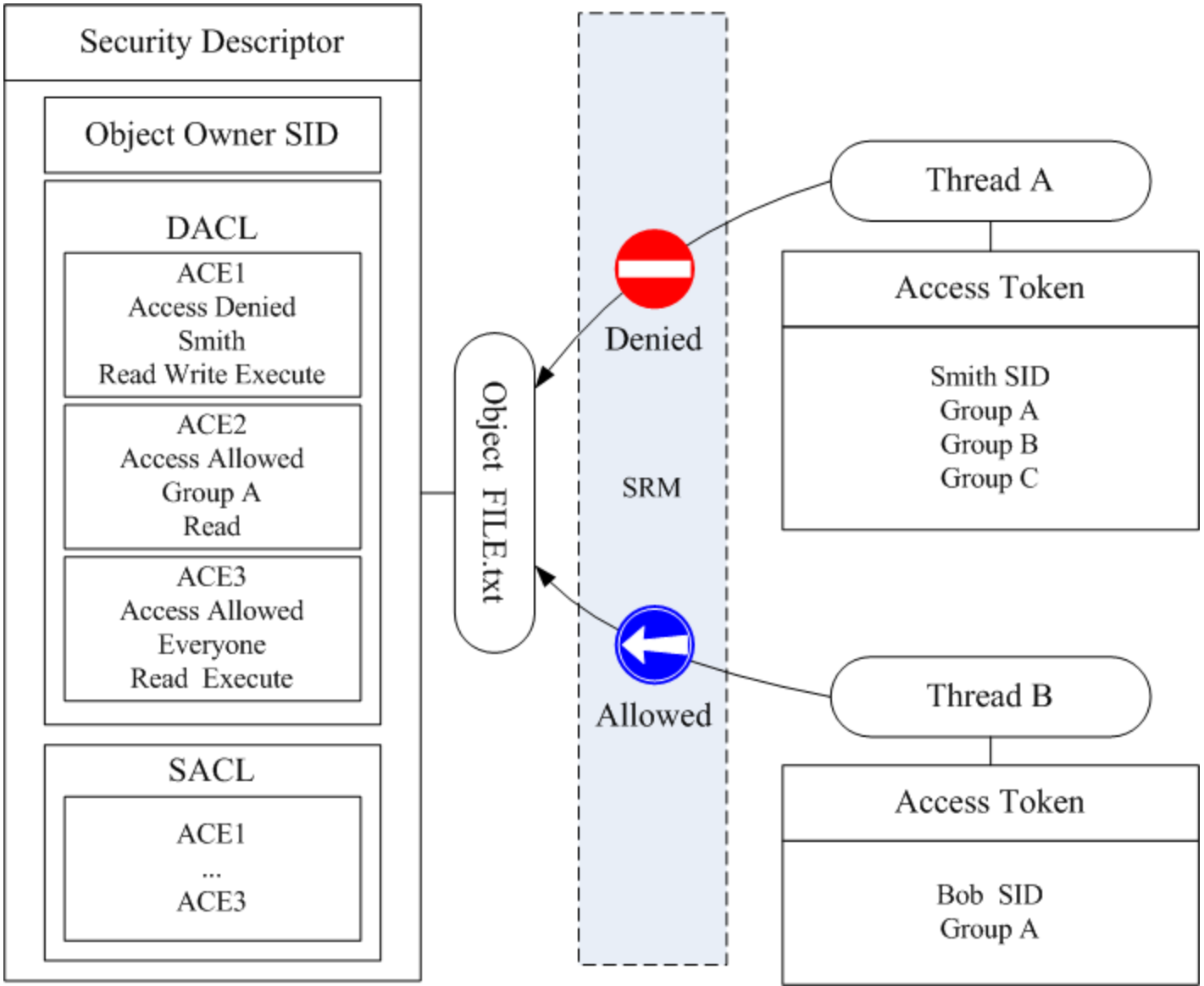
访问令牌

- 每个访问令牌都与特定的Windows账户相关联，访问令牌包含该帐户的SID、所属组的SID以及帐户的特权信息。

```
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.
C:\>whoami /all
[User]   = "Smith\Administrator"  S-1-5-21-2000478354-842925246-1202660629-500
[Group 1] = " Smith \None"  S-1-5-21-2000478354-842925246-1202660629-513
[Group 2] = "Everyone"  S-1-1-0
[Group 3] = " Smith \Debugger Users"  S-1-5-21-2000478354-842925246-
1202660629-1004
[Group  4] = "BUILTIN\Administrators"  S-1-5-32-544
[Group  5] = "BUILTIN\Users"  S-1-5-32-545
[Group  6] = "NT AUTHORITY\INTERACTIVE"  S-1-5-4
[Group  7] = "NT AUTHORITY\Authenticated Users"  S-1-5-11
[Group  8] = "LOCAL"  S-1-2-0
```

