CA Access Control

端点管理指南：用于 UNIX

12.6

第三方通知

示例脚本和示例 SDK 代码

CA Access Control 产品随附的示例脚本和示例 SDK 代码均“按原样”提供，仅供参考之用。 在特定环境下，可能需要对它们进行调整，而且在生产系统中部署它们之前，未经事先测试和验证不应将其用于生产。

CA Technologies 不向这些示例提供支持，而且对于这些脚本可能会引起的任何错误概不负责。

CA Technologies 产品引用

本文档引用以下 CA Technologies 产品：

- CA Access Control 企业版
- CA Access Control
- CA Single Sign-On (CA SSO)
- CA Top Secret®
- CA ACF2™
- CA Audit
- CA Network and Systems Management（CA NSM，以前为 Unicenter NSM 和 Unicenter TNG）
- CA Software Delivery（以前为 Unicenter Software Delivery）
- CA Service Desk（以前为 Unicenter Service Desk）
- User Activity Reporting（以前是 CA Enterprise Log Manager）
- CA Identity Manager

文档约定

CA Access Control 文档使用以下约定：

格式	含义
等宽字体	代码或程序输出
斜体	重点或新术语
粗体	必须完全按照显示内容键入的文本
正斜杠 (/)	用于描述 UNIX 和 Windows 路径的独立于平台的目录分隔符

文档在解释命令语法和用户输入（以等宽字体显示）时还会使用以下特殊约定：

格式	含义
斜体	您必须提供的信息
用方括号括起来 ([])	可选运算符

格式	含义
用大括号括起来 ({})	强制运算符集
用管道符 (|) 分隔的选项。	分隔可选运算符（选择一项）。 例如：下面的示例*既*可以表示用户名，*也*可以表示组名： `{username|groupname}`
…	指明前面的项或项组可以重复
下划线	默认值
前面带空格的行尾反斜杠 (\)	有时，在本指南中，一行无法容纳一个命令。 在这些情况下，行结尾处的空格加上反斜杠 (\) 就表示该命令延续到下一行。 **注意：**请勿复制反斜杠字符，并且请省略换行符。 这些不是实际命令语法的一部分。

示例：命令表示法约定

下面的代码说明了在本指南中使用命令约定的方式：

```
ruler className [props({all|{propertyName1[,propertyName2]...})]
```

在该示例中：

- 命令名称 (ruler) 以常规等宽字体显示，必须按照显示内容键入。
- *className* 选项以斜体显示，因为它是一个类名（例如 USER）的占位符。
- 即使不带有方括号中的第二部分，您也可以运行该命令，方括号表示该运算符是可选的。
- 使用可选参数 (props) 时，可以选择关键字 *all*，也可以指定一个或多个属性名（以逗号分隔）。

文件位置约定

CA Access Control 文档使用以下文件位置约定：

- *ACInstallDir*—默认 CA Access Control 安装目录。
  - Windows—C:\Program Files\CA\AccessControl\
  - UNIX—/opt/CA/AccessControl/
- *ACSharedDir*—CA Access Control for UNIX 使用的默认目录。
  - UNIX—/opt/CA/AccessControlShared

- *ACServerInstallDir*—默认 CA Access Control 企业管理 安装目录。
  - /opt/CA/AccessControlServer
- *DistServerInstallDir*—默认分发服务器安装目录。
  - /opt/CA/DistributionServer
- *JBoss_HOME*—默认 JBoss 安装目录。
  - /opt/jboss-4.2.3.GA

联系技术支持

要获取在线技术帮助以及办公地址、主要服务时间和电话号码的完整列表，请联系技术支持：http://www.ca.com/worldwide。

文档更改

该版本没有任何文档更改。

目录

第 1 章： 简介

此部分包含以下主题：

关于本指南

本指南介绍了 CA Access Control for UNIX（为开放系统提供完整安全解决方案的产品）所使用的概念， 还介绍了 UNIX 端点管理任务和概念。

本指南也随 CA Access Control 企业版 提供，CA Access Control 企业版 提供企业管理和报告功能，以及高级策略管理功能。

为了简化术语，在本指南中我们将此产品称为 CA Access Control。

使用本指南的用户

本指南是为负责实现和维护受 CA Access Control 保护的环境的安全管理员和系统管理员而编写的。

第 2 章： 管理端点

CA Access Control 是动态绑定到操作系统的软件产品，是开放系统的主动、全面的安全软件解决方案。 用户每次请求有关安全的操作（例如，打开文件、替换用户 ID 或获取网络服务）时，CA Access Control 会实时截获该事件并评估其有效性，然后才将控制权转交给标准的操作系统 (OS) 功能。

此部分包含以下主题：

什么是 CA Access Control？

CA Access Control 为您提供用来管理本地平台安全性的强大工具，从而能够实施可完全根据企业安全要求自定义的安全策略。 CA Access Control 可以为本地操作系统中可用用户、组和资源之外的用户、组和资源提供安全保护， 在整个组织范围内集中管理安全性，并将您的 Windows 和 UNIX 安全策略集成到一个异类环境中。

为什么 UNIX 需要保护？

许多操作系统都具有使用某种技术的内置访问控制。 IBM 的 z/OS 是颇具声望的、成熟的大型机操作系统，其中包括的系统授权工具 (SAF) 就是该操作系统本身为验证用户授权而发出的调用集。

z/OS 环境中的 Access Control 软件为 SAF 调用设置返回代码，然后 z/OS 会根据该代码批准或拒绝访问。 设置什么返回代码由安全管理员根据安全数据库中定义的访问规则和策略决定。

其他操作系统（如 OS/2）也提供类似的访问控制技术。 OS/2 Access Control 模块（称为安全启用服务 (SES)）的概念基础与 z/OS SAF 相同。

遗憾的是，基于 UNIX 的操作系统却没有采用这种设计方式。 授权决定主要是针对文件访问做出的，并且由操作系统本身使用文件 *inode* 条目中的九个位 (rwx-rwx-rwx) 执行。 与 SAF 不同的是，没有为事件截获提供出口点。 因此，要执行比大型机类型安全软件包的功能更加复杂的功能，需要进一步提高安全性。

它是如何工作的？

除了提供常规的安全功能（例如访问规则数据库、审核日志和管理工具）之外，CA Access Control 还截获要保护的操作系统事件。 由于 CA Access Control 必须与很多不同的操作系统协同工作，因此它截获内存中的事件。 不会对系统文件进行任何更改，也根本不会修改操作系统。

保护的对象是什么？

CA Access Control 保护下列实体：

- **文件**

  用户是否有权访问特定文件？

  CA Access Control 限制用户访问文件的能力。 您可以给予用户一种或多种访问权限，例如读取、写入、执行、删除和重命名。 访问权限的指定可以与单个文件相关，也可以与一组命名相似的文件相关。

- **终端**

  用户是否有权使用特定终端？

  该检查是在登录过程中完成的。 在 CA Access Control 数据库中，可以定义单个终端和终端组及其访问规则（即描述允许哪些用户或用户组使用终端或终端组）。 终端保护确保未经授权的终端或工作站不能用来登录到授予强大权限的用户帐户。

- **登录时间**

  用户是否有权在特定日期的特定时间登录？

  大多数用户只在工作日和工作时间使用其工作站；工作日和工作时间登录限制以及节假日限制，是为了防止黑客和其他未经授权的访问者的登录。

- **TCP/IP**

  另一个工作站是否有权从本地计算机接收 TCP/IP 服务？ 另一个工作站是否有权向本地计算机提供 TCP/IP 服务？ 是否允许另一个工作站从本地工作站的每个用户接收服务？

  开放系统（计算机和网络都开放的系统）的优点也是缺点。 一旦计算机连接到外面的世界，您就无法确保谁进入系统以及外来用户可以进行何种破坏（无论有意还是无意）。 CA Access Control 提供“防火墙”，可以防止本地工作站和服务器向未知工作站提供服务。

- **多个登录权限**

  是否允许用户从第二个终端登录?

  术语*并发登录*指的是用户从多个终端登录到系统的能力。 CA Access Control 可以阻止用户多次登录。 这可以防止入侵者登录到已经登录的用户的帐户。

- **用户定义的实体**

  可以定义和保护常规实体（例如 TCP/IP 服务和终端）和功能实体（也称为*抽象*对象；例如在数据库中执行事务和访问记录）。

- **管理员权限方面**

  CA Access Control 提供了向操作员指派超级用户权限而又同时限制超级用户帐户权限的方法。

- **替换-用户**

  用户是否有权替换他们的用户 ID?

  UNIX *setuid* 系统调用是操作系统提供的最敏感的服务之一，CA Access Control 通过截获该调用来检查用户是否有权执行替换。 替换用户权限检查包括程序通路 - 只允许用户通过特定程序替换他们的用户 ID。 这在控制哪些用户可以替换 root 从而获得 root 访问权限时尤其重要。

- **替换-组**

  用户是否有权发出 newgrp（替换组）命令?

  替换组保护类似于替换用户保护。

- **Setuid 和 setgid 程序**

  某个 setuid 或 setgid 程序是否可以受托? 用户是否有权调用它?

  安全管理员可以对标记为 setuid 或 setgid 可执行文件的程序进行测试，以确保它们不会包含任何可用来获得非法访问权的安全漏洞。 通过测试并被视为安全的程序定义为受托程序。 CA Access Control 自我保护模块（也称为 CA Access Control *Watchdog*）知道哪个程序在特定时间处于控制之下，并检查该程序在被归为受托程序之后是否经过修改或移动。 如果受托程序经过修改或移动，则该程序不再被视为受托程序，CA Access Control 将不允许它运行。

另外，CA Access Control 可以防止各种故意的和偶然的威胁，包括：

- **终止尝试**

  CA Access Control 可以用来保护关键服务器和服务或后台程序，防止终止尝试。

- **密码攻击**

  CA Access Control 防止各种密码攻击、强制您的站点实施密码定义策略并检测入侵尝试。

- **密码缺点**

  CA Access Control 策略描述了强制用户创建和使用高质量密码的规则。 为了确保用户创建和使用可接受的密码，CA Access Control 可以设置密码的最长和最短使用期限、限制某些字词、禁用重复字符并强制遵守其他限制。 密码使用期限不能太长。

- **帐户管理**

  CA Access Control 策略确保正确处理睡眠帐户。

- **域管理**

  CA Access Control 可以实施密码保护，并跨 NIS 和非 NIS 域强制安全。

如何保护它？

CA Access Control 在操作系统完成初始化后立即启动。 CA Access Control 将 hook 放置在必须保护的系统服务中。 这样，便可以在执行服务之前将控制权传递给 CA Access Control。 CA Access Control 决定是否应将该服务授予用户。

例如，用户可以尝试访问受 CA Access Control 保护的资源。 该访问请求生成了对内核的系统调用，从而可以打开资源。 CA Access Control 截获系统调用，并决定是否授予访问权限。 如果授予权限，则 CA Access Control 将控制权转交给常规系统服务；如果 CA Access Control 拒绝权限，它会将标准权限拒绝错误代码返回到激活系统调用的程序，系统调用即结束。

授权决定基于数据库中定义的访问规则和策略。 数据库描述了两种对象：访问者和资源。 *访问者*是用户和组。 *资源*是要保护的对象，例如文件和服务。 数据库中的每个记录都描述了访问者或资源。

每个对象都属于一个类 - 同类对象的集合。 例如，TERMINAL 是包含 CA Access Control 所保护的终端（工作站）对象的类。

类激活

CA Access Control 存储有关 CLASS 在数据库中是否活动的信息。 当 CA Access Control 启动时，它会将活动类的列表传递给 SEOS_syscall，因此 CA Access Control 无需经常拦截这些类。 只有在用户更改类的活动状态时 CA Access Control 才会拦截该类。 如果类处于停怠状态，则不会截获对资源的访问。

您可以对以下类使用停怠类绕过：FILE、HOST、TCP、CONNECT 和 PROCESS。

访问者元素

每个用户由一个访问者元素 (ACEE) 代表（该元素是数据库中的用户记录在内存中的反映）。 CA Access Control 在登录过程中构建访问者元素。 访问者元素与用户的进程相关。 无论何时进程请求 CA Access Control 所保护的系统服务，或发出访问资源的暗示请求，CA Access Control 都会访问该资源的记录。 然后，它将确定以前创建的访问者元素中的信息（例如用户的安全级别、模式和组）是否允许用户访问该资源。

扩展本地安全性

下列 CA Access Control 功能可以扩展本地安全性。

超级用户帐户限制

管理操作系统的用户通常是在系统安装期间自动创建的预定义帐户的成员，例如 UNIX 系统中的 root 帐户及 Windows 系统中的管理员帐户。 每个预定义帐户的存在都是为了执行一组特定的系统功能。

担当 root 用户或管理员角色的用户可以执行多种任务，从创建、删除和修改用户到锁定、重新配置和关闭服务器。

这些操作系统中的主要安全风险之一是未经授权的用户可以获得对这些帐户的控制权。 如果发生这种情况，未经授权的用户可能会对系统造成巨大破坏。

通过 CA Access Control 可以限制授予这些帐户的权限，并限制作为用户组（这些帐户是该组成员）成员的用户的权限。 这将会减少操作系统的漏洞。

CA Access Control 管理员

安装 CA Access Control 时，系统会要求您命名一个或多个 CA Access Control 管理员。 CA Access Control 管理员有权修改整个或部分规则数据库。 您应该至少有一个具有完全权限的管理员。 该管理员可以自由修改或创建访问规则，并可以指定其他级别的管理员。

为系统定义用户后，可以通过向其他用户分配 ADMIN 属性来向他们分配管理权限。

注意：具有 ADMIN 属性的用户拥有强大的权限。 因此，应该严格限制 ADMIN 用户的数量。 在您设置一个或多个 CA Access Control 安全管理员后，将本地超级用户和 ADMIN 的角色分开，从超级用户删除 ADMIN 属性，这不失为一个好策略。

由于始终需要至少一个具有数据库管理权限的用户，因此 CA Access Control 不允许删除最后一个具有 ADMIN 属性的用户。

如果希望所有 CA Access Control 管理员都可以从该工作站管理其他主机，请确保该主机上的数据库中的规则向他们授予从该工作站读取和写入的访问权限。

子管理

CA Access Control 包含 *子层管理*功能。 通过该功能，管理员可授予特定权限，常规用户使用该权限可管理特定类。 这些用户则称为子管理员。

例如，可以允许特定用户只管理用户和组。

还可以通过不仅为特定类还为这些类中的特定对象授予访问权限，来指定更高级别的子层管理。

常规用户的管理权限

您可以使用 CA Access Control 向普通用户（即非管理员用户）授予必需的权限，以便这些用户不必成为管理员组的成员即可执行管理任务。 能够以这种精细方式通过授予管理权限指派任务是 CA Access Control 最重要的优点之一。

- SUDO 类中的记录存储了命令脚本，允许用户使用借来的权限运行该脚本。
- 数据属性值是命令脚本。 通过将可选脚本参数值添加到该值，可以对该值进行修改。

- SUDO 类中的每个记录都标识一个命令，一个用户可以借用另一个用户的权限来执行该命令。
- SUDO 类记录的关键字是 SUDO 记录的名称。 当用户执行 SUDO 记录中的命令时，会使用该名称来代替命令名称。

程序通路

*程序通路*是一种与文件关联的访问规则，该规则要求只能通过特定程序访问文件。 程序通路大大提高了敏感文件的安全性。 通过 CA Access Control 可以使用程序通路来为系统中的文件提供额外的保护。

B1 安全级别认证

CA Access Control 包括下列 B1“橙皮书”功能：安全级别、安全类别和安全标签。

- 可以向数据库中的访问者和资源分配一个*安全级别*。 安全级别是 1 至 255 之间的整数。 只有当访问者拥有的安全级别等于或大于分配给资源的安全级别时，访问者才可以访问资源。
- 数据库中的访问者和资源可以属于一个或多个*安全类别*。 只有当访问者属于分配给资源的所有安全类别时，访问者才能访问该资源。
- *安全标签*是将特定安全级别与一组（零个或更多）安全类别相关联的名称。 将用户分配给某一安全标签会授予该用户与该安全标签相关联的安全级别和所有安全类别。

注意：有关 B1 橙皮书功能的详细信息，请参阅*《实施指南》*。

端点管理

CA Access Control 提供了两种方式来管理企业中的资源并控制哪些用户可访问这些资源：

- **selang** - CA Access Control 命令语言。

  通过 selang 命令语言，您可以在 CA Access Control 数据库中进行定义。 selang 命令语言是命令定义语言。

  注意：有关使用 selang 的详细信息，请参阅*《selang 参考指南》*。

- **CA Access Control 端点管理** - 端点管理界面。

  使用基于 Web 的界面可以通过中央管理服务器管理远程端点。

  注意：有关安装 CA Access Control 端点管理 的详细信息，请参阅*《实施指南》*。

第 3 章： 管理用户和组

此部分包含以下主题：

用户和组

在 CA Access Control 中，每个操作和访问尝试都是代表负责提交该请求的用户执行的。 因此，系统中的每个进程都与特定的用户名相关联。 用户名将用户标识到 CA Access Control。

*用户*是能够登录的人员，也可以是批处理或后台程序的所有者。 在 CA Access Control 中，每次访问尝试都由用户执行。 CA Access Control 可以使用 CA Access Control 数据库及企业用户存储中的用户信息。 CA Access Control 在其数据库的 USER 记录或 XUSER 记录中存储用户信息。

注意：*企业用户存储*是操作系统中存储用户或组的存储，例如 UNIX 系统的 /etc/passwd 和 /etc/groups，或 Windows 的 Active Directory。

*组*是用户的集合。 组为组中的用户定义通用访问规则。 组可以嵌套（属于其他组）。 CA Access Control 可以使用 CA Access Control 数据库及企业用户存储中的组信息。 通常，基于角色（例如，database_administrators）创建组并向其分配用户。

用户记录是关键的访问者记录。 在 CA Access Control 中使用组的主要目的是同时向组中的所有用户分配访问权限。 同时分配访问权限比分别向每个用户分配访问权限更轻松且更不容易出错。

关于访问者的信息的存储位置

CA Access Control 使用的关于用户和组的信息存储在 CA Access Control 数据库和主机操作系统中。 主机操作系统信息存储称为*企业用户存储*或简称为*企业存储*。 默认情况下，将配置 CA Access Control 以便其不使用企业存储。 然而，您也可以配置 CA Access Control，以便 CA Access Control 找不到在其数据库中定义的用户或组时，会查找在企业存储中定义的用户和组成员资格并使用其中的信息。

注意：CA Access Control 可以使用企业存储中的信息，但仅当您在本地环境中使用 selang 命令时才能向企业存储中写入信息。

检查授权时，CA Access Control 始终在检查企业存储之前先检查在其自己的数据库中定义的访问者：如果您的企业用户的名称与在 CA Access Control 数据库中定义的用户名称相同，则 CA Access Control 将忽略企业用户。

CA Access Control 如何查找用户记录

用户登录后，CA Access Control 按以下顺序执行搜索，直至其找到与该用户关联的记录：

1. CA Access Control 搜索在其数据库中定义的用户。
2. CA Access Control 为具有该名称的企业用户搜索缓存。

   网络出现故障后，操作系统 (OS) 允许用户使用 OS 缓存凭据登录。 CA Access Control 缓存的目的是使 CA Access Control 在此类情况下也能使用企业用户的记录。
3. CA Access Control 使用操作系统为具有该名称的用户搜索企业用户存储。
4. 如果 CA Access Control 没有在其数据库或企业存储中发现与该用户关联的记录，那么 CA Access Control 会将 _undefined USER 记录中的属性分配给用户。

与企业用户存储集成

通常，将 CA Access Control 配置为使用在企业用户存储中定义的组和用户。

如果您这样配置了 CA Access Control，在默认情况下，当创建与企业用户或组相关的访问规则后或当用户登录到操作系统后，CA Access Control 会在其数据库中为该用户或组创建记录（如果先前没有记录）。这些记录具有类 XUSER（对于企业用户）或 XGROUP（对于企业组），并且具有 CA Access Control 强制执行访问规则所需的属性。 因为 CA Access Control 根据需要创建记录，所以无需对其进行管理。

CA Access Control 从企业用户存储提取的企业用户或组的属性仅是名称和组成员资格属性。

在企业存储中管理访问者的指南

如果您决定在企业用户存储中管理访问者，则应该仔细阅读以下部分中的指南。

必须在数据库中定义的用户和组

CA Access Control 需要在其数据库中定义某些用户和组，而不是在企业用户存储中定义。 这些信息包括：

- 预定义用户 (p. 34)
- 预定义组 (p. 35)
- 一个 CA Access Control 管理员
- 配置文件组
- 逻辑用户

企业用户使用限制

CA Access Control 强制实行以下企业用户使用限制：

- 如果 CA Access Control 中的企业用户与在数据库中定义的用户名称相同，则不能创建或引用该企业用户。
- 不能使用 selang AC 环境创建、删除或修改企业用户。

- 不能将企业用户用作逻辑用户。
- 默认情况下，不能在 CA Access Control 中创建企业用户，除非已经在企业用户存储中定义该用户。 但是，您可以在 UNIX 系统中启用或禁用此行为。

更多信息：

在 UNIX 中创建 XUSER 记录之前启用或禁用企业存储检查 (p. 32)

企业组使用限制

CA Access Control 强制实行以下企业组使用限制：

- 不能在 selang AC 环境中创建或删除企业组。
- 不能在 selang AC 环境中更改企业组的成员资格。
- 不能将企业组用作配置文件组 (p. 36)。

启用或禁用企业用户和组的使用

默认情况下，CA Access Control 不能使用在企业用户存储中定义的组和用户，但您可以让 CA Access Control 执行此操作。 我们建议您启用此功能，除非您需要与早期版本的 CA Access Control 兼容。

要让 CA Access Control 使用企业用户和组，请将配置设置 osuser_enabled 设置为 yes。 要禁用此行为，请将 osuser_enabled 的值设置为 no。

示例：在 Windows 中启用企业用户和组的使用

以下注册表设置可以在 Windows 中启用企业用户和组的使用：

- 注册表键：HKLM\SOFTWARE\ComputerAssociates\AccessControl\OS_user
- 名称：osuser_enabled
- 类型：REG_DWORD
- 值：yes

示例：在 UNIX 中启用企业用户和组的使用

以下命令可停止 CA Access Control、在 UNIX 中启用企业用户和组的使用以及重新启动 CA Access Control：

```
secons -s
seini -s OS_User.osuser_enabled yes
seload
```

在企业用户登录时启用或禁用 XUSER 记录的创建

如果启用 CA Access Control 对企业用户的使用，则默认情况下它会在用户登录时为该用户创建记录（在 XUSER 类中）。 有时您不需执行此操作，例如，如果每天成千上万的用户同时登录。

要防止 CA Access Control 在用户登录时创建 XUSER 记录，请将配置设置 create_user_in_db 的值更改为 0（零）。 要重新启用此行为，请将该值设置为 1（一）。

示例：在企业用户登录 Windows 时禁用 XUSER 记录的自动创建

以下注册表设置可以在 Windows 中禁用 CA Access Control 中企业用户记录的自动创建：

- 注册表键：
  HKLM\Software\ComputerAssociates\AccessControl\OS_user
- 名称：create_user_in_db
- 类型：REG_DWORD
- 值：0

示例：在企业用户登录 UNIX 时禁用 XUSER 记录的自动创建

以下命令可停止 CA Access Control、在 UNIX 中禁用 XUSER 记录的自动创建以及重新启动 CA Access Control：

```
secons -s
seini -s OS_User.create_user_in_db 0
seload
```

在 UNIX 中创建 XUSER 记录之前启用或禁用企业存储检查

有时，当未在企业用户存储中定义用户时您可能希望在 CA Access Control 中创建企业用户。 在 Windows 中，不能在 CA Access Control 中创建企业用户，除非该用户存在于 Windows 用户存储中。 在 UNIX 中，默认行为与 Windows 相反。 但是，在 UNIX 中，您可以启用或禁用此默认行为。

要禁用检查（以便在没有企业用户等同项时允许 CA Access Control 创建 XUSER 记录），请将配置设置 verify_osuser 的值更改为 0。 要强制执行检查，请将该值设置为 1。

示例：启用 XUSER 记录的创建而不检查企业用户存储

以下命令集可终止 CA Access Control、启用不具有企业存储等同项的 XUSER 记录的创建以及重新启动 CA Access Control：

```
secons -s
seini -s OS_User.verify_osuser 0
seload
```

Windows 中的循环企业存储帐户

*循环帐户*是已经删除然后又重新创建（使用相同名称）的企业存储用户或组。 从用户存储中删除一个用户（例如，当用户辞职时），然后为新用户创建一个与已删除的旧用户的名称相同的新帐户时，可能会产生循环帐户。

循环帐户存在安全问题，因为您不一定希望新访问者具有授予名称相同的旧帐户的那些访问权限。 要解决此问题，CA Access Control 授权要基于 SID。 这意味着当您创建新访问者，而该访问者名称与具有现有访问权限的已删除访问者的名称相同时，新访问者不自动接收旧访问者的旧权限。

重要说明！ 循环帐户访问者*不*继承旧的访问权限。 但是，依据涉及访问者名称（不是 SID）的数据库访问规则，可能看起来这些规则仍然适用。 使用 secons -checkSID 命令解决此问题。

在 Windows 中解析循环企业帐户

如果企业帐户（用户或组）具有相关联的数据库规则，然后被循环使用（删除然后使用相同名称创建），则可能看起来旧的数据库规则仍然适用于新帐户。 但是，由于 CA Access Control 授权基于 SID，这些规则不再适用，您需要为新组创建新规则。 可以创建新规则之前，您需要解析循环帐户。

要解析循环企业帐户，请打开命令提示符，然后运行以下命令：

```
secons -checkSID -users
secons -checkSID -groups
```

CA Access Control 处理它所具有的所有企业用户帐户（XUSER 记录），然后处理所有组帐户（XGROUP 记录），并标识 SID 与企业帐户的 SID 不同的帐户。 它使用以下命名约定在 CA Access Control 中重命名这些帐户：*SID* (*accountName*)

现在您可以为循环帐户创建新规则了。

注意：当用户登录或尝试访问资源时，将以此种方式解析循环用户帐户。 我们建议当您创建企业帐户时，将 secons -checkSID 命令作为排定任务运行。

示例：循环组帐户

公司 ABCD 在其企业存储中有个名为 *interns* 的组。 该组有九位成员，他们在从事 productA 的工作。 管理员使 CA Access Control 知道该组并为组成员分配访问文件所需的访问权限，如下所示：

```
nxg interns owner(msmith)
auth file c:\products\productA\materials\* xgid(interns) access(all)
auth file c:\HR\interns\* xgid(interns) access(read)
```

当 interns 在 ABCD 的使用期满后，企业存储管理员会删除该组。 三个月后会在企业存储中创建一个名称相同的新 interns 组，该组中有六位成员。 CA Access Control 数据库中的旧规则仍存在，因此看起来好像是新的 *interns* 组继承了旧组的权限。 但是，这些规则适用于旧的 interns 组，而 CA Access Control 管理员需要为新组创建新的规则。

要执行此操作，管理员需要识别并解析循环 interns 帐户，如下所示：

```
secons -checkSID -groups interns
```

此命令将 XGROUP 资源及参考该资源的任何访问规则重命名为“*SID* (*domain*\interns)”。 现在，管理员可以为从事 productB 工作的新 interns 组创建新的规则：

```
nxg interns owner(msmith)
auth file c:\products\productB\materials\* xgid(interns) access(all)
auth file c:\HR\interns\* xgid(interns) access(read)
```

注意：有关 secons 实用程序的详细信息，请参阅 *《参考指南》*。

数据库访问者

无论您决定如何管理您的用户，都必须在 CA Access Control 数据库中定义某些访问者，如以下部分中所述。

预定义用户

CA Access Control 预定义以下用户，您不能将其删除：

+devcalc

(Windows) CA Access Control 用于运行偏差计算进程 devcalc 的用户名。

_dms

_dms 安装在高级策略管理服务器组件的数据库（DMS、DH 读取程序和 DH 书写程序）中，policyfetcher 和 devcalc 使用 _dms 用户与 DH 和 DMS 进行通信。

nobody

nobody 用户是不能对应真实用户的用户记录。 使用此记录可以创建不授予任何用户相关联权限的规则。 例如，您可以将 *nobody* 设置为资源的所有者，这就是说任何用户都不能获得与拥有该记录相关联的权限。

+reportagent

CA Access Control 用于运行报告代理的用户名。

_seagent

_seagent 是 CA Access Control 用于运行某些内部进程的用户名，例如：

- PMDB 进程 sepmdd
- (UNIX) 偏差计算进程 devcalc
- 用户和组记录更新退出进程

_seagent 用户具有 SERVER 属性。

_sebuildla

(UNIX) _sebuildla 用户是 CA Access Control 运行 sebuildla 实用程序为 CA Access Control 后台进程 seosd 创建后备数据库的用户名。

_seoswd

(UNIX) _seoswd 是用户名，用于运行 seoswd watchdog 后台进程以监视程序（数据库中定义为受信程序）的文件信息和数字签名。

_undefined

_undefined 表示未在 CA Access Control 中定义的所有用户。可以使用 _undefined 在 ACL 中包括未定义的用户。

预定义组

CA Access Control 附带预定义组。 除了 _interactive 和 _network 组以外，您可以将用户以与添加到任何其他组相同的方式添加到这些组。

_abspath

如果用户在登录时位于 _abspath 组中，则该用户必须使用绝对路径名调用程序。

_interactive

属于 _interactive 组的用户仅可进行访问尝试。 如果用户登录到他们尝试访问的资源所在的主机，则这些用户是 _interactive 组的成员。 CA Access Control 动态并自动管理 _interactive 组的成员资格，您不能更改成员资格。

_network

这是 _interactive 的补充组。 属于 _network 组的用户仅可进行访问。 如果用户尝试访问的资源来自其他主机而不是资源所属的主机，则这些用户是 _network 组的成员。 CA Access Control 动态并自动管理 _network 组的成员资格，您不能更改成员资格。

_restricted

对于 _restricted 组中的用户，所有文件以及 Windows 注册表键都受 CA Access Control 的保护。如果文件或 Windows 注册表键没有显式定义访问规则，则将由该类（FILE 或 REGKEY）的 _default 记录控制访问权限。

注意：_restricted 组中的用户可能没有足够的授权开展工作。 如果您打算向 _restricted 组添加用户，请考虑在开始就使用警告模式。

_surrogate

如果用户使用 _surrogate 组的成员作为代理，则 CA Access Control 在代理操作审核跟踪中写入全部跟踪，并用原始用户名进行标记。

示例：使用 selang 将用户添加到 _restricted 组

以下 selang 命令将企业用户 john_smith 添加到 _restricted 组：

```
joinx john_smith group(_restricted)
```

配置文件组

*配置文件组*是在包含用户属性默认值的 CA Access Control 数据库中定义的组。 将用户分配到配置文件组后，配置文件组将向用户提供这些值，除非已经为用户设置这些值。

创建用户时可以为用户指定配置文件组，也可以稍后将用户分配到配置文件组。

通过配置文件组，管理员可以为分配给该组的任何新用户有效地创建带有特定权限的标准设置。 该设置可以指定以下内容作为该用户的主目录：审核属性、定义访问权限的 PMDB 以及影响配置文件组关联用户的各种密码规则。

CA Access Control 如何使用配置文件组确定用户属性

以下过程说明 CA Access Control 如何使用配置文件组确定用户属性。

1. CA Access Control 检查 USER 或 XUSER 类中的用户记录是否有针对属性的值。

   如果用户的记录有针对属性的值，CA Access Control 则使用该值。

2. CA Access Control 检查是否将用户分配到配置文件组。

   如果将用户分配到配置文件组，那么过程将继续。 如果未将该用户分配到配置文件组，那么 CA Access Control 将默认属性值分配给该用户。

3. CA Access Control 检查配置文件组是否有针对该属性的值。

   如果配置文件组有针对该属性的值，那么 CA Access Control 将该值分配给该用户。 如果配置文件组没有针对该属性的值，那么 CA Access Control 将默认值分配给该用户。

   注意：如果没有设置用户或配置文件组的审核属性，那么组的审核属性会影响用户的审核属性。

更多信息：

CA Access Control 如何为用户确定审核模式 (p. 166)

访问者管理

可以使用 CA Access Control 端点管理 或使用 selang 创建、修改和删除数据库或企业用户或组记录。

管理用户或组

如果要查看或修改特定访问者的属性，或者要删除访问者，必须先找到该访问者。

管理用户或组

1. 在 CA Access Control 端点管理 中，执行如下操作：
   a. 单击“用户”。
   b. 单击“用户”*或*“组”子选项卡。

   根据您的选择，将显示“用户”或“组”页面。
2. 在“搜索”区域中完成以下字段：

   用户/组名

   定义要查找的访问者的掩码。 您可以输入查找的访问者的全名，也可以使用掩码。 例如：使用 *admin* 列出名称包含“admin”的访问者。

   使用 *（星号）可列出所有访问者，使用 ?（问号）可代替单个字符。

   用户/组存储库

   指定要从中提取访问者列表的源。源可以是以下两者中的任意一个：

   - **内部帐户**－在 CA Access Control 数据库中定义的访问者。
   - **企业帐户**－在特定企业用户存储中定义的访问者。

仅显示 AC 帐户/配置文件

指定是否仅列出在 CA Access Control 数据库中有记录的帐户，如下所述：

- 如果选择了“内部帐户”，应用程序将仅列出存在于 CA Access Control 数据库中的帐户（无本地帐户）。
- 如果选择了“企业帐户”，应用程序将仅列出具有 CA Access Control 企业配置文件（XUSER 或 XGROUP 记录）的帐户。

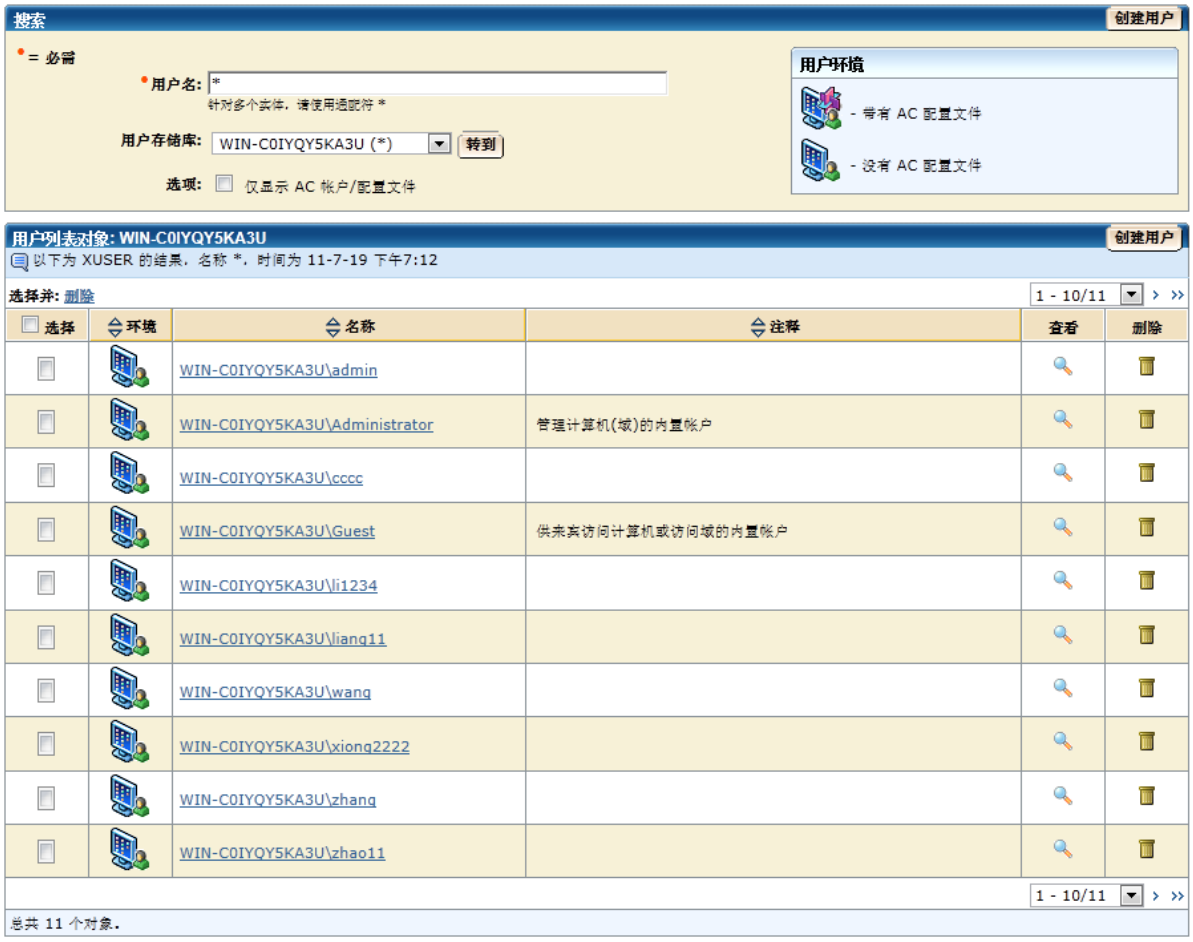
单击“执行”。

将显示您所选择的存储库中的访问者列表。

3. 请执行下列操作之一：
   - 单击“查看”列中的 以查看访问者的属性。
   - 单击“删除”列中的 以删除访问者。
   - 单击访问者的名称以修改访问者的属性。
   - 选择要删除的访问者，然后单击“删除”。
   - 单击“创建用户”或“创建组”以在 CA Access Control 数据库中创建用户或组记录。

示例：在存储库中搜索企业用户

以下图形向您显示了查找 ABC-DM1 企业用户存储中的所有用户的结果。



使用 selang 管理用户

将以下 selang 命令用于企业用户记录：

- **newxusr** 和 **editxusr** - 定义新的企业用户记录
- **chxusr** 和 **editxusr** - 更改企业用户的 CA Access Control 属性
- **find xuser** - 列出具有 CA Access Control 记录的企业用户
- **rmxusr** - 删除用户
- **show xuser** - 显示企业用户的 CA Access Control 属性

将以下 selang 命令用于 CA Access Control 数据库用户记录：

- **newusr** 和 **editusr** - 定义新的用户记录
- **chusr** 和 **editusr** - 更改用户的属性
- **rmusr** - 删除用户
- **find user** - 列出数据库用户
- **show user** - 显示用户的属性

示例：使用 selang 在数据库中定义用户

以下 selang 命令在 CA Access Control 数据库中定义了安全级别为 100 的新用户：

```
newusr internalUser level(100)
```

示例：使用 selang 更改企业用户的属性

以下 selang 命令向企业用户 Terry 授予了 AUDITOR 属性：

```
chxusr Terry auditor
```

使用 selang 管理组

除不能更改企业组的名称或成员资格外，您可以更改任意组的任意属性（从 CA Access Control 中）。

要更改组属性或分配与组相关联的访问权限，可以使用 CA Access Control 端点管理 或以下 selang 命令：

- **join[-]** 和 **joinx[-]**

  更改内部组的成员资格

  使用 join 将内部访问者添加到组。 使用 joinx 将企业组和用户添加到内部组。 使用命令的 -（减号）格式删除访问者。

- **editgrp**、**newgrp**、**chgrp**

  更改内部组的非成员资格属性

- **editxgrp**、**newxgrp**、**chxgrp**

  更改企业组的非成员资格属性

- **rmgrp**、**rmxgrp**

  删除用户组

示例：使用 selang 在数据库中定义组

以下 selang 命令在数据库中定义了新组“sales”。 该组的全名是“Sales Department”：

```
newgrp sales name('Sales Department')
```

示例：使用 selang 更改在数据库中定义的组的属性

以下 selang 命令使 CA Access Control 审核组 AC_admins 成员的所有事件：

```
chgrp AC_admins audit(all)
```

示例：使用 selang 将企业组添加到 ACL

以下 selang 命令将企业组 mygroup 添加到 myfile 的 ACL：

```
Authorize FILE (myfile) xgid(mygroup)
```

示例：使用 selang 将企业用户添加到在数据库中定义的组

以下 selang 命令将企业用户 mydomain\administrator 添加到在数据库中定义的组 AC_admins：

```
joinx mydomain\administrator group(AC_admins)
```

示例：使用 selang 将企业组添加到在数据库中定义的组

以下 selang 命令将企业组 Guests 添加到 _restricted 组：

```
joinx Guests group(_restricted)
```

第 4 章： 管理资源

此部分包含以下主题：

资源

*资源*是访问者可以访问且受访问规则保护的实体，或者是与该实体对应的 CA Access Control 数据库记录。 资源包括文件、程序、主机和终端等。

在 CA Access Control 中创建资源记录的主要目的是定义对与资源记录相对应的资源的访问权限。 访问资源所需的访问权限在资源记录的 Access Control 列表中指定。

资源组

*资源组*是包含其他资源列表的资源。 资源组是以下类之一的成员：CONTAINER、GFILE、GSUDO、GTERMINAL 或 GHOST。

由于资源组本身就是一种资源，因此它与其成员资源具有相同的属性。因此，使用资源组的优势就是简化了管理。 可以通过更改资源组的属性来更改所有成员资源的属性。

注意：在 Windows 上，检查用户对资源的授权时，CA Access Control 会考虑资源组所有权。 该操作在 12.0 中曾经介绍。 在先前版本中，授权进程只考虑资源的所有者。

例如，您使用没有所有者的默认访问权限来定义 FILE 资源。 而 FILE 资源是具有指定所有者的 GFILE 资源的成员。 在 CA Access Control r12.0 及更高版本中，命名的组所有者对该文件拥有完全访问权限。 在较早版本中，没有用户可以访问该文件。

类

在 CA Access Control 中，记录的*类*定义记录可以拥有的属性。 类中的所有记录具有相同的属性，尽管这些属性的值不同。

类包括：

- TERMINAL 类等。 该类包含终端的记录，例如 tty1、tty。
- FILE 类。 该类包含文件的记录。
- PROGRAM 类。 该类包含程序的记录。

每个记录均包含适用于记录类的属性的值。 例如，XUSER 类中的记录包括企业用户的位置和工作时间这样的属性，而 HOSTNET 类中的记录包括网络服务和 IP 地址数据这样的属性。

CA Access Control 包括预定义类。 还可以定义新类，称为用户定义的类。

类的默认记录

大多数类都可以包括指定该类资源访问类型的默认记录 (_default)，这些资源未在其自己的数据库记录中定义。

像其他资源记录一样，_default 记录可以包括 ACL 和 defaccess 字段。 您可以为除 USER、GROUP、CATEGORY、SECLABEL 和 SEOS 之外的所有类创建 _default 记录。

UACC 类（摒弃）

不再建议使用 UACC 类。 要为类中的记录指定默认值，请使用 _default 记录。

某些早期版本的 CA Access Control 将称为 UACC 的单独类用于与其他类的 _default 记录相似的记录。 现已不再建议使用 UACC 类，如果使用 _default 记录，则不检查 UACC 类中的同等记录。 在将来的版本中，可能不再支持 UACC 类。

例如，假设用户 Henderson 尝试终止进程 store_log。 CA Access Control 将按以下顺序检查授权。 主要问题是：是否在数据库中定义了进程 store_log？ CA Access Control 在数据库的 PROCESS 类中搜索名为 store_log 的记录。

- 如果未找到这种记录，则没有在 CA Access Control 中定义该进程。 在这种情况下，CA Access Control 会使用 PROCESS 类中的 _default 记录或 UACC 类中的 PROCESS 记录，以确定是否允许 Henderson 终止 store_log。
  - 如果用户 Henderson 出现在了 _default 记录的 ACL 中，则应用在其中指定的权限。
  - 如果 Henderson *未*出现在 _default 记录的 ACL 中，则应用在 _default 记录的 defaccess 属性中指定的权限。 该权限应用于 _default ACL 中未明确出现的所有用户。
- 如果数据库中定义了进程 store_log，则问题是用户 Henderson 是否出现在数据库进程 store_log 的 ACL 中。
  - 如果用户 Henderson 出现在进程 store_log 的 ACL 中，则应用在该处指定的权限。
  - 如果 Henderson *未*出现在 ACL 中，则 CA Access Control 应用在 store_log 资源的默认访问属性中指定的权限。 该权限称为资源的默认访问权限。

注意：如果 _default 的默认访问权限 (defaccess) 设置为 NONE，或者如果未指定 _default 且 UACC 类中相应资源的默认访问权限为 NONE，则拒绝任何尝试访问该类中未定义的资源的访问者访问资源。

如果 _default（或 UACC）的默认访问权限设置为最高权限（ALL，或在某些情况下为 READ 或 EXECUTE），则每个用户都可以访问未明确保护的任何资源。

预定义类

预定义类可以分为以下类型：

类类型	用途
访问者	定义访问资源的对象，例如用户和组
定义	定义对安全实体（例如安全标签和类别）进行定义的对象
安装	定义对 CA Access Control 的行为进行控制的对象
资源	定义访问规则所保护的对象

下表包含所有预定义类的列表。

类	类类型	说明
ADMIN	定义	使用该类可以将管理责任指派给不具有 ADMIN 属性的用户。 您可以为这些用户提供全局授权属性并限制他们的管理权限范围。
AGENT	资源	不适用于 CA Access Control
AGENT_TYPE	资源	不适用于 CA Access Control
APPL	资源	不适用于 CA Access Control
AUTHHOST	访问者	不适用于 CA Access Control
CALENDAR	资源	使用该类可以为实施时间限制的用户、组和资源定义 Unicenter TNG 日历对象。
CATEGORY	定义	使用该类可以定义安全类别。
CONNECT	资源	使用该类可以保护传出连接。 该类中的记录定义哪些用户可以访问哪些 Internet 主机。 激活 CONNECT 类之前，请确保数据流模块处于活动状态。
CONTAINER	资源	使用该类可以定义其他资源类中的一组对象，因此可以在某个规则适用于多个不同的对象类时简化定义访问规则的工作。
FILE	资源	使用该类可以保护文件、目录或文件名掩码。
GAPPL	资源	不适用于 CA Access Control
GAUTHHOST	定义	不适用于 CA Access Control