- Microsoft Windows [版本 6.1.7600]
- 版权所有 (c) 2009 Microsoft Corporation。保留所有权利。
- C:\Users\zjh>whoami
- zjh-pc\zjh
- C:\Users\zjh>whoami/all
- 用户信息
- ----------
- 用户名 SID
- ========== ==========
- zjh-pc\zjh S-1-5-21-868672325-3564177769-4166673043-1000
- 组信息
- ----------
- 组名 类型 SID 属性
- ========== ====== ==========
- Everyone 已知组 S-1-1-0
- 必需的组, 启用于默认, 启用的组
- zjh-PC\Debugger Users 别名 S-1-5-21-868672325-3564177769-4166
- 673043-1001 必需的组, 启用于默认, 启用的组
- BUILTIN\Administrators 别名 S-1-5-32-544
- 只用于拒绝的组
- BUILTIN\Users 别名 S-1-5-32-545
- 必需的组, 启用于默认, 启用的组
- NT AUTHORITY\INTERACTIVE 已知组 S-1-5-4
- 必需的组, 启用于默认, 启用的组
- 控制台登录 已知组 S-1-2-1
- 必需的组, 启用于默认, 启用的组
- NT AUTHORITY\Authenticated Users 已知组 S-1-5-11
- 必需的组, 启用于默认, 启用的组
- NT AUTHORITY\This Organization 已知组 S-1-5-15
- 必需的组, 启用于默认, 启用的组
- LOCAL 已知组 S-1-2-0
- 必需的组, 启用于默认, 启用的组
- NT AUTHORITY\NTLM Authentication 已知组 S-1-5-64-10
- 必需的组, 启用于默认, 启用的组
- Mandatory Label\Medium Mandatory Level 标签 S-1-16-8192
- 必需的组, 启用于默认, 启用的组
- 特权信息
- ----------
- 特权名 描述 状态
- ========== ========== ======
- SeShutdownPrivilege 关闭系统 已禁用
- SeChangeNotifyPrivilege 绕过遍历检查 已启用
- SeUndockPrivilege 从扩展坞上取下计算机 已禁用
- SeIncreaseWorkingSetPrivilege 增加进程工作集 已禁用
- SeTimeZonePrivilege 更改时区 已禁用