类	类类型	说明
GFILE	资源	该类中的每个记录定义一组文件或目录。 与将用户连接到组的方式一样，通过将文件或目录（FILE 类的资源）显式连接到 GFILE 资源，可完成分组。
GHOST	资源	该类中的每个记录定义一组主机。 与将用户连接到组的方式一样，通过将主机（HOST 类的资源）显式连接到 GHOST 资源，可完成分组。
GROUP	访问者	该类中的每个记录定义一个内部组。
GSUDO	资源	该类中的每个记录定义一个用户可以执行的一组命令（好像另一个用户正在执行一样）。 sesudo 命令使用该类。
GTERMINAL	资源	该类中的每个记录定义一组终端。
HNODE	定义	HNODE 类包含有关组织的 CA Access Control 主机的信息。 该类中的每个记录代表企业中的一个节点。
HOLIDAY	定义	该类中的每个记录定义用户需要额外权限才能登录的一个或多个时间段。
HOST	资源	该类中的每个记录定义一个主机。 主机由其名称或其 IP 地址标识。 对象包含确定本地主机是否可以从该主机接收服务的访问规则。 激活 HOST 类之前，请确保数据流模块处于活动状态。
HOSTNET	资源	该类中的每个记录均由 IP 地址掩码标识，且包含访问规则。
HOSTNP	资源	该类中的每个记录定义一组主机，其中，属于该组的主机都具有相同的名称模式。 每个 HOSTNP 对象的名称包含一个通配符。
LOGINAPPL	定义	LOGINAPPL 类中的每个记录定义一个登录应用程序，标识可以使用该程序进行登录的用户，并控制使用该登录程序的方式。
MFTERMINAL	定义	MFTERMINAL 类中的每个记录定义一个 CA Access Control 大型机管理计算机。
POLICY	资源	POLICY 类中的每个记录定义部署和删除策略所需的信息。 它包括指向 RULESET 对象的链接，而这些对象包含用于部署和删除策略的 selang 命令列表。
PROCESS	资源	该类中的每个记录定义一个可执行文件。
PROGRAM	资源	该类中的每个记录定义一个可与条件访问规则一起使用的受托程序。 受托程序是 Watchdog 进行监视以确保不被篡改的 setuid/setgid 程序。
PWPOLICY	定义	PWPOLICY 类中的每个记录定义一个密码策略。

类	类类型	说明
RESOURCE_DESC	定义	不适用于 CA Access Control
RESPONSE_TAB	定义	不适用于 CA Access Control
RULESET	资源	RULESET 类中的每个记录都定义一组用于定义策略的规则。
SECFILE	定义	该类中的每个记录定义一个不能更改的文件。
SECLABEL	定义	该类中的每个记录定义一个安全标签。
SEOS	安装	该类中的一个记录指定活动的类和密码规则。
SPECIALPGM	安装	SPECIALPGM 类中的每个记录对 Windows 中的备份、DCM、PBF 和 PBN 功能或 UNIX 中的 xdm、备份、邮件、DCM、PBF 和 PBN 程序进行注册，或将需要特殊授权保护的应用程序与逻辑用户 ID 关联。 这样便可根据所执行的操作而不是执行该操作的人员来设置访问权限。
SUDO	资源	sesudo 命令使用该类来定义一个用户（如常规用户）可以执行的命令（好像另一个用户（如 root 用户）正在执行一样）。
SURROGATE	资源	该类中的每个记录包含访问者的访问规则，此访问规则定义了谁可以将该访问者用作代理。
TCP	资源	该类中的每个记录定义一个 TCP/IP 服务，例如邮件或者 http 或 ftp。
TERMINAL	资源	该类中的每个记录定义一个终端（用户可以从其登录的设备）。
UACC	资源	定义每个资源类的默认访问规则。
USER	访问者	该类中的每个记录定义一个内部用户。
USER_ATTR	定义	不适用于 CA Access Control
USER_DIR	资源	不适用于 CA Access Control
XGROUP	资源	该类中的每个记录在 CA Access Control 中定义一个企业组。
XUSER	资源	该类中的每个记录在 CA Access Control 中定义一个企业用户。

注意：默认情况下，CA Access Control 数据库类 TCP 和 SURROGATE 处于非活动状态。

如果您从 TCP 类处于活动状态的早期版本升级，但是没有任何 TCP 记录且没有更改 _default TCP 资源，CA Access Control 则会在升级期间停用该类。 对于 SURROGATE 类同样如此。

如果您从 SURROGATE 类处于活动状态的早期版本升级，并且已经定义了 SURROGATE 记录或已经更改了 SURROGATE 记录的默认值，CA Access Control 则会在升级之后保留 SURROGATE 类配置。 该类仍然保持活动状态，而内核模式截获保持启用状态。

注意：有关 CA Access Control 类的详细信息，请参阅 *《selang 参考指南》*。

用户定义的类

您可以使用 CA Access Control 定义新类，以便可以通过为抽象对象创建适当的记录来保护抽象对象。

示例：用于数据库视图的用户定义的类

站点可以使用数据库存储和显示专有数据。

您可以定义用户定义的类 DATABASE_VIEWS，并将每个数据库视图定义为该类的资源成员。 为资源提供一个创建该数据库视图所需的定义访问权限的 ACL。 当用户尝试创建数据库视图时，CA Access Control 会检查该用户的访问权限，并基于 ACL 允许或禁止创建。

用户定义的类资源中的通配符

通过在用户定义的类中的资源名称中使用通配符，您可以创建对应于多个物理资源的资源记录：名称与通配符模式匹配的所有物理资源均受到与资源记录相关联的访问权限的保护。

可用通配符如下：

- * 表示任意数量的任意字符
- ? 表示任意单个字符

如果物理资源名称与多个资源记录名称相匹配，则最长的非通配符匹配项将用于该资源。

CA Access Control 不接受以下通配符模式作为资源名称：

- *
- /*
- /tmp/*
- /etc/*

用户定义的类 - 示例

假设您的系统为银行服务，并且您想要保护帐户间大笔金额的转帐，则可以根据以下概述来设置该安全性。

1. 定义包含描述转帐、调用（例如 TRANSFERS）的记录的类。
2. 对于您可能希望保护的每次资金转帐，请在 TRANSFERS 类中定义记录。

   例如，可以定义名为 Upto.$1K、Upto.$1M、Upto.$10M 和 Over.$10M 的记录。

   将您希望控制转帐的任何其他资源定义为 TRANSFERS 类的成员。
3. 要给予不同用户执行不同的最大金额转帐的权限，可以批准或拒绝他们访问 TRANSFERS 类中的各个记录。
4. 此外，要处理编程传输，在银行的资金转帐程序中插入对 CA Access Control API 的调用，从而 API 会在检查用户的权限后才允许转帐进行。

第 5 章： 管理授权

此部分包含以下主题：

访问权限

CA Access Control 的主要目的是分配和强制执行访问权限。

访问权限始终具有以下组件：

- 所访问的资源，例如，文件、主机或终端
- 访问的类型，例如读取、写入、删除、登录、运行
- 访问者，可以是用户也可以是组

如果以下一种或多种情况属实，则用户有权以某种方式访问资源：

- 用户具有由资源 ACL 授予的访问权限
- 用户是具有访问权限的组的成员。
- 用户运行具有访问权限的程序。 例如，用户具有在 SPECIALPGM 类中运行程序的权限，或者在 SUDO 类中运行命令的权限。

注意：有关按类的访问权限的详细信息，请参阅 *《selang 参考指南》*。

设置访问权限 - 示例

示例：向内部用户授予读取访问权限

以下 selang 命令将内部用户 internal_user 添加到终端 tty30 的 ACL，向终端授予读取访问权限：

```
authorize TERMINAL tty30 access(READ) uid(internal_user)
```

示例：向企业用户授予读取访问权限

以下 selang 命令将企业用户 Terry 添加到终端 tty30 的 ACL，向终端授予读取访问权限：

```
authorize TERMINAL tty30 access(READ) xuid(Terry)
```

示例：更改企业用户对资源的访问权限

以下 selang 命令将 Terry 对终端 tty30 的访问权限设置为 none，因此拒绝了 Terry 的访问：

```
authorize TERMINAL tty30 access(NONE) xuid(Terry)
```

示例：从资源中删除企业用户的访问权限

以下 selang 命令从终端 tty30 中的 ACL 中删除 Terry：

```
authorize- TERMINAL tty30 xuid(Terry) access-
```

Terry 现已具有对终端的默认访问权限。

示例：向企业用户授予子管理员访问权限

以下 selang 命令将企业用户 Terry 设置为具有管理用户和文件权限的子管理员：

```
authorize ADMIN USER xuid(Terry)
authorize ADMIN FILE xuid(Terry)
```

访问控制列表

对资源的访问权限在 Access Control 列表中指定。 每个资源记录至少具有两个 Access Control 列表：

ACL

指定被授予资源访问权限的访问者及其被授予的访问权限的类型。

NACL

指定被拒绝授权资源访问权限的访问者及其被拒绝的访问权限的类型。

访问权限还取决于访问权限所在的环境，例如用户是否是在本地登录。

条件访问控制列表

条件 Access Control 列表 (CACL) 提供了 ACL 的扩展。 当访问者尝试访问资源时，如果资源的 ACL 和 NACL 未为该用户定义访问权限，则 CA Access Control 将检查条件 Access Control 列表。

条件 Access Control 列表以一种特定方式指定对所访问资源的访问权限，例如通过使用指定的程序。

例如，您可以使用条件 Access Control 列表定义程序通路规则。

CA Access Control 允许使用以下条件 Access Control 列表：

- 程序 Access Control 列表 (PACL)
- TCP 类 Access Control 列表
- CALENDAR 类 Access Control 列表

要在条件 Access Control 列表条目中定义一个条目，可以使用 selang authorize 命令的 via 选项。

与其他 Access Control 列表一样，条件 Access Control 列表中的每个条目指定被授予访问资源权限的访问者及其被授予的访问权限的类型。 此外，条件 Access Control 列表中的条目还指定分配权限的条件。 对于 PACL，条件是程序的名称，访问者需要运行该程序才能具有访问权限。

示例：使用 PACL

要使企业用户 sysadm1 仅可通过运行程序 secured_su 成为超级用户，您可以使用以下 selang 命令指定相应的条件访问规则：

```
authorize SURROGATE user.root xuid(sysadm1) via(pgm(secured_su))
```

defaccess - 默认访问字段

资源的记录可以包括默认访问字段 defaccess。 defaccess 字段的值指定允许任何资源 Access Control 列表中均未包含的访问者使用的访问权限。

如何确定对资源的访问权限

当访问者尝试访问资源时，CA Access Control 通过按照预先确定的顺序运行一次或多次检查来检查访问权限，直至其获得结果。 如果在任何一次检查中生成了访问结果（拒绝或允许访问），则 CA Access Control 不会进一步执行检查而是返回结果。

CA Access Control 运行这些检查的顺序很重要。 默认情况下，对于每项资源，CA Access Control 均按以下顺序检查访问记录：

1. 基于资源时间的限制
2. 资源的所有权（允许所有者进行访问）
3. B1 检查
4. 资源的 NACL
5. 资源的 ACL
6. 资源的 PACL
7. 资源的 defaccess 字段

最后两项检查的顺序由 accpacl 选项的设置确定。 您可以通过使用 selang 命令设置选项 setpacl- 来禁用对资源 PACL 的使用。

一个 Access Control 列表可以包含多个影响用户的条目。 例如，可以包含与用户显式相关的条目，还可以包含用户所属的每个组的条目。 CA Access Control 在进入下个级别之前检查每个级别所有可能的条目。 有关 CA Access Control 如何解决每个级别的冲突规则的详细信息，请参阅《用户和组访问权限之间的互动》 (p. 55)。

示例：对文件的结果权限

在下表中，假设名为 user1 的访问者尝试读取资源 file1。

在下表中，CA Access Control 按照 accpacl 选项的默认设置使用 PACL。

NACL 中用于 user1 的条目	ACL 中用于 user1 的条目	PACL 中用于 user1 的条目	defaccess 中的条目	结果权限
读取	*(Any)*	*(Any)*	*(Any)*	拒绝读取权限
(Not defined)	无	*(Any)*	*(Any)*	拒绝读取权限
(Not defined)	读取	*(Any)*	*(Any)*	授予读取权限

NACL 中用于 user1 的条目	ACL 中用于 user1 的条目	PACL 中用于 user1 的条目	defaccess 中的条目	结果权限
(Not defined)	*(Not defined)*	via pgm securereader	*(Any)*	允许通过 securereader 程序读取
(Not defined)	*(Not defined)*	*(Not defined)*	读取	授予读取权限

如果条目显示为 *(Not defined)*，则表示 Access Control 列表中不存在用于 user1 的条目。

如果条目显示为 *(Any)*，则表示该 Access Control 列表中的条目无关紧要，因为 CA Access Control 不对其进行检查。

CA Access Control 检查的顺序为从左到右。 请注意，对于所有行来说，具有定义的访问权限的单元格右侧的单元格均具有值 *(Any)*。 相反，包含定义的访问权限的单元格左侧的所有单元格均具有值 *(Not defined)*。

用户和组访问权限之间的互动

您可以向用户以及用户所属的组显式授予或拒绝访问权限。 有时这些权限可能会有冲突。 下例显示了如果在用户属于两个组（组 1 和组 2）时向同一资源分配有冲突的访问权限，会出现怎样的结果。

假设已设置“累积组权限” (p. 56)选项（默认设置）。

用户的访问权限	组 1 的访问权限	组 2 的访问权限	结果访问权限
访问被拒绝	*(Any)*	*(Any)*	访问被拒绝
访问已授权	*(Any)*	*(Any)*	访问已授权
(Not defined)	访问已授权	*(Not defined)*	访问已授权
(Not defined)	*(Not defined)*	访问已授权	访问已授权
(Not defined)	访问已授权	访问已授权	访问已授权
(Not defined)	访问被拒绝	*(Any)*	访问被拒绝
(Not defined)	*(Any)*	访问被拒绝	访问被拒绝

如果条目显示为 *(Not defined)*，则表示没有为用户或组定义任何条目。

如果条目显示为 *(Any)*，则表示访问权限无关紧要，因为 CA Access Control 不对其进行检查。

累积组权限 (ACCGRR)

*累积组权限*选项 (ACCGRR) 影响 CA Access Control 检查资源的 ACL 的方式。 如果启用 ACCGRR，则 CA Access Control 会检查 ACL 以获得用户所属的所有组授予的权限。 如果禁用 ACCGRR，则 CA Access Control 会检查 ACL 以查看是否有任何可应用的条目包含值 none。 如果有，则会拒绝访问。 否则 CA Access Control 将忽略所有组条目，Access Control 列表中的第一个可应用的条目除外。 默认情况下，该选项为启用。

要启用 ACCGRR 选项，可以使用以下 selang 命令：

```
setoptions accgrr
```

要禁用 ACCGRR 选项，可以使用以下 selang 命令：

```
setoptions accgrr-
```

安全级别、类别和标签

安全级别和安全类别提供了限制资源访问的其他方式，这是对 Access Control 列表用法的补充。

安全标签是一种将安全级别和类别捆绑在一起的方式，从而更轻松地对其进行管理。

安全级别

*安全级别*是可以分配给访问者和资源的 0 到 255 之间的整数。 如果访问者的安全级别小于分配给资源的安全级别，即使在资源的 Access Control 列表中向用户授予了访问权限，访问者也不能访问资源。 如果资源的安全级别为零，将不对该资源进行安全级别检查。

安全级别为零的访问者不能访问安全级别不为零的任何资源。

安全类别

*安全类别*是 CATEGORY 类中记录的名称。 可以向访问者和资源分配安全类别。 只有当将访问者分配给向资源分配的所有安全类别时，访问者才能访问该资源。

安全标签

*安全标签*是 SECLABEL 类中记录的名称。 安全标签将安全级别和一组安全类别捆绑在一起。 将安全标签分配给访问者或资源，会授予该访问者或资源与该安全标签相关联的组合的安全级别和安全类别。 安全标签会覆盖访问者或资源中任何特定的安全级别和类别分配。

示例：使用安全标签 High_Security

假设 High_Security 是包含安全级别 255 与安全类别 MANAGEMENT 和 CONFIDENTIAL 的安全标签。

如果将用户 user1 分配给安全标签 High_Security，则 user1 的安全级别为 255，另外还具有安全类别 MANAGEMENT 和 CONFIDENTIAL。

第 6 章： 保护帐户

此部分包含以下主题：

为什么要保护帐户？

用户帐户通常是密码攻击的对象。 Root 帐户保护包括监视替换用户 (su) 请求以及使用 Surrogate DO (SUDO) 工具，后者解决超级用户权限两难的处境。 CA Access Control 提供一个两级的密码保护系统：serevu（撤销用户后台进程）和 PAM（可插入身份验证模块）。 您还可以通过指定在用户处于停怠状态一段时间后自动锁定来保护帐户。

安全用户替换

通过 UNIX su 命令，用户可以使用目标用户的密码切换到其他用户。 要切换用户 ID 的用户必须记住目标用户的密码，记下密码，或者要求目标用户使用简单的密码。 这违反了一些密码策略。 另外，su 命令并不记录谁调用的该命令，因此假扮某个帐户所有者的用户与实际的所有者难以区分。

CA Access Control 包括 sesu 实用程序，它是 UNIX su 命令的增强版本。 您可以配置 sesu 以提示用户输入自己的密码作为身份验证的方式，而不是提示输入目标用户的密码。 身份验证过程基于在 SURROGATE 类中定义的访问规则，并且还基于执行命令的用户密码（可选）。

不同于对 su 的许可，对 sesu 的许可不依赖于是否知道目标用户的密码。 而是依赖于数据库中指定的许可；因为记录了用户的登录身份，所以用户需要对自己的操作负责。

如果某用户是 _surrogate 组中用户之一的代理，则 CA Access Control 会将对该用户操作的全部跟踪作为对新用户的跟踪发送到审核跟踪。

为了避免不经意使用该程序，在文件系统中对 sesu 进行了标记，没有任何用户可以运行它。 安全管理员必须将该程序标记为可执行并将 setuid 设为 root，您才能使用该程序。

重要说明！ 在您使用 sesu 实用程序前向 CA Access Control 数据库定义所有用户并设置 sesu 先决条件。 这样可以避免未定义到 CA Access Control 的用户打开整个系统。

设置用户 ID 替换规则

为防止或允许用户替换其他用户，您需要设置用户 ID 替换规则。 这些规则通过 SURROGATE 类资源进行管理。 要定义任何用户替换规则，您需要创建 SURROGATE 记录。

设置用户 ID 替换规则

1. 在 CA Access Control 端点管理 中，单击“用户”选项卡，然后单击“授权和指派”子选项卡。

   “授权和指派”菜单选项将显示在左侧。

2. 单击“用户 ID 替换”。

   将显示“用户 ID 替换”页面。

3. 单击“创建用户 ID 替换”。

   将显示“创建用户 ID 替换”页面。

4. 完成选项卡式页面中的字段，然后单击“保存”。

   注意：有关 SURROGATE 类属性的详细信息，请参阅 *《selang 参考指南》*。

如何为用户替换安装 sesu

默认情况下，在文件系统中对 sesu 实用程序进行了标记，没有人可以运行它。 为了保证安全使用，您必须在设置完数据库规则后再将 sesu 设为用户可用。 然后您需要锁定系统的 su 实用程序，这样用户只能转为使用 CA Access Control 的 sesu 实用程序。

要安装 sesu，请执行以下操作：

1. 设置基本用户替换规则 (p. 61)。
2. 使用 CA Access Control 的 sesu 实用程序替代系统的 su 实用程序 (p. 61)。
3. 避免用户运行系统的 su 实用程序 (p. 64)。

注意：完成该安装后，运行 CA Access Control 时系统的 su 实用程序将不会执行，因而用户将不得不使用安全的 sesu 实用程序。 CA Access Control 没有运行时，系统的 su 实用程序将正常工作。

设置基本用户替换规则

开始使用 sesu 实用程序之前，您应在数据库中设置某些常用用户替换规则。 这些规则防止出现未知用户替换特权用户帐户这种不希望的情况，并允许特定用户和进程执行必要的用户替换活动。

设置基本用户替换规则

1. 为 root 用户 (USER.root) 创建具有以下属性的代理资源：
   - *nobody* 作为所有者
   - 默认访问权限为 *none*
   - 所有的管理员应拥有完全控制权

   这将防止所有用户替换 root 用户，除非显式授权。 显式授权所有管理员替换 root 用户。

   注意：您可以分别授权给个别管理员，也可以使用管理员组授权给所有管理员。
2. 为 root 用户组 (GROUP.other) 创建具有以下属性的代理资源：
   - *nobody* 作为所有者
   - 默认访问权限为 *none*
   - 所有的管理员应拥有完全控制权

   这将防止所有用户替换 root 用户组，除非显式授权。 显式授权所有管理员替换 root 用户组。

   注意：大部分 UNIX 系统上的 root 用户组是 *other* 或 *sys*。

3. 按如下所示更改 USER._default 的用户替换规则：

   - *nobody* 作为所有者
   - 默认访问权限为 *none*
   - 授权 root 用户替换任何未定义的用户
   - 授权管理员组替换任何未定义的用户

   这将防止所有用户替换任何组（除非显式授权），并且授权 root 用户和 root 用户组替换任何用户（除非显式拒绝）。

   注意：需要特别授权 root 用户，才能允许程序（例如 dtlogin）将会话所有权从 root 用户（默认 X 窗口所有者 uid=0）切换到任何其他用户。 否则，尝试登录将失败，因为 CA Access Control 会阻止任何未经显式授权的用户替换活动。

4. 按如下所示更改 GROUP._default 的组替换规则：

   - *nobody* 作为所有者
   - 默认访问权限为 *none*
   - 授权 root 用户替换任何未定义的组
   - 授权管理员组替换任何未定义的组

   这将防止所有用户替换任何组，除非显式授权，并授权 root 用户和 root 用户组替换任何组，除非显式拒绝。

示例：使用 selang 设置基本用户替换规则

使用以下 selang 命令在您的环境中设置基本用户替换规则：

```
nr surrogate USER.root defacc(n) own(nobody)
auth surrogate USER.root gid(sys_admin_GID) acc(a)
nr surrogate GROUP.other defacc(n) own(nobody)
auth surrogate GROUP.other gid(sys_admin_GID) acc(a)
cr surrogate USER._default defacc(n) own(nobody)
cr surrogate GROUP._default defacc(n) own(nobody)
auth surrogate USER._default uid(root) acc(a)
auth surrogate GROUP._default uid(root) acc(a)
auth surrogate USER._default gid(sys_admin_GID) acc(a)
auth surrogate GROUP._default gid(sys_admin_GID) acc(a)
```

使用 CA Access Control 的 sesu 实用程序替代系统的 su 实用程序

默认情况下，在文件系统中对 sesu 实用程序进行了标记，没有人可以运行它。 为允许用户使用 sesu 实用程序替换其他用户，您必须首先启用 sesu 并使用它替代系统的 su 实用程序。

使用 CA Access Control 的 sesu 实用程序替代系统的 su 实用程序

注意：您需要成为 root 用户或其他授权用户才能执行以下步骤。

1. 使用以下命令允许用户运行 sesu 实用程序：

   ```
   chmod +s /opt/CA/AccessControl//bin/sesu
   ```

2. 使用以下命令找出系统 su 实用程序的位置：

   ```
   which su
   ```

3. 使用以下命令重新命名系统的 su 实用程序：

   ```
   mv su_dir/su su_dir/su.ORIG
   ```

   其中 *su_dir* 是 su 所在的目录。

4. 将 sesu 实用程序链接到 su 命令：

   ```
   ln -s /opt/CA/AccessControl//bin/sesu su_dir/su
   ```

   这样就可以允许用户继续使用 su 命令，虽然正在运行的是 sesu 实用程序。

5. 使用以下命令停止 CA Access Control：

   ```
   secons -s
   ```

6. 使用以下命令修改 CA Access Control 配置设置：

   ```
   seini -s sesu.SystemSu su_dir/su.ORIG
   seini -s sesu.UseInvokerPassword yes
   ```

   设置标记 SystemSu 是为了便于 sesu 在 CA Access Control 未运行时调用原始系统的 su 实用程序。

   设置标记 UseInvokerPassword 是为了让 CA Access Control 提示用户输入他们的原始密码，而不是 root 用户密码或另一个用户的密码。重新通过身份验证后，才能允许用户进行用户替换。

7. 使用以下命令重新加载 CA Access Control：

   ```
   seload
   ```

避免用户运行系统的 su 实用程序

虽然 sesu 实用程序已进行了配置，但所有用户还是可以使用 root 用户的密码或某个用户的密码运行 su.ORIG（重新命名的系统 su 实用程序）。为了避免这种情况，可以使用 PROGRAM 类显式阻止在 CA Access Control 运行时执行 su.ORIG。

注意：如果您在安装和配置 CA Access Control 时使用了 seuidpgm，则不必执行该步骤。 su 将不会运行，因为它已被修改（重新命名为 su.ORIG）。

避免用户运行系统的 su 实用程序

1. 在 selang 中，使用以下命令设置 CA Access Control 监控已重新命名的 su 实用程序：

   ```
   nr program su_dir/su.ORIG defacc(x) own(nobody)
   ```

2. 以 root 用户的身份登录，使用以下命令更改文件访问和修改时间：

   ```
   touch su_dir/su.ORIG
   ```

   CA Access Control 将监视 su.ORIG，并且因为已经使用 *touch* 命令对文件进行了更改，CA Access Control 还将阻止执行该文件。

设置 Surrogate DO 工具

操作员、生产人员和最终用户通常需要执行只有超级用户才能执行的任务。 这些任务包括下面的内容：

- 挂接 CD-ROM
- 使用备份脚本
- 设置打印机

传统的解决方案是向所有这些用户提供超级用户密码，这会危及到站点的安全。 安全的替代方法（即保持密码的保密性）会使系统管理员因处理用户要执行例行任务的合法请求而导致负担过重。

Surrogate DO (sesudo) 实用程序解决了这个难题。 它允许用户执行 SUDO 类（其中每条记录包含一个脚本）中定义的操作，指定哪些用户和组可以运行脚本，以及基于这一目的借给他们必要的权限。

例如，要以用户 root 的身份定义一个挂接 CD-ROM 的 SUDO 资源，请输入以下命令：

```
newres SUDO MountCd data('mount /usr/dev/cdrom /cdr') targuid(root)
```

该 newres 命令将 MountCd 定义为受保护的操作，某些获得 root 权限的用户才可执行。 该示例使用 targuid(root) 参数指出 root 是目标用户（借出权限的用户）的 ID。 实际上，在此示例中不需要该参数，因为 root 是 SUDO 记录的默认目标 ID

重要说明！ 请在数据属性中使用完整的绝对路径名。 相对路径名可能会意外地执行未受保护的目录中植入的特洛伊木马程序。

此外，通过使用 authorize 命令，可以授权用户执行 MountCd 操作。 例如，要允许用户 *operator1* 挂接 CD-ROM，请输入下列命令：

```
authorize SUDO MountCd uid(operator1)
```

还可以使用 authorize 命令明确防止用户执行受保护的操作。 例如，要防止用户 *operator2* 挂接 CD-ROM，请输入下列命令：

```
authorize SUDO MountCd uid(operator2) access(None)
```

运行 sesudo 实用程序将执行受保护的操作。 例如，使用下列命令，用户 *operator1* 将挂接 CD-ROM：

```
sesudo MountCd
```

sesudo 工具首先检查是否授权用户执行 SUDO 操作，然后，如果授权用户使用资源，则执行资源中定义的命令脚本。 在我们的示例中，sesudo 检查 operator1 是否有权执行 MountCd 操作，然后调用命令 mount /usr/dev/cdrom /cdr。

如果您希望 sesudo 在执行前请求用户的密码，请使用包括 PASSWORD 参数的命令来定义或修改 SUDO 记录。 如果不使用该参数，则用户能否执行该命令只取决于 SUDO 对象的访问规则。

注意：有关 sesudo 实用程序与管理 SUDO 记录（editres 命令）的详细信息，请参阅 *《参考指南》*。

定义 SUDO 记录

SUDO 类中的记录存储命令脚本，这样，用户便可以通过借用的权限来运行该脚本。 借用权限的能力由 SUDO 记录以及执行这些脚本的 sesudo 命令严格控制。

在 SUDO 记录中，注释属性用于特殊用途，通常也称为**数据属性。**

data 属性的值是命令脚本，也可以在其中添加一个或多个要禁用或允许的脚本参数值。 整个 data 属性值必须放在单引号中，并且应该使用可执行文件的完整路径名称引用可执行文件，以防止特洛伊木马获取它们的位置。

以下是 data 属性的格式：

```
data('cmd[;[prohibited-values][;permitted-values]]')
```

因为禁用和允许的值的列表是可选的，所以一个简单的 data 属性值可能是：

```
newres SUDO MountCd data('mount /dev/cdrom /cdr')
```

命令中的简单值表明命令 sesudo MountCd 执行脚本 mount /dev/cdrom /cdr。 不禁用特定脚本参数值；允许使用所有脚本参数值。

使用通配符和功能强大的变量，可以灵活地指定禁用的参数和允许使用的参数。 可以使用的通配符是标准的 UNIX 通配符。 变量如下表所示：

变量	说明
$A	字母值
$G	现有 CA Access Control 组名
$H	用户的主路径模式
$N	数字值
$O	执行者的用户名
$U	现有 CA Access Control 用户名
$e	不带任何参数的 SUDO 命令
$f	当前文件名
$g	当前 UNIX 组名
$h	当前主机名
$r	具有 UNIX 读取权限的当前 UNIX 文件名
$u	当前 UNIX 用户名
$w	具有 UNIX 写入权限的当前 UNIX 文件名
$x	具有 UNIX 执行权限的当前 UNIX 文件名

如果将*禁用*参数值的列表附加到脚本，请执行以下操作：

- 用分号分隔脚本和禁用的参数值，但将它们都保留在单引号内。 例如，如果要禁止用户使用 -9，但允许用户使用所有其他参数，请输入下面的命令：

  ```
  newres SUDO scriptname data('cmd;-9')
  ```

  其中 *cmd* 代表您的脚本。

  另外，如果不允许使用任何参数值，但希望默认使用所有参数，请按以下方式定义 SUDO 记录：

  ```
  newres SUDO scriptname data( 'cmd;*' )
  ```

- 如果脚本参数有多个禁用的值，请使用空格字符作为分隔符。 例如，如果要禁止用户使用 -9 和 -HUP，但却允许用户使用所有其他参数，请输入下面的命令：

  ```
  newres SUDO scriptname data('cmd;-9 -HUP')
  ```

- 如果多个脚本参数有禁用值，请使用管道符 (|) 作为各组禁用值之间的分隔符。 例如，如果要禁止用户将 -9 和 -HUP 用作脚本的第一个参数并禁止其将任何当前的 UNIX 用户名用作第二个参数（请参阅前面的变量列表），请输入下面的命令：

  ```
  newres SUDO scriptname data('cmd;-9 -HUP | $u')
  ```

  如果脚本的参数多于列表中的参数，则最后一组禁用参数将适用于所有其余参数。

如果将*允许*参数值的列表附加到脚本：

- sesudo 实用程序强制执行两次检查：**不仅参数值不能匹配任何相应的禁用值，而且它们还必须至少匹配一个相应的允许值。**
- 用分号将*允许*值的列表与*禁用*值的列表分开，但要将它们都保留在单引号内。 即使您没有禁用值的列表，仍需要使用分号；否则，将禁用您要允许的值。 例如，如果仅允许值 NAME 作为脚本的参数值，请输入下面的命令：

  ```
  newres SUDO scriptname data(cmd;;NAME)
  ```

- 正如在另一个列表中一样：
  - 如果脚本参数有多个允许值，请使用空格字符作为分隔符。
  - 如果多个脚本参数具有允许值，请使用管道符 (|) 作为各组允许值之间的分隔符。

  例如，如果您有两个参数，第一个参数必须是数字型但不得是 UNIX 用户名，第二个参数必须是字母型但不得是 UNIX 组名，请输入以下命令：

  ```
  newres SUDO scriptname data( 'cmd; $u | $g ; $N | $A' )
  ```

  如果脚本的参数多于列表中的参数，则最后一组允许使用的参数将适用于所有剩余参数。

因此，data 属性的完整格式为：**首先是脚本，然后是逐个参数的禁用值，再然后是逐个参数的允许值：**

```
data( 'cmd;
 param1_prohib1 param1_prohib2 ... param1_prohibN | \
 param2_prohib1 param2_prohib2 ... param2_prohibN | \
 ...
 paramN_prohib1 paramN_prohib2 ... paramN_prohibN ; \
 param1_permit1 param1_permit2 ... param1_permitN | \
 param2_permit1 param2_permit2 ... param2_permitN |
 ...
 paramN_permit1 paramN_permit2 ... paramN_permitN')
```

防止密码攻击

最常见的未经授权的访问类型是黑客猜测密码进行的访问。 CA Access Control 提供了两个检测和防止密码攻击的工具：serevu 和 pam_seos。

防止密码攻击的另一个方法是通过设置密码策略规则来控制您的环境中使用的密码。

serevu

serevu 后台程序锁定那些执行登录尝试的次数比指定的登录尝试次数多的用户的帐户。 这可以通过拒绝进一步的帐户进入尝试来防止潜在的密码攻击，还可以防止“词典攻击”。

通常，使用用户锁定实用程序的危险在于它将系统暴露给拒绝服务攻击。 最常见的拒绝服务攻击类型是尝试闯入系统管理员的帐户。 经过若干次尝试后，将吊销系统管理员帐户，使系统管理员无法再进行登录。 如果对所有关键用户帐户执行类似的攻击，则会导致系统无法使用且无法恢复。 为了避免发生上述情况，serevu 后台程序提供了下列两种操作模式：

- 将帐户撤消一段指定的时间，在该时间段后帐户自动还原。
- 永久撤消帐户。

serevu 从不吊销 root 用户，因此，从不锁定系统。

注意：有关 serevu 后台程序的详细信息，请参阅*《参考指南》*。

注意：要特别注意 root 用户的密码，以防止对 root 用户的词典攻击成功。

pam_seos

pam_seos 是 CA Access Control 用于高级帐户管理功能的可插入身份验证模块 (PAM)。 CA Access Control 在任何登录程序的登录过程中都调用 pam_seos。 该模块是可以动态加载以根据需要提供所需功能的共享对象。

可以配置 pam_seos 以执行下列三个操作：

- 检测登录失败

  帐户管理组件检测所有失败的登录尝试，并将其记录到审核文件和特殊失败登录文件中。 该模块检测 UNIX 失败，不包括 CA Access Control 拒绝访问的情况。

  CA Access Control 将失败的登录尝试写入特殊文件。 serevu 实用程序读取该文件，并使用其中的信息确定是否应该吊销以及何时吊销用户访问权限。

- 提供调试模式

  当 CA Access Control 拒绝登录时，它通常不在登录会话期间显示拒绝原因。 如果设置了 pam_seos 模块的调试模式，则 CA Access Control 会提供有关拒绝登录原因的简短说明。 例如，“宽限登录”意味着用户已没有剩余登录次数。

- 检查过期的密码和宽限登录

  密码管理组件调用 segrace 实用程序，该实用程序检查用户的密码是否到期和宽限登录的次数。 如果用户的密码到期，且用户已没有剩余的宽限登录次数，则 segrace 调用 sepass 实用程序以允许该用户更改密码。

注意：只有当需要更改密码时，CA Access Control 才调用 segrace。

注意：要从 SSH 获取失败的登录事件，必须将您正在使用的 SSH 版本进行编译和配置以支持 PAM。 如果您的 SSH 版本不使用 PAM，则 CA Access Control 无法检测用户是否违反了失败登录规则。

安装程序将相关行添加到 pam.conf 配置文件中，并将旧的配置文件存储为 /etc/pam.conf.bak。

对 pam_seos 模块的配置是通过 seos.ini 文件执行的。 根据所需的功能，设置 [pam_seos] 部分的下列标记：

要使用“密码过期和宽限登录”检查，请设置 seos.ini 文件中的下列标记：

```
call_segrace = Yes
```

要使用“登录调试模式”，请设置 seos.ini 文件中的下列标记：

```
debug_mode_for_user = Yes
```

要让 serevu 使用 pam_seos 登录失败检测，请设置 seos.ini 文件中的下列标记：

```
serevu_use_pam_seos = Yes
```

约束和限制

本节中所介绍的保护技术具有下列约束和限制：

- 在 Sun Solaris 中，进行五次登录尝试失败后，将通知 serevu。
- 只在支持 PAM 的 Sun Solaris、HP-UX 和 Linux 版本中实施 pam_seos 模块。

检查用户无操作状态

无操作状态功能防止通过其所有者离开的帐号或者组织不再采用的帐号未经授权地访问系统。 非活动日是指用户不登录的日子。 您可以指定在用户帐户被挂起和无法登录之前必须经过的不活动天数。 一旦挂起帐户，则必须手动重新激活它。

注意：在无操作检查方面，会将密码更改视为活动。 如果用户密码更改，则该用户不能因为无操作而被挂起。

可以使用 USER 类记录或 GROUP 类记录的无操作属性设置无操作天数。 后者只影响将该组作为配置文件组的用户。 您还可以使用 SEOS 类的 INACT 属性，为系统范围内的所有用户设置无操作。

在 selang 中，使用以下命令可以通过全局方式指定无操作状态：

```
setoptions inactive (numdays)
```

要为组设置天数（将覆盖该组的系统范围内的无操作设置），请使用以下命令：

```
editgrp groupName inactive (numdays)
```

要为用户设置天数（将覆盖该用户的组设置和系统范围设置），请使用以下命令：

```
editusr userName inactive (numdays)
```

要重新激活挂起的用户帐户，请使用以下命令：

```
editusr userName resume
```

要重新激活挂起的配置文件组，请使用以下命令：

```
editgrp userName resume
```

要在系统范围级别禁用无操作登录检查，请使用以下命令：

```
setoptions inactive-
```

要禁用对组的无操作登录检查，请使用以下命令：

```
editgrp groupName inactive-
```

要禁用对用户的无操作登录检查，请使用以下命令：

```
editusr userName inactive-
```

第 7 章： 管理用户密码

此部分包含以下主题：

密码控制

密码是最常用的身份验证设备，但是密码保护也存在众所周知的问题：

- 简单的密码很容易被猜中。
- 使用了很多年的密码和循环使用的代码最终会被破解。
- 侦听者可以捕捉以明文形式通过网络发送的密码。

定义密码策略

最重要的密码规则是：用户不要直接或间接（通过使用简单密码）泄露自己的密码。 只有通过培训和教育才能实现令人满意的密码安全保护。 虽然 CA Access Control 不能替代教育，但它可以强制执行规则和策略，这些规则和策略可强制用户使用最低质量限度的密码。 您可以指定的规则包括以下内容：

- 新密码不能与以前的密码匹配。
- 新密码不能包含用户名。
- 新密码不能包含所替换的密码。
- 所替换的密码不能包含新密码。
- 新密码不能与所替换的密码匹配（不区分大小写）。
- 新密码必须至少具有最低数量的字母数字字符、特殊字符、数字、小写字符和大写字符。
- 新密码不能有比较多的重复字符。
- 新密码不能是 seos.ini 文件中的 Dictionary 标记指向的词典中的限制性单词之一。

- 每个密码必须有最长使用期限（即，它到期必须失效），从而强制用户在某一时间间隔后选择新的密码。
- 每个密码必须有一个最短使用期限。（通过指定一个最短使用期限，您可以防止用户快速频繁地更改密码。 如果快速更改密码，会溢出密码历史记录列表，然后重新使用先前的密码。）

重要说明！ 密码规则仅影响 sepass，不会影响本地密码工具。 确保使用指向 sepass 的链接替换密码。

配置密码质量检查

配置密码质量检查

1. 在 CA Access Control 端点管理 中，单击“配置”选项卡。

   在左侧将显示配置菜单选项。
2. 在“杂项”区域的选项中单击“类激活”。

   将显示“类激活”页面。
3. 在“用户身份控制”区域中选择“PASSWORD”，然后单击“保存”。

   此操作将激活密码质量检查。
4. 在“策略”区域的选项中单击“用户密码策略”。

   将显示“用户密码策略”页面。
5. 定义用于密码检查的规则，然后单击“保存”。

   您定义的密码检查规则将在密码被更改时立即强制执行。
6. （仅适用于 UNIX）通过使用 sepass 实用程序更新新密码。

   注意：有关 sepass 实用程序的详细信息，请参阅*《参考指南》*。

示例：定义密码检查规则

以下 selang 命令将激活密码质量检查，并定义规定至少使用以下内容的密码规则：

- 六个字母数字字符
- 三个小写字母字符
- 两个数字字符

```
setoptions class+ (PASSWORD)
setoptions password(rules(alpha("6") lowercase("3") numeric("2")))
```

注意：有关 setoptions 命令格式的详细信息，请参阅*《参考指南》*。

更改密码

CA Access Control 包括可执行文件 *ACInstallDir*/bin/sepass（其中 *ACInstallDir* 是 CA Access Control 的安装目录，默认情况下为 /opt/CA/AccessControl/），多数用户使用它（而不使用 /bin/passwd）更改自己的密码。

- 只有 sepass 能确保新密码符合 CA Access Control 密码策略。 也只有 sepass 能使用新密码更新数据库，以及更新密码更改的日期。 此外，sepass 执行的功能与 /bin/passwd 相同。
- 不应使用原始的可执行文件 /bin/passwd，除非您选择*放弃*由 CA Access Control 执行的密码质量检查。 在这种情况下，您可以继续使用该原始的 /bin/passwd，CA Access Control 将接受系统密码而不执行任何密码质量检查。

还可以使用 selang 更改密码。 请输入以下命令为用户指定密码：

```
chusr userName password(string)
```

注意：如果您（以管理员身份）更改另一用户的密码且密码检查已启用，则该用户必须在下次登录时更改密码。

密码过期和宽限登录

时间间隔参数设置可以使用密码的最大天数。 经过指定的天数后，CA Access Control 将通知用户当前密码已到期。 用户可立即更新密码，也可以继续使用旧密码直到达到宽限登录次数。 在后一种情况下，用户无法访问系统，必须与系统管理员联系才能选择新密码。

指定密码时间间隔

在系统范围级别，您可以使用 setoptions 命令指定在系统提示所有用户使用新密码之前的时间间隔。 如果 segrace 实用程序是用户登录脚本的组成部分，或者您将 PAM 配置为调用 segrace（如果您的本地操作系统支持 PAM），则当达到指定天数时，CA Access Control 将通知用户当前密码已到期。 然后，用户可立即更新密码，或继续使用旧密码，直到达到宽限登录次数。 达到宽限登录次数后，将拒绝用户访问系统，并且用户必须与系统管理员联系以选择新密码。

要在系统范围级别设置或取消密码时间间隔，请使用以下命令：

```
setoptions password({interval(NumDays)|interval-})
```

NumDays 的值必须是零或正整数。 如果时间间隔为零，则禁用对用户的密码时间间隔检查。 如果不希望密码过期，请将时间间隔设置为零。 天数为零的时间间隔应仅用于安全需求较低的用户。

interval- 参数可取消密码时间间隔设置。 如果该用户所具有的配置文件组具有该参数的值，则使用该值。 否则，将使用 setoptions 命令设置的默认值。 该参数仅与 chusr 或 editusr 命令一起使用。

设置单个用户或组的密码时间间隔

您还可以为特定用户或配置文件组设置时间间隔。 这些设置将覆盖这些用户或组的系统范围的时间间隔。 当达到指定天数时，CA Access Control 将通知用户当前密码已到期。 然后，用户可立即更新密码，或继续使用旧密码，直到达到宽限登录次数。 达到宽限登录次数后，将拒绝用户访问系统，并且用户必须与系统管理员联系以选择新密码。

要设置或取消用户的密码时间间隔，请输入以下命令：

```
editusr {interval(NumDays) | interval-}
```

要设置或取消组的密码时间间隔，请输入以下命令：

```
editgrp password{(interval(NumDays)) | (interval-)}
```

NumDays 的值必须是零或正整数。 如果时间间隔为零，则禁用密码时间间隔检查。 如果不希望密码到期，请将时间间隔设置为零。 天数为零的时间间隔应仅用于安全需求较低的用户。

interval- 参数可取消密码时间间隔设置。 如果取消了该设置，并且在用户记录中设置了时间间隔值，则将使用该值。 否则，将使用 setoptions 命令设置的默认值。 请仅将该参数与 setoptions、chgrp 或 editgrp 命令一起使用。

宽限登录

启用密码检查后，每次用户尝试登录时 CA Access Control 均将检查该用户的密码是否到期。 密码到期后，用户可以得到“宽限”，有机会再登录若干次，之后，用户不能再登录。

宽限登录选项设置密码到期后、用户挂起前允许的最多登录次数。 宽限登录次数必须介于 0 和 255 之间。 达到宽限登录次数后，将拒绝用户访问系统，用户必须与系统管理员联系以选择新密码。 如果将 grace 设置为零，则用户无法登录。 默认的宽限登录次数为 5 次。

您可以使用这种方法来强制用户更改其密码。 重置用户的密码，并给其一次宽限登录的机会，以便于其更改密码。

跟踪宽限登录

要允许最终用户跟踪到期后的宽限登录，请在该用户的 .login、.profile 或 .cshrc 文件中插入对 segrace 实用程序的调用。 之后，segrace 实用程序将向用户显示一条消息，说明剩余的宽限登录次数。 您还可以使用 segracex 实用程序以图形方式检查用户的密码是否到期。