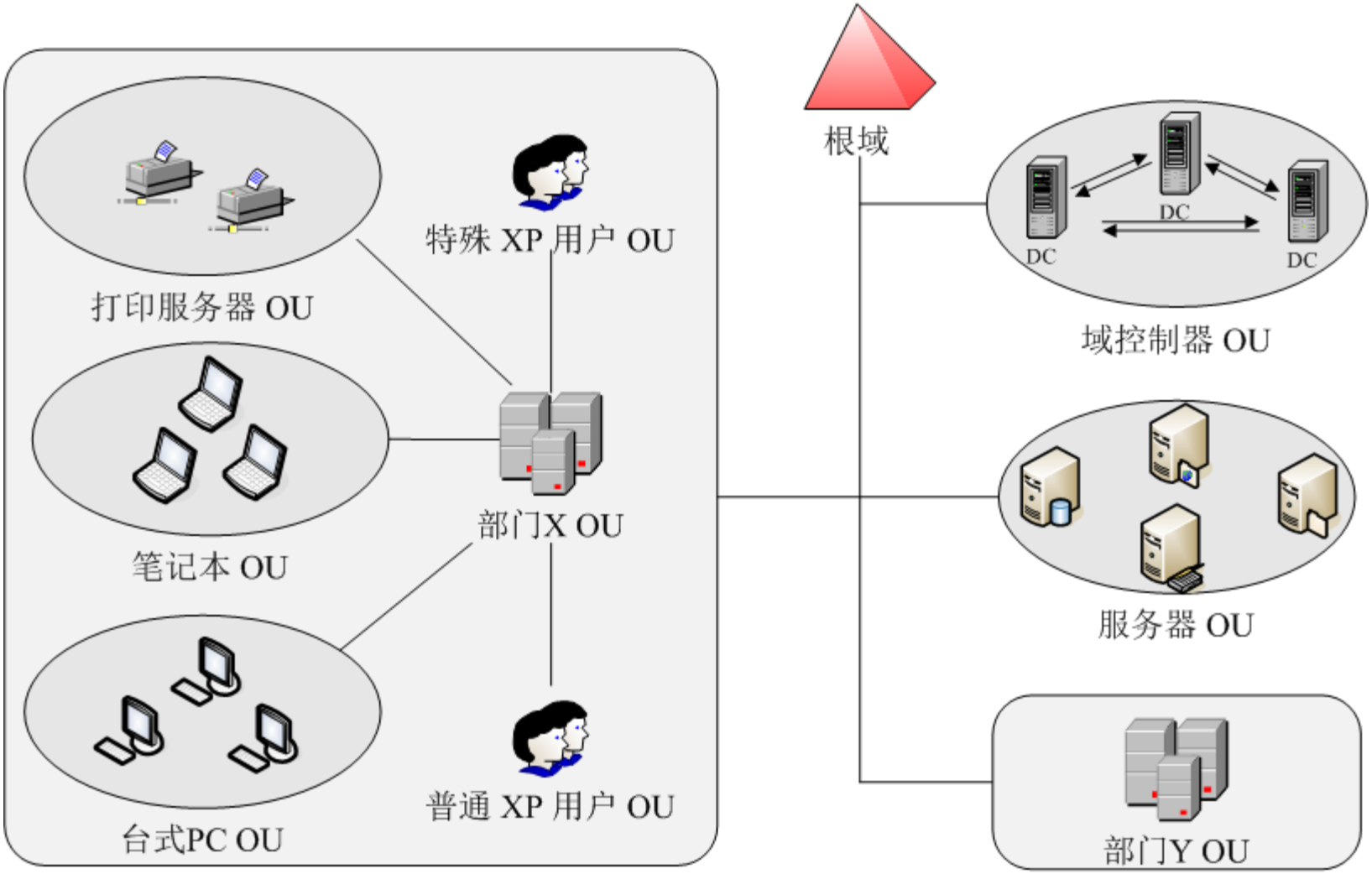
Window 访问控制


Security Descriptor
Object Owner SID
DACL
ACE1
Access Denied
Smith
Read Write Execute
ACE2
Access Allowed
Group A
Read
ACE3
Access Allowed
Everyone
Read Execute
SACL
ACE1
...
ACE3
Object FILE.txt
Denied
SRM
Allowed
Thread A
Access Token
Smith SID
Group A
Group B
Group C
Thread B
Access Token
Bob SID
Group A

5.3.3 活动目录与组策略

- 活动目录AD（Active Directory）是一个面向网络对象管理的综合目录服务。
- 网络对象包括用户、用户组、计算机、打印机、应用服务器、域、组织单元（OU）以及安全策略等。
- AD 提供的是各种网络对象的索引集合，也可以看作是数据存储的视图，
- 将分散的网络对象有效地组织起来，建立网络对象索引目录，并存储在活动目录的数据库内。