注意：有关 segrace 和 segracex 实用程序的详细信息，请参阅 *《参考指南》*。

要为宽限登录次数设置系统范围的默认值，请输入以下命令：

```
setoptions password(rules(grace(nLogins)))
```

要设置或取消特定用户的宽限登录，请输入以下命令：

```
chusr userName {grace(nLogins) | grace-}
```

要设置或取消配置文件组的宽限登录，请输入以下命令：

```
chgrp groupName {grace(nLogins) | grace-}
```

chusr 或 chgrp 命令设置的值覆盖该命令中指定的用户的系统值。

注意：GROUP 类的宽限属性（即全局宽限登录设置）设置用户密码到期**后**该用户的宽限登录次数。 然而，USER 类中的宽限属性设置将密码设置为立即到期；宽限登录在用户密码到期后自动设置（使用 GROUP 记录或系统默认设置）。 您不能为组设置密码到期，仅可为用户设置。

第 8 章： 保护文件和程序

此部分包含以下主题：

限制对文件和目录的访问

CA Access Control 不改变 UNIX 的权限系统，但向其添加一个增强的访问权限控制层。

CA Access Control 截获下列各种文件的访问操作，并在将控制返回 UNIX 之前验证用户是否拥有执行特定操作的权限。 访问类型位于括号中。

- 创建文件 (create)
- 打开文件进行读取（读取）

  注意：如果您需要*读取*权限以控制用户是否可以执行获取文件（例如 ls -l）相关信息的操作，请将 STAT_intercept 配置设置设为 1。 有关详细信息，请参阅 *Reference Guide*。
- 打开文件进行写入（写入）
- 执行文件 (execute)
- 删除文件 (delete)
- 重命名文件（删除、重命名）
- 更改权限位 (chmod)
- 更改所有者 (chown)
- 更改时间戳 — 例如，作为执行 touch 命令的结果 (utime)
- 编辑本机 ACL — 使用 acledit 命令 — 对于支持 ACL 的系统 (sec)
- 更改目录 (chdir)

CA Access Control 访问检查与本地 UNIX 授权存在下列不同：

- CA Access Control 基于登录用户的原始用户 ID 而非有效用户 ID (euid) 进行授权检查。 例如，如果 *userA* 调用 su 命令替代其他用户，则 *userA* 仍然只能访问允许 *userA* 访问的那些文件。 与在 UNIX 中不同，代理其他用户的操作*不*会向原始用户自动授予对目标用户文件的访问权限。
- CA Access Control 不向超级用户 (root) 授予自动访问系统上每个文件的权限。 超级用户与系统的所有其他用户一样，必须接受授权检查。
- 授权检查基于 CA Access Control 常规和条件访问权限列表、日期和时间限制、安全级别、安全类别和安全标签。
- 如果未显式授权用户访问某个文件，则 CA Access Control 会检查该用户是否属于已授权访问该文件的任何组。
- 将通过常规 CA Access Control 审核过程审核每次文件访问。
- 删除文件时，CA Access Control 将要求用户拥有对指定文件的删除访问权限，而 UNIX 要求用户拥有对父目录的写入权限。
- 要重命名文件，用户必须拥有对源文件的删除访问权限和对目标文件的重命名访问权限。 UNIX 还要求该用户具有对父目录的写入访问权限。
- 所有用户都被授予对文件 /etc/passwd 和 /etc/group 的永久读取权限（最低权限），而无论这些文件的默认设置为何。 这可避免可能的系统挂起。
- CA Access Control 数据库中 FILE 对象的所有者始终拥有对该对象保护的文件的完全访问权限。
- chdir 访问权限类型专门控制 chdir 命令但不执行该命令，这与 UNIX 相同。

下面是文件保护系统的一些限制：

- 对于不属于特殊 _restricted 组成员的用户，CA Access Control 仅能保护下列文件和目录：
  - 由数据库中的相应文件和目录名称定义的文件和目录
  - 与数据库中定义的名称模式（例如，/etc/*）匹配的文件和目录

  对于属于 _restricted 组的用户，所有系统文件都受 CA Access Control 的保护。 对于未在数据库中定义的文件，授权基于 FILE 类的 _default 记录。

- CA Access Control 维护一个表，该表中的所有文件名和目录名（包括使用通配符的模式）指明了需要保护的资源。 可用于该表的内存量有限。 通常，可通过数据库中的各个名称定义的文件和目录的最大数为 4096，名称模式的最大数为 512。
- 即使不存在用于某些文件的显式访问规则，这些文件仍会受到保护。 这些文件包括 CA Access Control 数据库文件、审核日志以及配置文件。

  注意：有关详细信息，请参阅 *《参考指南》*中的 FILE 类。

CA Access Control 支持下列文件访问类型。

- ALL
- CHDIR
- CHMOD
- CHOWN
- CONTROL
- CREATE
- DELETE
- EXECUTE
- 无
- READ
- RENAME
- SEC
- UPDATE
- UTIME
- WRITE

文件保护系统用于保护包含敏感数据的选定文件集。 例如，您可以使用 CA Access Control 保护下列文件：

- /etc/passwd
- /etc/group
- /etc/hosts
- /etc/shadow

应使用 CA Access Control 保护数据库（应仅为服务器后台程序授予访问权限）和您站点上所有其他敏感文件。

始终需要访问控制的部分文件由规则管理，即使您并没有指定相应规则。

文件保护原理

当 seosd 后台程序启动时，它将对在数据库中定义的每个离散文件对象执行 UNIX stat 命令。 然后，它将在内存中构建一个表，其中包含对应每个文件对象的条目。 另外，对于每个离散文件，该表还包含文件的 inode 和设备；使用该信息，CA Access Control 还可以保护指向这些文件的硬链接，这是因为保护是根据设备和 inode 提供的。 数据库不保留关于文件的 inode 和设备的信息。

通过 CA Access Control 创建新的文件规则时：

- 如果文件存在于 UNIX 中，CA Access Control 首先对文件执行 stat 命令，然后向文件表中添加带有文件的 inode 和设备信息的新条目。
- 如果文件不存在于 UNIX 中，CA Access Control 将向文件表添加文件名称的新条目，该条目不带有 inode 和设备信息。 该条目与常规文件对象的条目相同。 与此同时，内核在其内部表中保留一个指示，指定在创建期间必须检查该文件的 inode 和设备信息。 随后创建该文件时，内核将拦截其创建过程，并通知 seosd 该文件的 inode 和设备信息，以便 seosd 在文件表中更新该文件的条目。

在删除文件时，CA Access Control 将删除 seosd 文件表中该文件的条目，但该条目仍保留在 CA Access Control 数据库中，以备您重新创建该文件。

保护文件

要用 selang 定义受保护文件，请输入以下命令：

```
newres FILE filename
```

例如，要注册一个名为 /tmp/binary.bkup 的文件，请输入以下命令：

```
newres FILE /tmp/binary.bkup
```

注意：如果定义文件规则而未指定其访问类型，则会分配默认的访问权限“无”。 此时，只有文件的所有者能够访问该文件。

应当保护大多数受保护文件，阻止超级用户访问这些文件。 否则，任何知道超级用户密码的用户都能够自动访问该文件。 同时，您还可以阻止除文件所有者以外的所有其他用户访问该文件。

要保护多个具有相似名称的文件，请使用包含通配符的文件名模式。通配符为 *****（表示零个或多个字符）和 **?**（表示除 **/** 之外的任一字符）。

您指定的模式与文件的完整路径名匹配，因此模式 /tmp/x* 可以和名为 /tmp/x1、/tmp/xxx 甚至 /tmp/xdir/a 的文件匹配。

CA Access Control 不允许您指定的模式包括 **/***、**/tmp/*** 和 **/etc/***。

重要说明！ 因为文件名模式是非常强大的工具，所以不应随意地试验使用它们。

例如，以下命令将 /tmp 目录中名称以 a 开头并以 b 结尾的所有文件（将包含类似 /tmp/axyz/axyzb 的文件）定义为受保护文件：

```
newres FILE /tmp/a*b
```

FILE 资源名称中的通配符

通过在 FILE 资源名称中使用通配符，您可以创建与多个文件对应的文件记录：名称与通配符模式匹配的任何文件均受与记录关联的访问权限的保护。

可用通配符如下：

- * 表示任意数量的任意字符
- ? 表示任意单个字符

如果物理资源名称与多个资源记录名称相匹配，则最长的非通配符匹配项将用于该资源。

CA Access Control 不接受以下模式的 FILE 资源名称：

- *
- /*
- /tmp/*
- /etc/*

示例：在 FILE 资源中使用通配符

FILE 资源 /usr/lpp/bin/* 保护 /usr/lpp/bin 下的所有文件和子目录（尽管深度嵌套）。

限制文件访问

要使用 selang 限制超级用户对文件的访问，请使用较长版本的 newres 命令。 例如，要防止超级用户和除用户 myuser 之外的任何其他用户访问文件 /tmp/binary.bkup，您可以使用下列 selang 命令：

```
newres FILE /tmp/binary.bkup owner(myuser) defaccess(N)
```

该命令执行以下步骤：

1. 将 /tmp/binary.bkup 定义为受保护文件。
2. 将 myuser 用户设置为该文件的所有者，授予 myuser 访问该文件的权限。
3. 将文件的默认访问权限设置为无，防止任何其他用户访问该文件。要允许其他用户访问该文件，必须显式定义该文件的访问规则。

重要说明！ 如果您使用 root 权限调用 selang 命令，然后定义 FILE 记录，但没有将另一用户显式指定为其所有者，则 root 将成为这些文件的所有者。 作为所有者，root（或以 root 身份登录的任何用户）均具有对文件的完全、自由的访问权限。

注意：可以将 seos.ini 文件中的标记 use_unix_file_owner 设置为 yes。这可使常规 UNIX 用户为自己拥有的文件定义访问规则。

阻止文件访问

有时，定义没有所有者的 FILE 记录很方便。 要使用 selang 定义没有所有者的 FILE 记录，请使用特殊所有者“nobody”。

例如，要将文件 /tmp/binary.bkup 定义为受保护文件并阻止所有用户访问该文件，请输入以下 selang 命令：

```
newres FILE /tmp/binary.bkup owner(nobody) defaccess(N)
```

该 newres 命令确保即使是定义该命令的用户（无论是否为 root 用户）也不能访问该文件。 阻止所有用户访问文件后，通常您必须显式授权一个或多个用户访问该文件。

要显式允许某个用户访问受保护文件，请使用 authorize 命令。 例如，要授予用户“*userJo*”对 /tmp 目录中以 Jo 开头的所有文件的更新访问权限，请输入以下命令：

```
authorize FILE /tmp/Jo* uid(userJo) acc(Update)
```

注意：CA Access Control 只保护在其数据库中定义的文件。

限制用户获得文件信息

如果您没有为用户提供对文件或目录的*读取*访问权限，则默认情况下，该用户仍然可以使用 stat 功能获得关于文件的信息。 例如，对文件 /tmp/abc 没有*读取*访问权限的用户可以执行下列操作：

```
ls -l /tmp/abc
```

要防止没有*读取*访问权限的用户获得文件信息，请将 STAT_intercept 配置设置设为 1。

注意：有关 STAT_intercept 配置设置的详细信息，请参阅*《参考指南》*。

查看默认访问权限

要查看 _restricted 组（找不到匹配记录时）中用户的默认访问权限，可对该类的 _default 记录使用 selang showres 命令。

例如，要查看 _restricted 组中的用户对 CA Access Control 数据库之外的文件所拥有的默认访问权限，可使用 showres selang 命令显示 FILE 类的 _default 资源：

```
showres FILE _default
```

注意：所有其他用户都拥有特定 CA Access Control 数据库规则所定义的访问权限。

使用条件访问控制列表

您可以指定对文件的访问必须以使用用于访问该文件的特定程序为条件。 这种设置条件的操作称为程序通路。

注意：如果指定的访问文件的程序为 shell 脚本，则 shell 脚本必须以 #!/bin/sh 作为其首行。

下列代码示例允许任何在密码更改程序 /bin/passwd 的控制下更新文件 /etc/passwd 的过程。 对于 /etc/passwd 文件，任何非源自 /bin/passwd 的访问尝试都将被阻止。

```
newres FILE /etc/passwd owner(nobody) defaccess(R)
authorize FILE /etc/passwd gid(users) access(U) via(pgm(/bin/passwd))
```

newres 命令将文件 /etc/passwd 定义到 CA Access Control，并允许任何用户（包括文件的所有者）读取该文件。 如果访问源自程序 /bin/passwd，authorize 命令将允许所有用户访问该文件。 以这种方式保护密码文件时，如果用户未使用 /bin/passwd 程序，则将阻止任何在 /etc/passwd 文件中插入条目的特洛伊木马，并阻止“users”组的用户对密码文件进行任何更新。

条件访问权限列表还用于控制对数据库管理系统 (DBMS) 文件的访问。通常，您应仅允许用户通过数据库提供商提供的程序和实用程序访问此类文件。 请考虑下列命令：

```
authorize FILE /usr/dbms/xyz uid(*) via(pgm(/usr/dbms/bin/pgm1)) access(U)
authorize FILE /usr/dbms/xyz uid(*) via(pgm(/usr/dbms/bin/pgm2)) access(U)
```

通过这组 authorize 命令，如果访问是源自程序 pgm1 或程序 pgm2（属于 DBMS 二进制目录），则所有 CA Access Control 用户均可以访问 DBMS 系统的文件 xyz。 注意星号在用户操作数中的使用。 星号指定定义到 CA Access Control 的所有用户。 从概念上说，使用星号与默认访问权限相似，只是默认访问权限还适用于未定义到 CA Access Control 的用户。 注意，您可以为 CA Access Control 数据库中未定义的用户使用 _undefined 组。

您还可以根据 Unicenter TNG 日历的状态，使用 Unicenter TNG 日历 ACL 属性来允许或拒绝特定用户和组对当前资源的访问。 ACL Unicenter TNG 日历有两类 ACL 属性，分别为常规属性和限制性属性。

例如，以下命令将名为 george 的用户添加到名为 basecalendar 的常规日历的条件访问控制列表中：

```
auth file file1 uid(george) calendar(basecalendar) access(rw)
```

以下命令从 Unicenter TNG 日历中删除名为 george 的用户：

```
auth- file file2 uid(george) calendar(basecalendar)
```

使用否定式访问控制列表

您可以使用否定式访问控制列表 (NACL) 拒绝特定于用户或组的访问类型。

通过 CA Access Control 语言 (selang)，使用以下命令拒绝访问：

```
auth className resourceName [gid(group-name...)] \
[uid({user-name...|*})] [deniedaccess(accessvalue)]
```

用 _abspath 组阻止特洛伊木马

$PATH 变量中的任何相对路径名都是安全漏洞，特别是表示“当前目录”的圆点 (.) 路径名。 请考虑以下情况：

- 代表根目录的 PATH 变量的顶端是当前 (.) 目录。

- 恶意用户创建了一个破坏性程序，即特洛伊木马，并将其存储为 /tmp/ls。
- 正如恶意用户所希望的，root 会在 /tmp 目录中及时发出 ls 命令。实际上，root 并没有运行常规的 ls 命令，而是使用完全管理权限运行已存储在 /tmp 目录中的特洛伊木马。

要消除该安全漏洞，CA Access Control 提供了名为 _abspath 的用户组。禁止 _abspath 组的所有成员在调用程序时使用相对路径名。

和向任何其他组中添加用户一样，您可以将用户添加到 _abspath 组。该用户在下次登录时生效，禁止该用户在访问程序时使用相对路径名。

与本地 UNIX 安全同步

虽然 CA Access Control 权限比本地 UNIX 权限更复杂，但可以将本地 UNIX 权限与您的 CA Access Control 权限同步。 也就是说，可以使这两种权限一致。 不过，同步受到以下限制：

- 同步不可逆。 一旦同步生效，它即可管理所有新发出的 CA Access Control 授权命令，但它不管理之前存在的访问规则。
- 在 CA Access Control 中授予的权限可以传递到 UNIX，但在 UNIX 中授予的权限不能传递到 CA Access Control。
- 由于自身权限系统的局限性，UNIX 可能无法采用多个简化形式的 CA Access Control 权限。 即使具有 Access Control 列表 (ACL) 功能的 UNIX 版本也无法反映 CA Access Control ACL 的所有复杂性。

可与 CA Access Control 同步且具有 ACL 的 UNIX 平台包括 Sun Solaris、HP-UX 和 Tru64。

在没有此类 ACL 的情况下，仍然可以将传统的 UNIX rwx 权限与 CA Access Control 权限在最大程度上同步。

同步由 authorize 命令的 UNIX 选项与 seos.ini 文件的 SyncUnixFilePerms 标记的组合控制：

- 通过包含 UNIX 选项，authorize 命令要求在 UNIX 和 CA Access Control 中实施。 该命令甚至可以在以前不存在权限的位置授予 UNIX 权限。

  （如果*未*使用 UNIX 选项，则 selang 命令对 UNIX 安全不起作用。 此外，在 UNIX 保留禁令的位置，CA Access Control 权限也无效。 因此，selang 可以克服 UNIX 禁令的唯一方法是在 authorize 命令中使用 UNIX 选项。）

- 在 authorize 命令中，必须在 seos.ini 文件的 [seos] 部分中正确设置 SyncUnixFilePerms 标记，UNIX 选项才会起作用。 该标记有几个允许使用的值：

  - **no** - 指定不同步 ACL 权限。 这是默认值。
  - **warn** - 指定不同步 ACL 权限，但在 CA Access Control 与本地 UNIX 权限冲突时发出警告。
  - **traditional** - 指定根据 CA Access Control ACL 调整组的 rwx 权限（并且不将单独用户的权限复制到 UNIX）。
  - **acl** - 指定根据 CA Access Control ACL 调整 UNIX ACL。
  - **force** - 指定根据 CA Access Control defaccess 权限调整 UNIX 全局访问属性。

  只有在重新启动 seosd 后台程序以后，SyncUnixFilePerms 标记值中的所有更改才会生效。

示例：同步

以下示例涉及名为 /var/temp/newdata 的文件和名为 fowler 的用户，并假设 FILE 类中的一个记录已代表该文件。

1. 关闭 seosd 后台程序，以便能够编辑 seos.ini 文件：

   ```
   # secons -s
   ```

2. 以拥有编辑 seos.ini 文件的权限的用户身份登录，编辑 seos.ini 文件，使 [seos] 部分中的 SyncUnixFilePerms 行如下所示：

   ```
   SyncUnixFilePerms = acl
   ```

   请记住，acl 表示 UNIX 选项根据 CA Access Control ACL 调整 UNIX ACL。 只要该标记保持设置为 acl，UNIX 选项就具有该功能。

3. 重新启动 seosd 后台程序：

   ```
   # seosd
   ```

4. 调用 selang，然后发出以下 selang 命令：

   ```
   authorize FILE /var/tmp/newdata uid(fowler) access(r w) unix
   ```

   该命令授予 fowler 对新数据文件的“Read”和“Write”访问权限，并通过指定 UNIX 选项授予相应的本地 UNIX 权限。

HP-UX 限制

限制 HP-UX 的 ACL 反映 CA Access Control 的 ACL 的方式。

- 在 HP-UX 中，ACL 按用户和组的*组合*分配访问权限。 也就是说，仅在指定用户的主组也被指定时，分配的访问权限才会应用于该用户。

  与此不同，CA Access Control 按用户或者按组（而不是按组合）分配访问权限。

  相应地，CA Access Control 权限映射到其中用户或组的设置与设置为“*”或“any”等效的 HP-UX 用户/组组合。

- HP-UX 不支持在卷管理器 (LVM) 的控制之下的文件系统上的 ACL。因此，某些重要的 HP-UX 计算机可能只允许在“root”文件系统上进行 ACL 同步。

- HP-UX 的 ACL 限制为 16 个条目。 CA Access Control 同步以尽可能高效的方式使用可用条目，但 16 个条目可能不足以完全反映每个 CA Access Control ACL。

Sun Solaris 限制

在 Sun Solaris 下，本地 UNIX ACL 不在 /tmp 目录中实现。

监视敏感文件

Watchdog 可以保护 setuid/setgid 程序的二进制文件，以及您指定的任何其他文件。 seoswd 实用程序（Watchdog 后台程序）会连续检查两个问题：

- seosd 后台程序是否处于活动状态以及是否响应。 （如果需要，Watchdog 后台程序将重新启动 seosd 后台程序。）
- 用户是否修改了任何受托程序或文件。 （如果已修改，seoswd 将阻止执行这些文件。）

seosd 后台程序派生时，它自动执行 seoswd 程序以启动监视程序。

注意：有关 seoswd 的详细信息，请参阅*《参考指南》*。

seos.ini 文件包含控制 watchdog 的扫描和超时值的若干个标记。 它还包含有关这些值的最新文档。

注意：有关 seos.ini 文件的说明，请参阅*《参考指南》*。

您可以使用监视程序对普通文件执行对 setuid 和 setgid 程序所执行的相同后台检查，包括在这些文件改变时生成审核报告。

例如，请考虑如下配置：只允许安全管理员修改文件 /etc/inittab。 要使 CA Access Control 监控该文件并在有任何修改时生成警报，可使用以下 selang 命令：

```
newres SECFILE /etc/inittab
```

现在，即可始终监视对文件 /etc/inittab 的修改。

内部文件保护

在安装期间，CA Access Control 制定规则来保护两种类型的内部文件：

- 内部规则 — 保护配置文件、日志文件和数据库文件。

  您无法删除内部规则。
- 默认规则 — 保护敏感文件，例如您用来加密和验证通信的根证书和服务器证书。

  您可以在安装之后删除默认规则。

内部文件规则

内部文件规则保护配置文件、日志文件和数据库文件。 内部文件规则在 selang 中不可见且无法删除。

CA Access Control 使用内部文件规则保护的文件具有以下访问权限：

- 对 CA Access Control 内部进程的完全访问
- 对所有其他访问者的读取和执行（有相关需要时）访问

您可以编写 FILE 规则来替换内部文件规则。 如果删除这些 FILE 规则，CA Access Control 会恢复到内部文件规则。

CA Access Control 使用内部文件规则保护下列文件。 该表的第二列列出了指定文件位置的配置设置（如果适用）。

注意：一些文件位置是内部定义的，没有相应的配置设置。 您不能配置这些文件的位置。

文件	seos.ini 中的配置设置和部分	默认文件位置
所有数据库文件	[seosd] dbdir	*ACInstallDir*/seosdb
seos.ini	-	*ACInstallDir*
privpgms.ini	-	*ACInstallDir*/etc
loginpgms.ini	-	*ACInstallDir*/etc
xdmpgms.init	-	*ACInstallDir*/etc
nfsdevs.init	[seosd] nfs_devices	*ACInstallDir*/etc
osver	-	*ACInstallDir*/etc
accommon.ini	-	*ACSharedDir*
seos.audit	[logmgr] audit_log	*ACInstallDir*/log
seos.audit.bak*	[logmgr] audit_back	*ACInstallDir*/log
seos.error	[logmgr] error_log	*ACInstallDir*/log
kbl.audit	[kblaudit] audit_log	*ACInstallDir*/log
kbl.audit.bak	[kblaudit] audit_back	*ACInstallDir*/log
kbl.error	[kblaudit] error_log	*ACInstallDir*/log

注意：有关配置设置的详细信息，请参阅*《参考指南》*。

默认文件规则

CA Access Control 在安装期间会创建默认文件规则来保护敏感文件。 默认文件规则在 selang 中可见且能够被删除。

下表列出 CA Access Control 使用默认文件规则保护的敏感文件以及这些文件的访问权限和允许的访问者。

在该表中，*PMDBDir* 是策略模型数据库 (PMDB) 所在的目录，而 *pmd_name* 是每个策略模型的名称。 默认情况下，*PMDBDir* 位于 *ACInstallDir*/policies。 *PMDBDir* 的位置在 seos.ini 文件的 pmd 部分中的 _pmd_directory_ 标记中定义。

文件	默认访问权限	允许的访问者
ACInstallDir/data/crypto/crypto.dat	无	sechkey
ACInstallDir/data/crypto/def_root.pem*	无	sechkey
ACInstallDir/data/crypto/sub.key	无	sechkey
ACInstallDir/data/crypto/sub.pem	无	sechkey
ACInstallDir/log/policyfetcher.log	读取	+policyfetcher
ACInstallDir/ladb/*db.la*	读取	sebuildla
/etc/passwd	全部	全部
/etc/shadow	全部	全部
PMDBDir/pmd_name/hsock	读取、写入、执行、创建、Chown、Chmod、Utime	seagent、sepmdd
PMDBDir/pmd_name/pmd.ini	读取	seagent、sepmdd
PMDBDir/pmd_name/seos_*	读取、写入、执行、创建、Chown、Chmod、Utime	seagent、sepmdd
PMDBDir/pmd_name/socket	读取、写入、执行、创建、Chown、Chmod、Utime	seagent、sepmdd

保护 setuid 和 setgid 程序

设置用户 ID (setuid) 程序是 UNIX 站点上最常用的程序之一。 调用 setuid 程序的进程会自动获得 setuid 程序所有者的身份。 如果 setuid 程序所有者是 root，则任何常规用户通过调用 setuid 程序都会自动成为超级用户。 当 setuid 程序启动时，该进程可执行超级用户可以执行的任何操作，因此，确保 setuid 程序准确执行应完成的操作而不执行任何其他操作非常重要。 setuid 程序中的后门或 shell 授予用户访问系统中任何内容的权限。

CA Access Control 使用 PROGRAM 类保护 setuid 和 setgid 程序。 完成安装之后，默认情况下 CA Access Control 允许执行任何程序。 在数据库中定义受托程序后，您可以更改 CA Access Control 的行为，从而禁止执行 setuid 或 setgid 程序，除非将该程序定义为受托程序。 例如，要允许 /bin/ps（进程状态程序）作为 setgid 程序运行（应完成的操作），可使用以下 selang 命令：

```
newres PROGRAM /bin/ps defaccess(EXEC)
```

CA Access Control 将 /bin/ps 程序注册为受托程序。 然后，CA Access Control 计算其 CRC、inode 号、大小、设备号、所有者、组、权限位、上次修改时间以及其他数字签名（此项可选），并将这些信息存储在数据库的 PROGRAM 类的记录中。

Watchdog 会定期检查该程序的 CRC、大小、inode 和其他特性。 如果其中的任何值已更改，Watchdog 会自动要求 seosd 从受托程序列表中删除该程序，并且拒绝对它的访问。 这确保不会有人通过修改或移动 setuid 程序而滥用该程序。 注意，newres 命令示例中的权限允许所有用户（包括未在数据库中定义的用户）运行 /bin/ps 命令。

未受托的 setuid 程序可能是基于 UNIX 的操作系统的最危险的安全漏洞。 通过使用受托程序的访问规则，安全管理员可以将 setuid 限制为仅供已经过确保完整性的测试和检查的特定受托程序使用。 但是，任何用户都不能自动启动受托的可执行程序；访问规则必须指定已授权访问该 setuid 程序的明确用户和组。 例如，下面一组 selang 命令仅授权 System Department 用户（sysdept 组）执行 /bin/su：

```
newres PROGRAM /bin/su defaccess(NONE)
authorize PROGRAM /bin/su gid(sysdept) access (EXEC)
```

使用星号 (*) 可以指定在数据库中定义的所有用户。 例如，要允许定义到 CA Access Control 的所有用户可执行 su 命令，请输入以下命令：

```
authorize PROGRAM /bin/su uid(*) access(EXEC)
```

该说明也适用于 setgid 可执行文件。

您可以使用 nr 和 er 命令在 PROGRAM 类中注册 setuid 和 setgid 程序。可以使用类似方法在 PROGRAM 类中注册一些重要的非 setuid 和 setgid 程序。为这些程序定义一个 FILE 规则，阻止未经授权的用户对它们进行升级。如果要允许执行取消受托的程序（在升级后，将执行该程序而不重新托管），请将 blockrun 属性设置为 no。

- 如果 blockrun 属性设置为 yes，则只有该程序重新成为受托程序之后才能执行，并且不允许该程序访问相关 PACL 允许的任何文件。在该程序重新成为受托程序前，PACL 实际上被禁用。
- 如果 blockrun 属性被设置为 no，则会执行该程序。但是，该程序无法访问相关 PACL 可能允许的任何资源。

要将 blockrun 属性的值设置为 yes，可使用以下 editres/newres 命令：

```
er program /bin/p blockrun
```

要将 blockrun 属性的值设置为 no，可使用以下 editres/newres 命令：

```
er program /bin/p blockrun-
```

在默认情况下，对于在 PROGRAM 类中注册的所有程序，blockrun 属性均设置为 yes。您可以使用 seos.ini 文件中的 SetBlockRun 标记更改该属性。有关详细信息，请参阅 seos.ini 文件说明。

自动定义 setuid/setgid 程序

CA Access Control 提供了自动定义所有 setuid 和 setgid 程序的方法。使用实用程序 /bin/seuidpgm，可以构建一组定义所有 setuid 程序及其权限的命令。

例如，要扫描整个文件系统以查找 setuid 和 setgid 程序，并将生成的 selang 命令写入 /tmp/pgm_script 文件，可输入以下 selang 命令：

```
# seuidpgm -qln / -x /home > /tmp/pgm_script
```

您可以在提交之前，根据需要编辑和修改 seuidpgm 生成的输出文件。

注意：有关 seuidpgm 实用程序的详细信息，请参阅*《参考指南》*。要了解如何对 setuid 和 setgid 以外的程序提供相似保护，请参阅*《参考指南》*中的 SECFILE 类。

条件访问

另一种复杂的权限技术是条件访问权限规则。 例如，假设您有一个称为 securedSU 的非常安全的 su 命令版本，在允许用户成为超级用户之前，该版本使用指纹识别器来验证用户的身份。

按如下所示设置条件访问权限规则是确保 *UserX* 仅成为该程序下的超级用户的一种方法（在设置该规则之前，还必须为 USER.root 设置 defaccess(none)。）：

```
authorize SURROGATE USER.root uid(UserX) via(pgm(securedSU))
```

保护登录命令

强烈建议您将 /bin/login 限制为仅供超级用户使用。 否则，知道另一用户的密码的任何用户都能够以该用户的身份登录，并提供该用户的密码以绕过所有替代限制和终端限制。

要使用 selang 更改 /bin/login 权限，请使用以下命令：

```
chres LOGINAPPL /bin/login defaccess(N) owner(root)
```

保护常规程序

CA Access Control 还可以用保护 setuid 和 setgid 程序的相同方法来保护常规程序。 为此，可将 PROGRAM 类中的 blockrun 属性设置为您选择的值。

注意：有关可能选项的详细信息，请参阅*《参考指南》*。

内核模块加载和下载保护

*内核模块*是 UNIX 操作系统的组件，您可以加载此组件以扩展运行的内核，并可在不再需要时进行卸载。 这提高了灵活性，您可以根据需要加载功能，而不会浪费内存资源，这些内存资源将可用于基本内核中所有可能的预期功能。

您可以在 CA Access Control 中禁用和启用内核模块保护。 如果您启用内核模块保护，则 CA Access Control 会拦截加载和卸载内核模块的系统调用，之后根据数据库中的关联记录（KMODULE 类的记录）检查请求的访问权限。 如果请求对内核模块记录 CA Access Control 的访问权限，则请求的访问权限为“加载”或“下载”。

在所有的非 Linux 系统上，KMODULE 记录的名称必须匹配内核模块文件的名称（而不是完整路径）。 这是因为模块的名称与文件的名称相同。在 Linux 上，KMODULE 记录的名称仅需要匹配内核模块的名称，后者可能不同于实际的文件名。 如果在 Linux 上更改文件名，不会更改 Linux 使用的模块的名称，且 KMODULE 记录仍然有效。

如果您在加载内核模块时启用文件路径检查，并加载请求的访问权限，则 CA Access Control 将执行以下其他检查：

- KMODULE 记录中的 filepath 属性仅保留有效的绝对文件路径。
- 使用路径名 filepath 的文件具有与 KMODULE 记录名称匹配的模块。
- 内核模块与 KMODULE 属性（非 Linux 系统的文件路径、Linux 系统的签名）匹配。

  注意：CA Access Control 为 Linux 系统上的内核模块文件生成一个唯一的签名，并将该签名作为签名属性值插入内核模块记录中。 CA Access Control 在每次访问时检查签名。 您无需自己输入签名，因为 CA Access Control 会自动计算和插入该签名。 不过，您可以使用 seretrust 实用程序自己输入签名。

更多信息：

加载内核模块时启用和禁用文件路径检查 (p. 98)

保护内核模块

您可以保护内核模块的加载和卸载，这样有助于保护操作系统。

保护内核模块

1. 确保您已启用内核模块保护。
2. 在 CA Access Control 中创建 KMODULE 记录。

   要创建内核模块，您需要定义：

   - 内核模块的名称

     在所有的非 Linux 系统上，KMODULE 记录的名称必须匹配内核模块文件的名称（而不是完整路径）。 这是因为模块的名称与文件的名称相同。 在 Linux 上，KMODULE 记录的名称仅需要匹配内核模块的名称，后者可能不同于实际的文件名。
   - 记录的所有者（默认为创建模块的用户）
   - （可选）内核模块文件的绝对文件路径，或文件路径列表（如果有多个模块版本）。

   注意：在 HP 系统和 Solaris 系统上，您可以定义专门的内核模块 _ALL_MODULES 来保护所有内核模块的卸载。
3. 定义有权限加载和卸载模块的用户或组。

示例：使用 selang 命令保护内核模块

下列 selang 命令将内核模块 serial.o 定义并授权到 CA Access Control，并授予企业用户 kadmin 加载和下载该内核模块的权限：

```
newres kmodule serial.o owner(kadmin) defaccess(none) \
filepath(/lib/modules/2.2.19/serial.o:/lib/modules/2.2.20/serial.o)
authorize kmodule serial.o access(load, unload) xuid(kadmin)
```

启用和禁用内核模块保护

如果启用了内核模块保护，则 CA Access Control 将检查在 CA Access Control 数据库中定义的内核模块的加载和卸载。

默认情况下，CA Access Control 启用内核模块保护。

要启用或禁用内核模块保护，请启用或禁用 KMODULE 类，例如通过使用 setoptions 命令。

示例：使用 selang 启用内核模块保护

下列 selang 命令启用内核模块保护：

```
setoptions class+(kmodule)
```

示例：使用 selang 禁用内核模块保护

下列 selang 命令禁用内核模块保护：

```
setoptions class-(KMODULE)
```

加载内核模块时启用和禁用文件路径检查

如果启用了内核模块保护，则还可以在加载内核模块时启用文件路径检查。 如果启用了文件路径检查，CA Access Control 将检查要加载的内核模块是否与 KMODULE 记录的 filepath 属性匹配（针对非 Linux 系统），或是否与 KMODULE 记录的签名匹配（针对 Linux 系统）。

要启用文件路径检查，请在配置文件 seos.in 的 seosd 部分中，将 special_check 标记设置为 yes（默认设置为 no）。

只有在同时启用文件路径检查和内核模块保护时，CA Access Control 才会检查文件路径。

示例：使用 seini 实用程序对内核模块加载启用文件路径检查

要对内核模块加载启用文件路径检查，您可以使用 seini 和 secons 实用程序，如下所示：

```
seini -s seosd.special_check yes
secons -rl
```

保护二进制文件不被 kill 命令终止

您必需保护关键任务进程（例如数据库服务器或应用程序后台进程）免受拒绝服务攻击。 本地 UNIX 安全系统基于进程用户 ID 执行进程保护。这表明在本地 UNIX 下，root 可以对任何进程执行任何操作。 CA Access Control 根据进程中运行的可执行文件定义规则，从而提高 UNIX 进程保护。 CA Access Control 进程保护*独立子*进程的用户 ID。 PROCESS 类中的记录必须定义 CA Access Control 将保护的每个进程。

例如，要保护 ASCII 查看器 /bin/more 不会被终止，请执行以下过程：

1. 启动 selang。
2. 输入以下 selang 命令：

   ```
   newres PROCESS /bin/more defaccess(N) owner(nobody)
   ```

   该命令将 /bin/more 定义为要针对终止尝试提供保护的进程；因此默认访问权限为*无*(N)。 **owner(nobody)** 设置确保即使是定义该规则的用户也不能终止 /bin/more 进程。
3. 退出 selang。
4. 测试第 2 步定义的规则：
   a. 输入以下命令：

      ```
      /bin/more /tmp/seosd.trace
      ```

   b. 假设 /tmp/seosd.trace 文件足够大，可防止 /bin/more 立即退出，那么按 Ctrl+Z 可挂起 /bin/more 进程。
   c. 尝试通过输入以下命令终止已挂起的作业：

      ```
      kill %1
      ```

      您的尝试应失败，CA Access Control 将显示“权限被拒绝”消息。

要指定允许特定用户终止 /bin/more 进程的例外，请输入以下 selang 命令：

```
authorize PROCESS /bin/more uid(username)
```

注意：使用相同过程可保护系统中的其他二进制可执行文件不会被终止。

CA Access Control 保护常规终止信号 (SIGTERM) 以及应用程序无法屏蔽的终止信号（SIGKILL 和 SIGSTOP）。 它将其他信号（例如 SIGHUP 或 SIGUSR1）传递到确定是忽略还是响应终止信号的进程。

第 9 章： 控制登录命令

此部分包含以下主题：

控制登录进程

CA Access Control 提供两类登录保护：通过终端和通过应用程序。 使用 TERMINAL 类，可以确定哪些用户可以从哪些终端或主机登录。

注意：有关 TERMINAL 类的详细信息，请参阅*《参考指南》*。

使用特定登录应用程序（如 Telnet、FTP 和 RLOGIN）和 LOGINAPPL 类，您还可以控制哪个用户或组可以登录。 通过建立该类的访问规则，可为每个登录应用程序定义特定规则。 例如，可以定义允许所有用户通过 ftp 登录到主机、允许一定数量的用户通过 telnet 登录到系统和不允许任何用户通过 rlogin 登录到系统的规则。 LOGINAPPL 类中的每个记录为一个特定登录应用程序定义访问规则。

示例：LOGINAPPL

例如，要只允许匿名用户使用 FTP 应用程序，请执行以下过程：

1. 使用以下 selang 命令将 ftp 默认访问权限更改为无：

   ```
   cr LOGINAPPL FTP defaccess(NONE) owner(nobody)
   ```

2. 使用以下 selang 命令允许匿名用户使用 ftp 进行登录：

   ```
   auth LOGINAPPL FTP uid(anonymous) access(X)
   ```

限制名为 Account 的组中的用户只能使用 Telnet：

1. 使用以下 selang 命令禁止使用 rlogin 和 rsh：

   ```
   auth LOGINAPPL(RLOGIN RSH) gid(account) access(N)
   ```

2. 使用以下 selang 命令允许名为 account 的组使用 telnet：

   ```
   auth LOGINAPPL TELNET gid(account) acc(X)
   ```

注意：上一示例显示了 RLOGIN 和 RSH 限制，但是也应该包括其他登录程序。

无论何时添加或使用新的登录程序，均必须添加新的 LOGINAPPL 记录。

登录截获序列始终从称为*触发器*的 setgid 或 setgroups 事件开始。 该序列以 setuid 事件结束，该事件用于将用户的身份更改为实际登录的用户。

登录应用程序发出各种系统调用，CA Access Control 使用这些系统调用来监控登录活动。 这些登录序列为标准登录应用程序而预设。 您可以通过查看 CA Access Control 跟踪文件来查看这些序列。

注意：有关 LOGINAPPL 类和设置序列的详细信息，请参阅*《selang 参考指南》*。

启用 SFTP 登录截获

当用户使用 SFTP 登录到端点时，SFTP 应用程序使用 SSH 来验证该用户。 当 CA Access Control 拦截来自 SFTP 应用程序的登录尝试时，默认情况下，会将登录视为 SSH 登录并使用 SSH LOGINAPPL 记录的规则允许或拒绝登录尝试。

要配置 CA Access Control 来辨别 SFTP 登录和 SSH 登录并为 SFTP 和 SSH 登录编写各自的规则，您必须启用 SFTP 登录截获。

启用 SFTP 登录截获

1. 在端点上打开命令提示符窗口。
2. 输入以下 selang 命令：

   ```
   er LOGINAPPL SSH loginflags(EXECLOGIN)
   ```

   该命令指定 SSH 登录的触发器是进程执行的第一个 EXEC 操作。
3. 输入以下 selang 命令：

   ```
   er LOGINAPPL SFTP loginpath(path) defaccess(a)
   ```

   loginpath(*path*)

   指定 SFTP 登录应用程序的完整路径。

   该命令创建名为 "SFTP" 的 LOGINAPPL 记录，定义了 SFTP 登录应用程序的路径，并指定如果没有其他的限制存在，所有用户都可以使用 SFTP 登录到端点。

示例：启用 SFTP 登录截获

该示例为位于 /usr/libexec/openssh/sftp-server 的 SFTP 登录应用程序启用 SFTP 登录截获。 第一个 selang 命令还指定了 CA Access Control 将 PAM 登录截获用于 SSH 登录：

```
er LOGINAPPL SSH loginflags(EXECLOGIN, PAMLOGIN)
er LOGINAPPL SFTP loginpath(/usr/libexec/openssh/sftp-server) defaccess(a)
```

注意：有关 LOGINAPPL 类的详细信息，请参阅 *《selang 参考指南》*。

控制一般登录应用程序

CA Access Control 还可以控制和保护通用登录应用程序；这意味着您可以用通用模式保护与某一规则匹配的登录应用程序组。 要定义一般登录应用程序，请使用 LOGINAPPL 类。

定义一般登录应用程序

要用 selang 定义一般登录应用程序，请使用与设置常规登录限制一样的命令，但不使用 LOGINPATH 参数（该参数应包括一个一般路径，由一般表达式组成，使用下列一个或多个字符：[、]、*、?）。 例如，要定义一个一般 telnet 应用程序，请使用下列命令：

```
er LOGINAPPL GENERIC_TELNET loginpath(/usr/sbin/in.tel*)
```

一般登录程序的截获

如果使用常规登录限制，激活的规则是显而易见的；如果数据库中存在 LOGINAPPL 对象，该对象具有为 loginpath 属性指定的被截获登录程序，则应用该对象的规则。

但是，对于通用 LOGINAPPL 对象，CA Access Control 执行下列操作：

1. seosd 搜索被截获的登录应用程序中是否有准确匹配 （LOGINAPPL 对象的匹配登录路径。）。如果找到，则应用该对象的规则。
2. 如果未找到，则使用匹配的一般登录路径继续搜索 LOGINAPPL 对象。
3. 如果有多个匹配，则应用具有更多特定匹配的对象的规则。

定义用户使用终端的权限

防止入侵者访问系统的最有效方法之一是采取终端（即登录源）保护。该源可以是用户从其中进行登录的主机或终端（如 X 终端或控制台）。

在如今的现代化体系结构中，终端不再是为其开发 UNIX 的 teletype 计算机。 在大多数站点中，通过 pseudo 终端服务器 (PTS) 或 X 窗口管理器分配“pseudo 终端”，对于安全系统而言，该终端的名称是没有意义的符号。 CA Access Control 保护我们视为终端的对象。 当 CA Access Control 以下列三种方法之一定义终端时，CA Access Control 在登录阶段中执行终端保护：

- 当用户使用 XDM 登录窗口从 X 终端登录时，CA Access Control 将通过 /etc/hosts、NIS 或 DNS 转换为主机名的 X 终端的 IP 地址用作用于登录请求的终端。 如果主机名转换失败或者您更喜欢使用 IP 地址，CA Access Control 还可以使用 IP 地址进行保护。
- 当用户从哑终端登录时，TTY 名称标识该终端。
- 当用户（通过 Telnet、RLOGIN、RSH 等）从网络登录时，通过 /etc/hosts、NIS 或 DNS 转换为主机名的请求 IP 地址被当作终端名。

通过在 TERMINAL 类中定义特定主机以及向对象的访问列表添加适当的用户和组，可以为该特定主机定义登录规则。 对于每个登录源，还可以通过设置 TERMINAL 对象的日期和时间限制来限制允许从该主机或终端登录的日期和时间。 您还可以在 TERMINAL 类中使用通配符来定义与模式（主机名或 IP 地址）匹配的主机。

在大多数情况下，必须限制拥有强大权限的用户（如超级用户或系统管理员）从位于安全位置的终端登录。 希望以超级用户身份进入系统的入侵者和黑客无法从他们自己的远程工作站执行该操作；他们必须从一个位于受保护位置的授权终端执行操作。

通过网络登录时，您不能确定用户确实坐在主机控制台前。 用户可能正坐在与该主机连接的任何终端前，或正在与网络中有权从请求主机接收服务的任何其他节点进行通讯。 允许用户从另一主机登录意味着不仅允许该用户从特定工作站登录，还允许该用户从该工作站授权的任何其他终端登录。 为了确保部门之间的隔离，需要对终端组进行定义，并只允许每个部门的用户从他们部门的终端组进行操作。

与其他资源不同，在终端授权中，授予用户访问信息的权限越大，用户的终端权限就会越低。 超级用户必须是终端访问中最受限制的用户，从而确保没有人可以以 root 身份从不安全的远程终端登录。

当定义终端时，CA Access Control 将要求您显式指定终端定义的所有者。 这是因为，如果 root 作为安全管理员成为终端的所有者（默认设置），则超级用户可登录该终端。 在大多数情况下，并不希望这样。 为了防止您犯下此类可能会不小心导致漏洞的错误，CA Access Control 会要求您在定义终端时定义所有者。