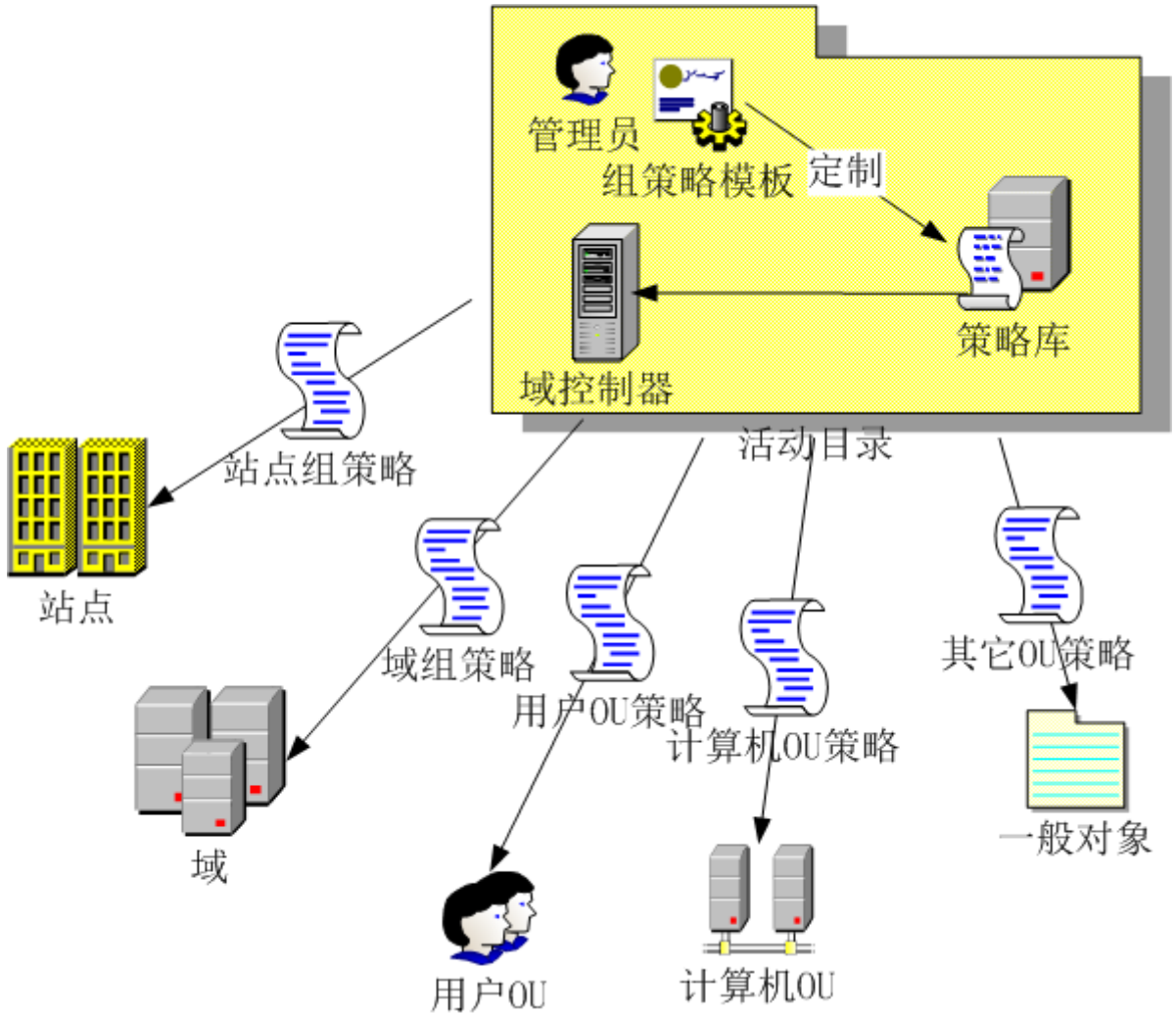
活动目录AD的管理划分



组策略GP

- 活动目录AD是Windows网络中重要的安全管理平台，组策略GP（Group Policy）是其安全性的重要体现。
- 组策略可以理解为依据特定的用户或计算机的安全需求定制的安全配置规则。
  - 管理员针对每个组织单元OU定制不同的组策略，并将这些组策略存储在活动目录的相关数据库内，可以强制推送到客户端实施组策略。
- 活动目录AD可以使用组策略命令来通知和改变已经登录的用户的组策略，并执行相关安全配置。

组策略工作流程


管理员
组策略模板
定制
策略库
域控制器
活动目录
站点组策略
站点
域组策略
用户OU策略
计算机OU策略
其它OU策略
域
用户OU
计算机OU
一般对象

组策略的实施

- 注册表是Windows系统中保存系统应用软件配置的数据库。
- 很多配置都是可以自定义设置的，但这些配置发布在注册表的各个角落，如果是手工配置，可想是多么困难和繁琐。
- 组策略可以将系统中重要的配置功能汇集成一个配置集合，管理人员通过配置并实施组策略，达到直接管理计算机的目的。
- 简单点说，实施组策略就是修改注册表中的相关配置。

组策略和活动目录AD配合

- 组策略分为基于活动目录的和基于本地计算机的两种：
  - AD组策略存储在域控制器上活动目录AD的数据库中，它的定制实施由域管理员来执行；而本地组策略存放在本地计算机内，由本地管理员来定制实施。
  - AD组策略实施的对象是整个组织单元OU；本地组策略只负责本地计算机。
- 组策略和活动目录AD配合
  - 组策略部署在OU、站点或域的范围内，也可以部署在本地计算机上。部署在本地计算机时，组策略不能发挥其全部功能，只有和AD配合，组策略才可以发挥出全部潜力。

组策略的主要工作

① 部署软件

② 设置用户权力

③ 软件限制策略

- 管理员可以通过配置组策略，限制某个用户只能运行特定的程序或执行特定的任务。

④ 控制系统设置：

- 允许管理员统一部署网络用户的Windows服务。

⑤ 设置登录、注销、关机、开机脚本。

⑥ 通用桌面控制

⑦ 安全策略

⑧ 重定向文件夹

⑨ 基于注册表的策略设置

Any question?