要定义终端 tty34，请使用以下命令：

```
newres TERMINAL tty34 defaccess(none) owner(userA)
```

该命令为终端 tty34 创建记录，将其默认访问权限设置为无，并将 userA 定义为其所有者。 请注意，将自动允许作为终端所有者的 userA 通过终端 tty34 进入系统。

要防止所有用户从终端 tty34 登录，请指定“nobody”作为所有者：

```
newres TERMINAL tty34 defaccess(none) owner(nobody)
```

要允许用户从特定终端登录，请输入以下命令：

```
authorize TERMINAL tty34 uid(USR1)
```

该命令允许 USR1 从终端 tty34 登录。

还可以授予组使用终端的权限。 例如，以下命令允许组 DEPT1 的成员使用终端 tty34：

```
authorize TERMINAL tty34 gid(DEPT1)
```

要定义一组终端（称为终端组），请输入以下命令：

```
newres GTERMINAL TERM.DEPT1 owner(ADM1)
```

要将成员终端添加到终端组 TERM.DEPT1，请输入以下命令：

```
chres GTERMINAL TERM.DEPT1 mem(tty34, tty35)
```

要授权 USR1 使用该终端组，请输入以下命令：

```
authorize GTERMINAL TERM.DEPT1 uid(USR1)
```

这可以向 USR1 授予使用 tty34 和 tty35 的权限。

限制 Root 用户的终端

要考虑的另一个问题是 TERMINAL 类的默认规则。 在初始实施阶段，默认规则设置为允许未定义的任何对象。 在 TERMINAL 情况中，这可能会成为缺点。

考虑以下情况：某站点有几百个终端，您希望大多数用户能够从任何终端登录，但是希望 root 只能从两个预定义的终端登录。

首先，假定将 TERMINAL 类的默认值设置为 READ 可以使任何用户（包括 root）从在数据库中没有特定 TERMINAL 记录的任何终端登录。 您不希望超级用户能够从任何终端登录。 此外，还假定将 TERMINAL 类的默认值设置为 NONE 会强制您定义数据库中的每个终端（这可能并不实际）。

为了解决该问题，CA Access Control 支持 TERMINAL 类的 _default 记录中 Access Control 列表的定义。 以下命令显示如何花费最少的工作将 root 限制到两个终端：

```
newres TERMINAL term1 defaccess(N) owner(root)
newres TERMINAL term2 defaccess(N) owner(root)
newres TERMINAL _default defaccess(R)
authorize TERMINAL _default uid(root) access(N)
```

前两个命令将 term1 和 term2 定义为 root 拥有的终端，以便超级用户可以登录这两个终端。 newres TERMINAL _default 和 chres 命令将默认访问权限设置为读取，以便任何人都可以访问数据库中未定义的任何终端。 authorize 命令显式拒绝超级用户对未定义终端的访问权限。

注意：UACC 类仍然存在；可以使用它指定资源的默认访问权限。 不过，使用 _default 记录指定资源的默认访问权限要简单得多。

建议使用的限制

如果 TERMINAL 类的默认访问权限为读取，则您应该限制使用回环终端、本地主机终端和工作站主机的名称。 如果允许用户使用这些终端，若其他用户知道目标用户的密码，则将允许所有这些用户替换他们自己的用户 ID。 例如，假设以下情况：

- 允许用户 U 从终端 T 操作。
- 不允许终端 T 用于超级用户登录。
- 不向用户 U 授予将用户 ID 替换成 root 的权限。
- 用户 U 设法获取超级用户的密码。
- 允许所有用户从终端回环登录。

用户 U 只需执行命令 telnet 回环、指定用户 ID root 和提供密码，即可跳过该组访问规则。 现在，超级用户会话已从终端 T 启动，该终端被假设为不允许超级用户登录。 用户可以利用本地主机或工作站主机的名称，从而以相似方式跳过访问规则。

要限制这三个漏洞，请使用以下定义：

```
newres TERMINAL loopback defaccess(N) owner(nobody)
newres TERMINAL localhost defaccess(N) owner(nobody)
chres TERMINAL hostname defacc(N) owner(nobody)
```

防止这种破坏安全操作的另一种方法是限制来自本地主机的 Telnet、FTP 等的 TCP 请求。

另一个选项是将 TERMINAL 组的默认访问权限设置为 NONE，然后指定 TERMINAL 和 GTERMINAL 规则。

密码检查和登录限制

CA Access Control 不替换 /bin/login 可执行程序。 即使当 CA Access Control 正在运行时，也会根据 /etc/passwd、Shadow 密码文件或 NIS 密码映射继续检查密码。 但是 CA Access Control 还执行其他检查，如下节所述。

登录检查

登录进程通过身份验证阶段后，CA Access Control 会拦截该进程并检查下列几点：

- 密码是否已过期？

  如果密码已到期，则用户在被拒绝访问之前会收到宽限登录次数和警告。 在访问被拒绝后，安全管理员必须重新指定用户的密码。 宽限登录的次数根据用户密码策略确定，您可以使用 setoptions 命令以全局方式指定该策略，也可以使用 chgrp 命令为配置文件组指定该策略。

  注意：有关 setoptions 命令的详细信息，请参阅*《参考指南》*。

  您可以使用 segrace 实用程序查看用户的剩余宽限登录次数、用户的当前密码过期前剩余的天数，或者用户上次登录的日期和时间以及使用的终端。

  注意：有关 segrace 命令的详细信息，请参阅*《参考指南》*。

- 用户是否从已授权的终端登录？

  如果是，则登录按常规继续下一个检查；如果不是，则用户不能登录。

- 当前的工作日和工作时间是否允许登录（按照预定义的限制）？

  如果允许，则登录按常规继续下一个检查；否则，用户不能登录。

- 该用户名未使用的天数是否已超过预定义的天数？

  如果是，则拒绝访问。 （默认值为 90 天；使用 setoptions 命令可以更改该默认值。）

定义时间和日期登录规则

信息安全在活动少的时候最容易受到攻击。 深夜和周末是攻击的最佳时间，因为很少会有人来监视审核记录。 如果设置适当的终端权限规则，可以强制入侵者使用处于受保护位置的终端。 设置工作日 (DOW) 和工作时间 (TOD) 访问规则会强制入侵者在办公室处于开放和活动状态的工作时间期间进行攻击尝试。 该组合可以严格限制外来攻击。

限制用户可以登录的日期和时间需要基于每个用户。 要为某用户定义 DOW 和 TOD 登录限制，请使用以下命令：

```
chusr USR1 restrictions(days(Mon,Tue,Wed)time(800:1700))
```

该命令只允许用户 USR1 在星期一、星期二和星期三的 8:00 和 17:00 之间登录。 USR1 不能在指定日期的指定时间以外的时间登录，也不能在指定日期以外的日期登录。

days 参数还接受 ANYDAY 值（允许在一周所有七天登录）和 WEEKDAYS（允许从星期一到星期五登录）。 time 参数还接受 ANYTIME 值（允许在一天中的任何时间登录）。

注意：您可以将 DOW 和 TOD 限制应用到数据库中定义的*许多*资源。 当为终端和终端组提供有限的可用期时，该功能尤其有用。

禁用并发登录

大多数基于 UNIX 的操作系统允许并发登录。 但是，如果允许用户从多个终端登录，则存在这样的危险：当该用户登录时，其他用户可能会从其他地点伪装成该用户进行登录。

登录后，您可以通过 CA Access Control 禁用自己的并发登录权限，以使其他人不能以您的身份从另一终端登录。 但是，您仍可以从所使用的特定终端重复登录。 请使用 secons 命令和下列开关参数：

```
# secons -d-     (disables concurrent login)
# secons -d+     (enables concurrent login)
```

任何用户都可以发出 -d 选项。 （所有其他选项只能用于具有 ADMIN 或 OPERATOR 属性的用户）。 想禁用并发登录的用户可以在其初始脚本中使用该命令。 虽然他们之后能够打开所需的任意数量的窗口，但是不能从其他终端登录。

注意：如果您使用 secons -d- 命令来阻止并发登录，必须记得在注销之前使用 secons -d+ 以避免被锁定在系统之外。 如果忘记恢复并发登录而尝试再次登录，则 CA Access Control 允许您登录（如果没有具有相同用户 ID 的进程正在运行）。

限制用户的并发登录

CA Access Control 可以用下列两种方法控制并发登录的数量：

管理员级别

在数据库中设置用户可以拥有的并发会话数量的系统范围定义。 可以使用全局方式为配置文件组或单个用户设置该值。

用户级别

用户单独控制允许自己使用的并发登录的数量。 这样，用户在登录时，可以阻止使用他们名称的更多登录会话的选项，从而保护了他们自己。

注意：并发登录的数量与用户在特定终端上运行的会话数无关。 一个终端上的多个会话被视为一次登录。 并发登录限制会限制用户可以从其并发登录的*终端*的数量，而不是从每个终端的登录次数。

限制全局并发登录

输入以下 selang 命令：

```
setoptions maxlogins(NumLogins)
```

限制单个并发登录

输入以下 selang 命令：

```
chusr username maxlogins(NumLogins)
```

为用户设置的并发登录限制会覆盖系统范围的限制。 要防止 CA Access Control 强制执行特定用户的并发登录限制，请将该用户的并发登录限制设置为零。 （注意，如果将最大并发登录数设置为一，则不能使用 selang。）

识别登录事件

CA Access Control 不会将更改进程用户 ID 的所有尝试视为登录事件。 通常，程序使用 setuid 系统调用来尝试更改其用户 ID。 SURROGATE 类控制这些事件，从 CA Access Control 角度而言，不必将这些事件视为登录事件，也不需要更改用户身份。

CA Access Control 始终保留原始用户身份，即用户最初登录时使用的身份。 普通 setuid 系统调用不会使 CA Access Control 注册用户身份的更改。

CA Access Control 要识别身份更改，必须将该事件识别为登录事件。 它使用下列规则识别登录事件：

- 尝试更改身份的程序被定义为*登录程序*。 LOGINAPPL 类中的所有程序都是登录程序。
- 该程序执行与 LOGINAPPL 类中其定义相对应的一系列系统调用。

当通过 selang 或 CA Access Control 端点管理 开始管理会话时，CA Access Control 执行虚拟登录事件。 这不是真实的登录；而是 CA Access Control 执行与登录检查类似的一些内部检查。

注意：有关详细信息，请参阅*《selang 参考指南》*中 LOGINAPPL 类的 SEQUENCE 属性。

管理会话启动时，将在要管理的计算机中检查用户名。 只有当您对执行会话所在的终端拥有写入访问权限时，才能获取对该计算机的访问权限以进行管理。

例如，如果登录到主机 Minerva 并且希望在主机 Artemis 上管理 CA Access Control，则需要满足下列两个条件：

- 名为 Minerva（或相关的完全限定名）的 TERMINAL 对象位于 Artemis 的数据库记录中。
- 您在该对象的 ACL 中被列出，并且您拥有 WRITE 权限。

在进行任何其他用户权限检查前，会检查这些条件。 注意，您还需要数据库的管理权限。

第 10 章： 保护 TCP/IP 服务

对于包含敏感数据的文件服务器，保护 TCP/IP 服务是极其重要的。 这些服务器必须仅向受托工作站提供一些服务，而绝不能向主机未知的入侵者或计算机提供。

此部分包含以下主题：

限制 TCP/IP 服务

在开放网络中，任何工作站都可以请求网络上其他计算机的服务。 可以使用 TCP/IP 协议提供许多服务。 其中一些服务（例如 rlogin、rcp、rsh、ftp、telnet 和 rexec 等）对所有基于 UNIX 的操作系统来说是通用的。 其他服务由内部和第三方软件提供。

CA Access Control 在主机上拦截 TCP/IP 的接受过程，并确定应该以正常方式继续还是覆盖接受程序。 CA Access Control 根据管理您定义的主机和服务的访问规则做出决策。 您可以在数据库中创建 TCP/IP 访问规则，以指定允许接收来自特定计算机的服务（例如文件传输、远程登录以及远程 shell）的计算机和网络。

以下示例显示如何定义和设置 TCP/IP 访问规则，以便有效地阻止不受欢迎的外来者。 如果您还没有时间开发一个完善的数据库，则可能需要使没有在数据库中定义的某个工作站来接收所有服务。 如果是这样，请设置 UACC 类中的 HOST 记录，如下所示：

```
chres UACC HOST defaccess(READ)
```

具有本地主机 TCP/IP 服务访问规则的工作站在数据库中 HOST 类下的记录中定义。 对于其中的每个工作站，将在记录中列出允许的服务。 例如，下列命令序列定义工作站 ws5 的记录，并禁止它接收来自本地主机的任何 TCP/IP 服务：

```
newres HOST ws5
authorize HOST ws5 service(*) access(NONE)
```

下列命令允许 ws5 对本地计算机执行 telnet：

```
authorize HOST ws5 service(telnet)
```

这些设置允许用户通过 telnet 访问本地计算机，这意味着远程用户必须先指定用户名和密码才能使用本地系统。 为了使工作站可以接收来自本地计算机的所有 TCP/IP 服务，您可以在服务关键字中使用星号。 例如，下列命令允许 ws5 调用来自本地计算机的任何 TCP/IP 服务：

```
authorize HOST ws5 service(*)
```

可以使用几种方法指定服务，其中有些方法需使用*端口号*。 端口号是服务的标识号。 所有服务都有端口号，而端口号被映射到文件 /etc/services 中的服务。 您可以采用下列方式指定服务：

- 按文件 /etc/services 中定义的名称
- 按其端口号
- 按照端口号的范围
- 按照 /etc/rpc 系统文件中列出的 RPC 端口

例如，下列命令允许 ws5 接收端口号在 7045 和 7050 之间的任何 TCP/IP 服务：

```
authorize HOST ws5 service(7045-7050)
```

在许多情况下，定义主机组并一次设置其权限而非分别为每台计算机设置权限，这种方式更为经济。 CA Access Control 提供 GHOST 类，其中每个 GHOST 记录都定义一个主机组。 要定义一个 GHOST 记录并向其成员列表中添加主机，请输入以下命令：

```
newres GHOST gh1 mem(ws2, ws3, ws5)
authorize GHOST gh1 service(ftp)
```

newres 命令定义包含成员 ws2、ws3 和 ws5 的主机组 gh1。 authorize 命令允许全部三个工作站接收 ftp（文件传输）服务。

管理主机组比管理单个工作站更容易，但为了提供更大的灵活性，CA Access Control 还支持定义网络访问规则。 在 HOSTNET 类中定义网络。例如，考虑下面一组命令：

```
newres HOSTNET hn1 mask(255.555.0.0) match(192.168.0.0)
authorize HOSTNET hn1 service(*) access(NONE)
authorize HOSTNET hn1 service(ftp)
```

- 在第一行中，newres 命令定义名为 hn1 的网络。 通过掩码和匹配值，它指定 IP 地址中前两个限定符为 192.168 的任何计算机都被视为来自 hn1 网络。
- 第二行和第三行的组合允许 hn1 网络中的任何工作站执行 ftp，但不允许执行主机中的任何其他服务。

CA Access Control 提供的另一种定义 TCP/IP 访问规则的方法是名称-模式访问规则。 CA Access Control 支持在 HOSTNP 类（主机名模式）中使用通配符定义通用记录。

注意：有关 CA Access Control 如何执行字符串匹配的信息，请参阅*《selang 参考指南》*。

例如，下列命令序列允许名称以字符“lin”开始并以字符“.org.com”结束的任何主机接受本地主机上的所有 TCP/IP 服务：

```
newres HOSTNP lin*.org.com
authorize HOSTNP lin*.org.com service(*).
```

注意：由 NIS 管理的主机必须按它们在 NIS 映射中显示的正式名称而不是别名来标识。 下一节中的图表总结了 TCP/IP 检查流程。

使用 TCP 类

另外，您可以使用 TCP 类按服务而不是按主机指定保护。

注意：有关 TCP 类的详细信息，请参阅*《参考指南》*。

使用 *TCP* 类来控制传入和传出服务。

例如，下列命令创建 ftp 服务的记录并将 READ（意味着服务可以使用）作为默认的访问权限类型，但阻止与名称模式 PUBLIC* 匹配的主机接收服务。

```
newres TCP ftp defaccess(READ)
authorize- TCP ftp hostnp(PUBLIC*) access(N)
```

您还可以指定只允许特定用户或组接收特定服务。 例如，要允许所有用户通过 ftp 访问名为 hermes 主机，但要指定只有名为 acctng 组的成员可以通过 telnet 访问 hermes，请输入以下命令：

```
newres HOST hermes
newres TCP ftp owner(nobody) defaccess(read)
newres TCP telnet owner(nobody) defaccess(read)
authorize TCP ftp uid(*) host(hermes) access(write)
authorize TCP telnet gid(acctng) host(hermes) access(write)
```

注意：defaccess(read) 禁用传出服务。 defaccess(write) 禁用传入的服务。

如果 HOST 类处于活动状态（即如果它用作访问的条件），则 TCP 类无法有效地处于活动状态。 您可以使用命令 setoptions class- HOST 来取消激活 HOST 类，然后使用命令 setoptions class+ TCP（如有必要）来激活 TCP 类。 如果取消激活 HOST 类，将自动取消激活 GHOST、HOSTNET 以及 HOSTNP。

而且，如果 TCP 类处于活动状态，则使用 setoptions 命令 class-CONNECT 来停止 CONNECT 类。

执行网络截获的数据流模块

默认情况下，TCP 类并不处于活动状态。 在您激活 TCP 类、CONNECT 类或 HOST 类之前，请确保启用了数据流模块。

要在 Solaris 上加载 CA Access Control 数据流模块，请完成以下步骤：

1. 停止 CA Access Control。 输入下面的命令：

   ```
   secons -s
   ```

2. 输入下面的命令：

   ```
   SEOS_load -s
   ```

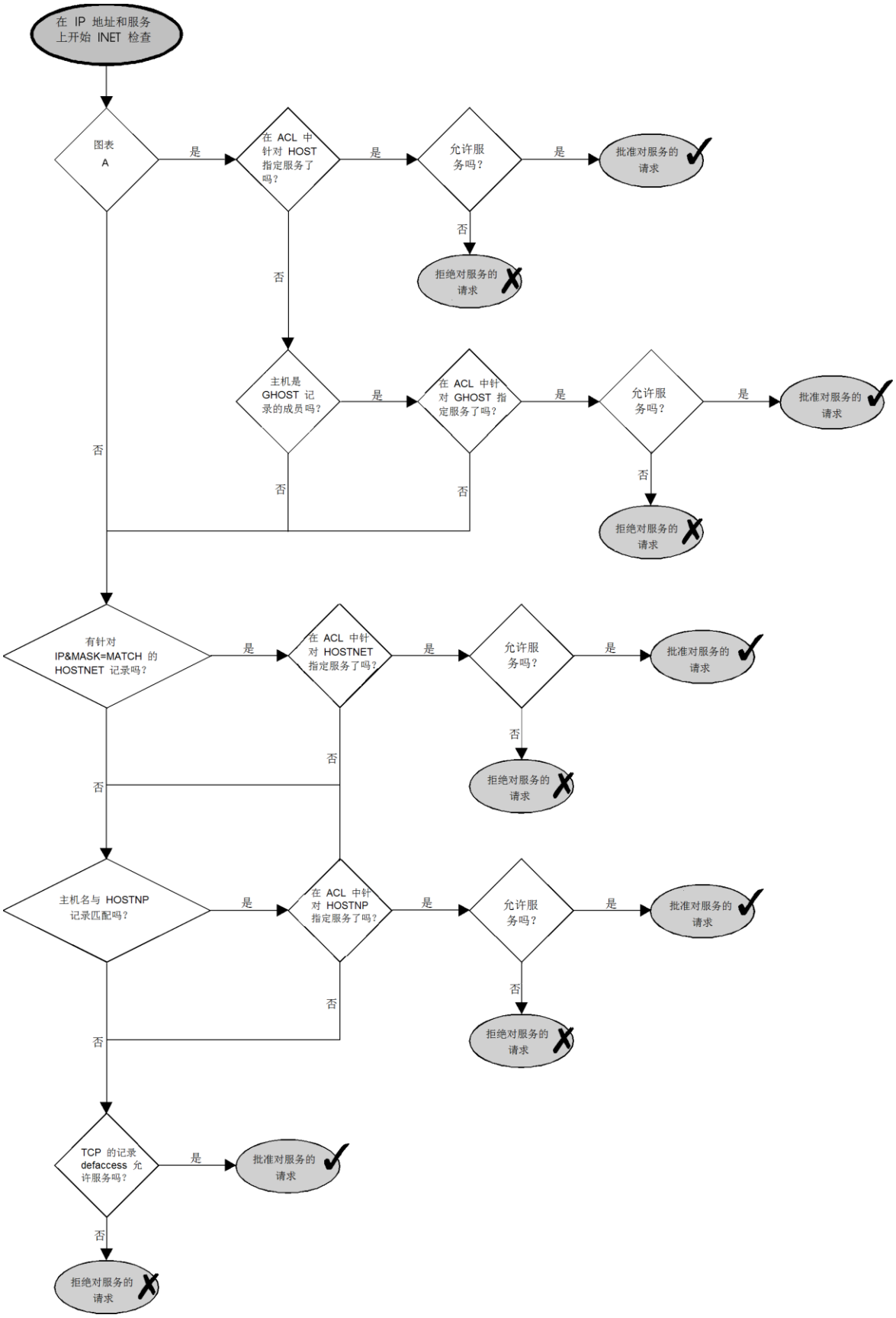
3. 启动 CA Access Control。 输入下面的命令：

   ```
   seload
   ```

注意：如果您尝试在未加载数据流模块时激活 TCP 类，则将出现以下错误：

```
错误：未加载数据流时，不能激活 className 类。
请使用 SEOS_load -s 加载数据流。
```

传入授权的算法是：


在 IP 地址和服务上开始 INET 检查
图表 A
是
在 ACL 中针对 HOST 指定服务了吗？
是
允许服务吗？
是
批准对服务的请求
否
拒绝对服务的请求
否
主机是 GHOST 记录的成员吗？
是
在 ACL 中针对 GHOST 指定服务了吗？
是
允许服务吗？
是
批准对服务的请求
否
否
否
拒绝对服务的请求
否
有针对 IP&MASK=MATCH 的 HOSTNET 记录吗？
是
在 ACL 中针对 HOSTNET 指定服务了吗？
是
允许服务吗？
是
批准对服务的请求
否
否
拒绝对服务的请求
否
主机名与 HOSTNP 记录匹配吗？
是
在 ACL 中针对 HOSTNP 指定服务了吗？
是
允许服务吗？
是
批准对服务的请求
否
否
拒绝对服务的请求
否
TCP 的记录 defaccess 允许服务吗？
是
批准对服务的请求
否
拒绝对服务的请求

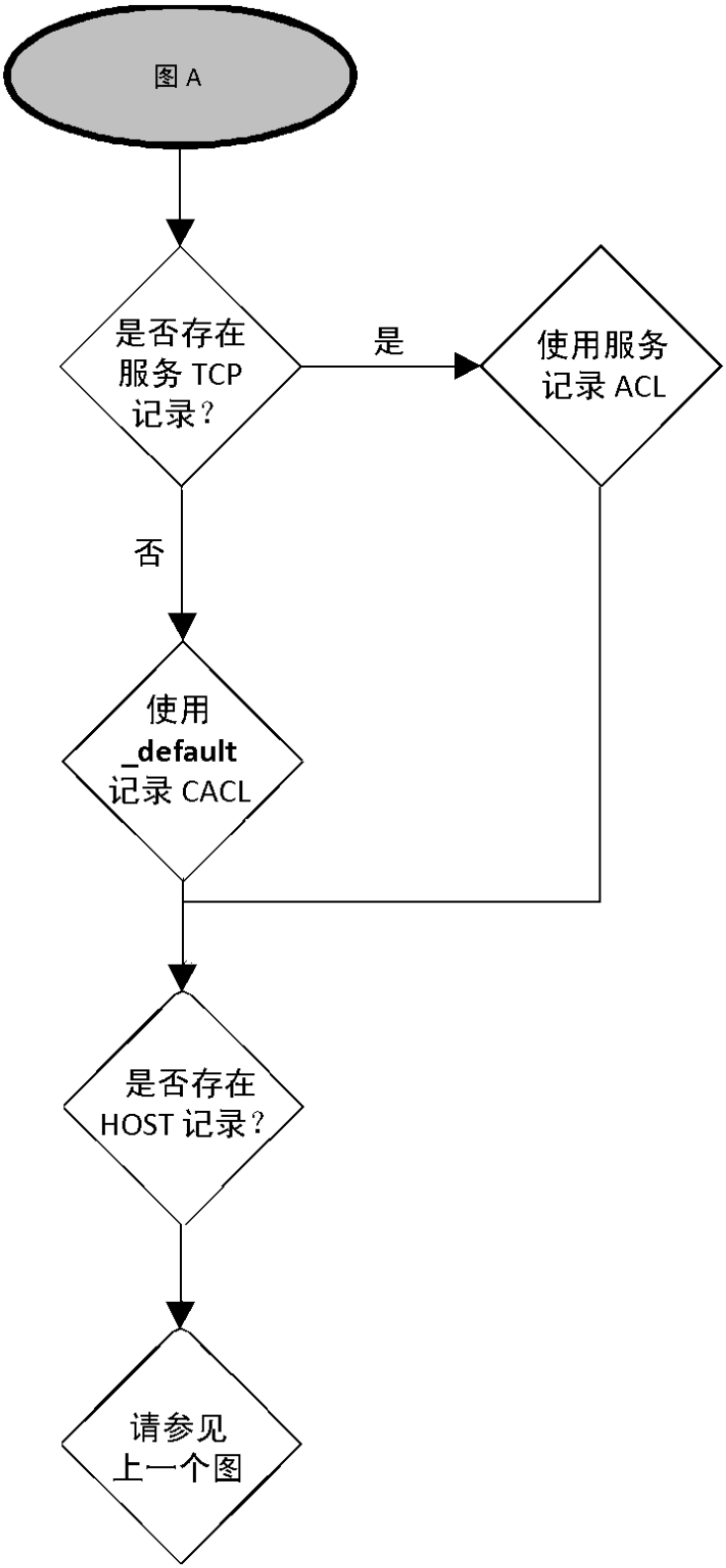
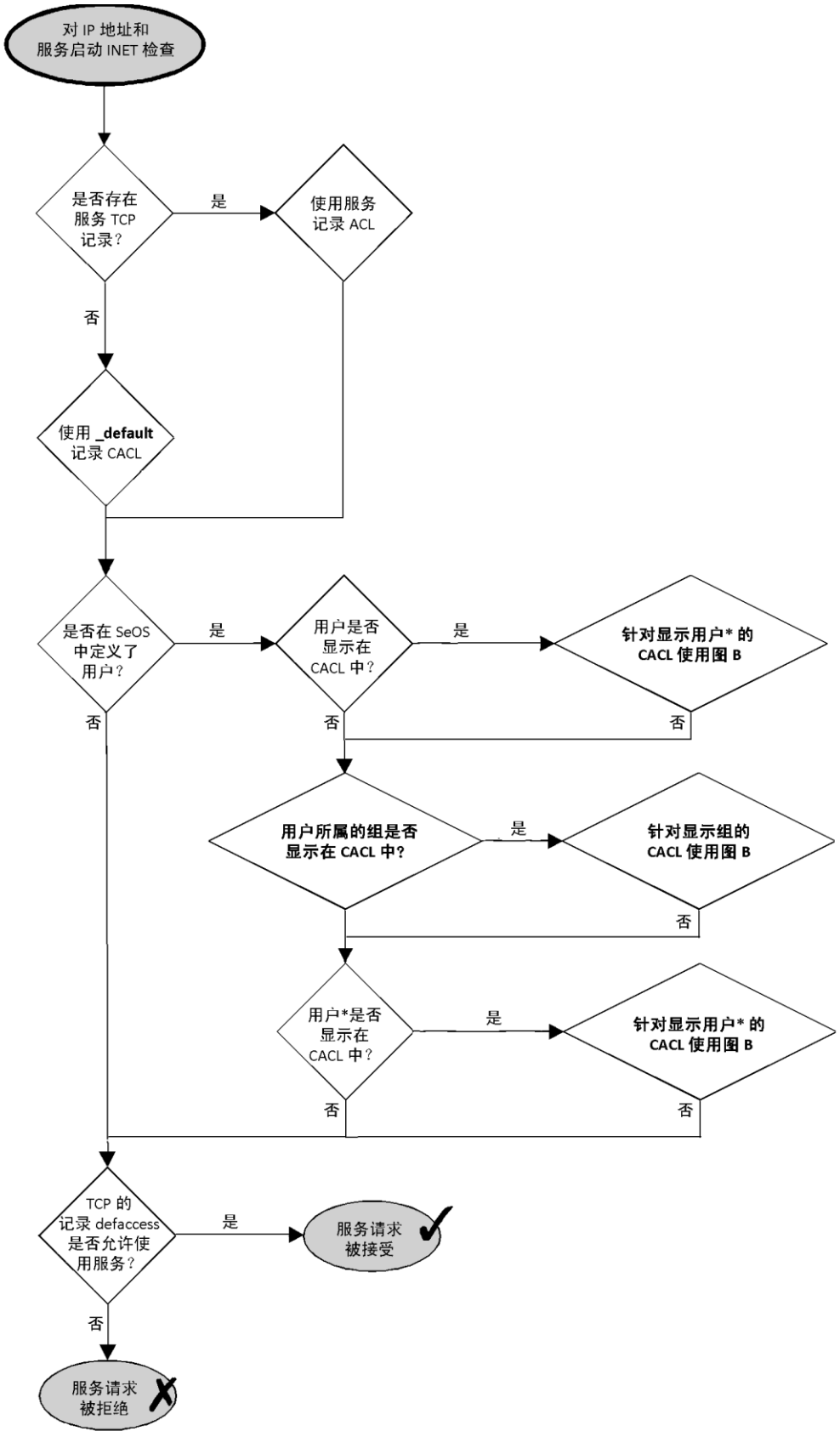


图 A
是否存在服务 TCP 记录？
是
使用服务记录 ACL
否
使用 _default 记录 CACL
是否存在 HOST 记录？
请参见上一个图

传出授权的算法是：


对 IP 地址和
服务启动 INET 检查
是否存在
服务 TCP
记录？
是
使用服务
记录 ACL
否
使用 _default
记录 CACL
是否在 SeOS
中定义了
用户？
是
用户是否
显示在
CACL 中？
是
针对显示用户* 的
CACL 使用图 B
否
否
否
用户所属的组是否
显示在 CACL 中?
是
针对显示组的
CACL 使用图 B
否
用户*是否
显示在
CACL 中？
是
针对显示用户* 的
CACL 使用图 B
否
否
TCP 的
记录 defaccess
是否允许使
用服务？
是
服务请求
被接受
否
服务请求
被拒绝

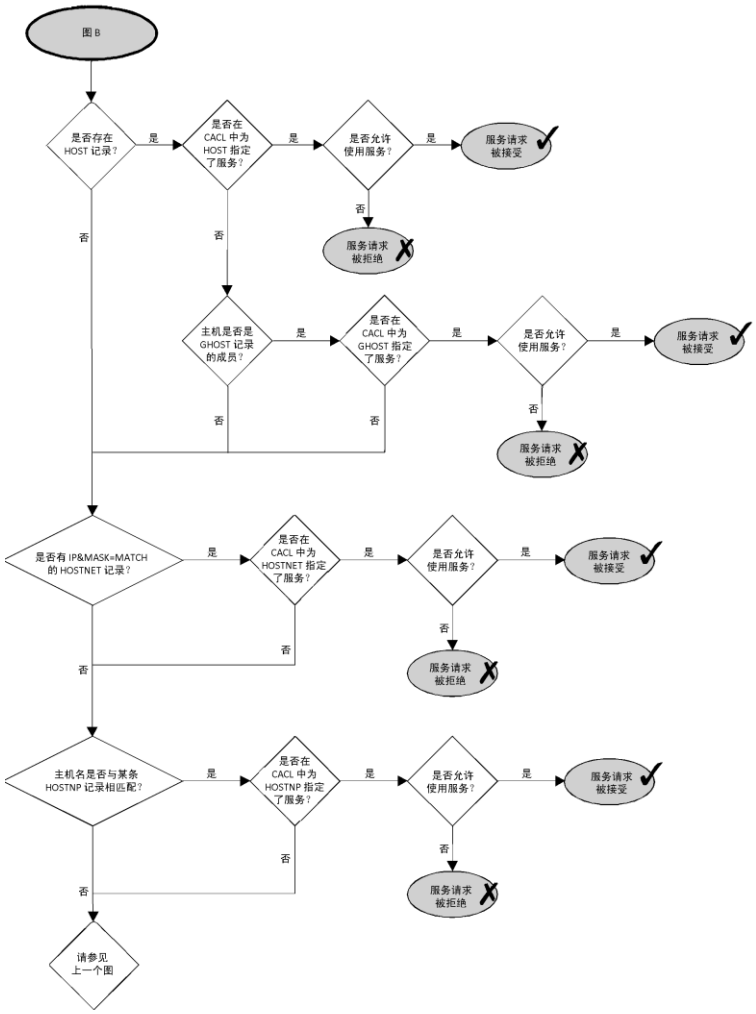


图 B
是否存在 HOST 记录？
是
是否在 CACL 中为 HOST 指定了服务？
是
是否允许使用服务？
是
服务请求被接受
否
服务请求被拒绝
否
否
主机是否是 GHOST 记录的成员？
是
是否在 CACL 中为 GHOST 指定了服务？
是
是否允许使用服务？
是
服务请求被接受
否
服务请求被拒绝
否
否
是否有 IP&MASK=MATCH 的 HOSTNET 记录？
是
是否在 CACL 中为 HOSTNET 指定了服务？
是
是否允许使用服务？
是
服务请求被接受
否
服务请求被拒绝
否
否
主机名是否与某条 HOSTNP 记录相匹配？
是
是否在 CACL 中为 HOSTNP 指定了服务？
是
是否允许使用服务？
是
服务请求被接受
否
服务请求被拒绝
否
否
请参见上一个图

第 11 章： 管理策略模型

此部分包含以下主题：

策略模型数据库

单独管理数十或数百个数据库并不现实。 CA Access Control 提供策略模型服务，通过该服务组件，您可以通过一个中央数据库管理多个数据库。 使用策略模型服务是可选的，但是它将极大地简化大型站点上的管理。

策略模型 (PMD) 服务使用策略模型数据库 (PMDB)。 与其他 CA Access Control 数据库一样，PMDB 包含用户、组、受保护的资源和管理资源访问的规则。 此外，PMDB 还包含一个 *订户*数据库列表。 每个订户都是位于单独计算机上的 CA Access Control 数据库，或是位于同一或不同计算机上的另一 PMDB。 更新订户的 PMDB 是订户的父项。

在管理具有类似权限限制和访问规则的多个数据库方面，PMDB 是一个非常有用的工具。

注意：有关管理 PMDB（sepmd 实用程序）的信息，请参阅*《参考指南》*。有关使用 selang 远程管理 PMDB 的信息，请参阅 *《selang 参考指南》*。

磁盘上的 PMDB 位置

所有 PMDB 都位于公共目录中（每台计算机都有一个公共目录）。 该目录的名称由 seos.ini 文件中 [pmd] 部分的 _pmd_directory_ 标记指定。 _pmd_directory_ 的默认值是 *ACInstallDir*/policies，其中，*ACInstallDir* 是 CA Access Control 的安装目录（默认情况下为 /opt/CA/AccessControl/）。

每个 PMDB 都占用公共目录中的一个子目录。 子目录的名称是策略模型的名称。 子目录中的文件包含定义策略模型必需的所有数据，包括 pmd.ini 文件。

管理本地 PMDB

CA Access Control 提供若干用于管理本地 PMDB 的实用程序：

sepmd

您可以使用 PMDB 管理实用程序来执行以下任务：

- 管理订户
- 截短更新文件
- 管理双重控制
- 管理策略模型日志文件
- 执行其他管理任务

sepmdadm

创建 PMDB，并配置它们具有用于设置层级结构的必要设置。

注意：有关策略模型实用程序的深入讨论，请参阅*《参考指南》*。

管理远程 PMDB

CA Access Control 还提供一系列可在 pmd 环境中使用的 selang 命令。 通过这些命令，您可以远程管理 PMDB：

backuppmd

备份 PMDB。

createpmd

创建 PMDB。

deletepmd

删除 PMDB。

findpmd

显示计算机上所有 PMDB 的名称。

listpmd

列出以下关于 PMDB 的信息：

- 订户及其状态
- PMDB 描述及其状态
- 更新文件中的命令及其偏移量
- 错误日志的内容

pmd

您可以使用 PMDB 管理命令来执行以下任务：

- 从不可用订户列表中删除订户
- 清除策略模型错误日志
- 锁定和解除锁定策略模型
- 启动和停止策略模型后台程序
- 截短更新文件
- 重新加载初始化文件

restorepmd

从其备份文件还原 PMDB。

subs

您可以使用 PMDB 订阅命令执行以下任务：

- 将现有的订户添加到父 PMDB 中
- 将新订户添加到父 PMDB 中
- 为数据库（CA Access Control 或另一 PMDB）指定父 PMDB

subspmd

为本地数据库指定父 PMDB。

unsubs

从 PMDB 中删除订户。

注意：有关可在 pmd 环境中使用的 selang 命令的深入讨论，请参阅 *《selang 参考指南》*。

体系结构相关性

部署 CA Access Control 时，您应考虑环境的层级结构。 在许多站点上，网络具有各种体系结构。 某些策略规则（例如受托程序列表）与体系结构相关。 另一方面，大多数规则都与系统体系结构无关。

您可以使用层级结构来包括这两种规则。 您可以为与体系结构无关的规则定义一个全局数据库，向它提供定义与体系结构相关的规则的订户 PMDB。

注意：根 PMBD 及其所有订户可以位于同一计算机或不同计算机上，这取决于您环境的实际需要。

示例：一个两层的部署层级结构

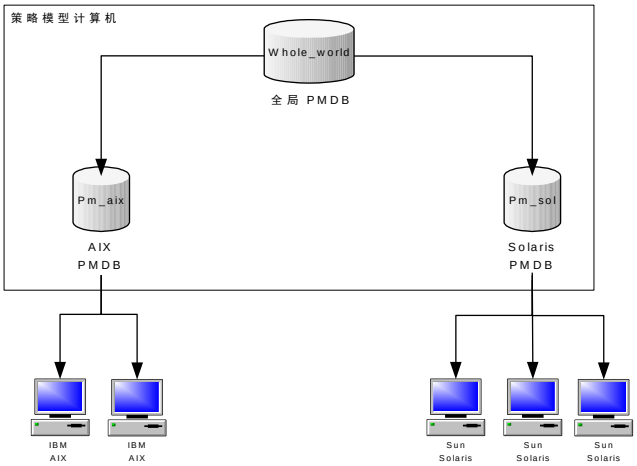
下面的 UNIX 示例也适用于 Windows 体系结构，不过需要做一些小修改。

在该示例中，站点包括 IBM AIX 和 Sun Solaris 系统。 由于 IBM AIX 上的受托程序列表与 Sun Solaris 上的受托程序列表不同，因此 PMDB 需要考虑体系结构的依赖性。

要设置多体系结构 PMDB，请按以下步骤设置 PMDB：

1. 定义名为 whole_world 的 PMDB，以包含用户、组和所有其他与体系结构无关的策略。
2. 定义名为 pm_aix 的 PMDB，以包含所有特定于 IBM AIX 的规则。
3. 定义名为 pm_sol 的 PMDB，以包含所有特定于 Sun Solaris 的规则。

   名为 pm_aix 和 pm_solaris 的 PMDB 是名为 whole_world 的 PMDB 的订户。 站点上的所有 IBM AIX 计算机都是 pm_aix 的订户。 站点上的所有 Sun Solaris 计算机都是 pm_sol 的订户。 下面以图表说明该概念。



4. 当您在 whole_world 中输入与平台无关的命令，例如添加用户或设置 SURROGATE 规则，则自动更新站点上的所有数据库。
5. 当您向 pm_aix 中添加受托程序时，只更新 IBM AIX 计算机，而不会影响 Sun Solaris 系统。

集中管理策略的方法

您可以使用 CA Access Control 采用以下方式从一台计算机管理多个数据库：

- **基于规则的自动策略更新** - 您在中央数据库 (PMDB) 中定义的常规规则会自动传播给已配置的层级结构中的数据库。

  注意：仅此方法提供双重控制 (p. 146)，并且仅适用于 UNIX。 有关基于规则的自动策略更新的双重控制信息，可以在*《端点管理指南：用于 UNIX》*上找到。 有关基于规则的自动策略更新的信息，可以在*《端点管理指南：用于 Windows》*上找到。

- **高级策略管理** - 根据主机或主机组分配将您部署的策略（规则组）传播到所有数据库。 您还可以取消部署（删除）策略以及查看部署状态和部署偏差。 要使用此功能，您需要安装并配置额外的组件。

  注意：有关高级策略管理的信息，可以在*《企业管理指南》*中找到。

基于规则的自动策略更新

您在中央数据库中进行的单一规则策略更新（常规 selang 规则）将自动传播给订户数据库。 通过将若干计算机订阅到同一数据库，以及将一个数据库订阅到另一数据库，您可以创建层级结构。 安装后，您可以为基于规则的自动策略更新配置您的环境。

注意：这种管理策略的方法限于使您在整个层级结构中进行单一规则策略更新。 其他功能只能通过实施高级策略管理和报告来使用。

基于规则的自动策略更新原理

为基于规则的自动策略更新配置环境时，您在中心数据库中定义的每条规则自动通过以下方式传播给它的所有订户：

1. 必须为至少有一个订户的任何 PMDB 定义一条规则。
2. PMDB 向所有订户数据库发送命令。

3. 订户数据库应用传播的命令。
   a. 如果订户数据库没有响应，则 PMDB 按规定时间间隔（默认情况下为每隔 30 分钟）发送命令，直到更新了订户数据库为止。

      另外，您可以在订户数据库可用后立即更新它们，其方法是在订户计算机的 seos.ini 文件的 [pmd] 部分中，将 pull_option 标记设置为 yes。

   b. 如果订户数据库正在响应，但拒绝应用命令，则 PMDB 会将命令放在策略模型错误日志 (p. 141)中。
4. 如果订户数据库是其他订户的父项，则将命令发送给它的订户。

示例：从层级结构中的所有计算机上删除用户

如果使用 rmusr 命令从 PMDB 删除用户，则相同的 rmusr 命令将发送至所有订户数据库。 这样，一个 rmusr 命令即可从各台计算机上的多个数据库中删除用户。

您使用 PMDB 来传播配置设置的方式

当您编辑策略模型的配置时，新的配置值会被传播给策略模型的订户。

以下过程说明了配置更新被传播给策略模型的订户的方式：

1. 编辑一个或多个策略模型的配置值。
2. 策略模型将新的配置值写入虚拟配置文件。

   注意：虚拟配置文件不包含 audit.cfg 文件的值。 策略模型不会将您对该文件所做的任何更改写入到虚拟配置文件。
3. 策略模型将新的配置值发送给订户。
4. selang 命令使用新的配置值更新每位订户。

虚拟配置文件

每个策略模型都有虚拟配置文件，其中包含其订户的配置值。 虚拟配置文件位于 PMD 目录，并命名为 cfg_*configname*，其中 *configname* 是策略模型配置的名称。

虚拟配置文件不包含 audit.cfg 文件中所具有的配置值。

新订户的配置方式

策略模型使用现有的配置值配置每位新订户。 现有的配置值存储在虚拟配置文件中。

注意：虚拟配置文件不存储 audit.cfg 文件中的配置值。 您在创建新订户前对 audit.cfg 文件所做的任何更改都不会传播给新订户。

以下过程说明策略模型配置新订户的方式：

1. 创建一个新订户到策略模型。
2. 策略模型读取其虚拟配置文件中的值。
3. 策略模型将配置值从其虚拟配置文件添加到 updates.dat 文件。updates.dat 文件还包含策略的访问规则。
4. 策略模型将 updates.dat 文件发送到新订户。
5. selang 命令使用 updates.dat 文件中的值配置新订户。

如何能够设置层级结构

CA Access Control 使用策略模型服务在已配置的层级结构中传播基于规则的策略更新。 通过为同一 PMDB 订阅多台 CA Access Control 计算机，以及通过为一个 PMDB 订阅另一 PMDB，您可以创建层级结构。

要启用基于规则的自动策略更新，请执行以下操作：

1. 创建和配置主 PMDB (p. 129)。
2. （可选）创建和配置订户 PMDB (p. 131)。
3. 为订阅计算机（称为*端点*）定义父 PMDB。 (p. 134)

注意：以下几节说明如何设置 PMDB 层级结构。 有几种创建 PMDB 并设置其层级结构的其他方法。 有关策略模型实用程序的深入讨论，请参阅*《参考指南》*。

创建和配置主 PMDB

要从中央位置管理策略，您首先需要创建并配置主 PMDB。 要在本地主机上执行此操作，可以使用 sepmdadm 命令。

注意：以下过程说明了 sepmdadm 命令的交互形式。 有关使用所有输入的命令行参数的信息，请参阅《*参考指南*》。

创建和配置主 PMDB

1. 在命令行中，输入以下命令：

   ```
   sepmdadm -i
   ```

   CA Access Control 将启动策略模型数据库管理脚本 (sepmdadm)，并显示您可以从中选择选项的菜单。

2. 输入 1 以选择第一个选项（创建主 PMDB 并定义其用户）。

   该脚本配置为提问您相关问题。

3. 按 Enter 键继续。

   脚本继续询问您第一个问题。

   注意：如果 CA Access Control 未运行，则脚本将发出警告，要求您先启动 CA Access Control 再重新运行脚本。

4. 输入您要创建的策略模型的名称。

   该脚本注册策略模型名称并继续。

5. 输入您要指定的第一个订户计算机的名称。

   脚本注册第一个订户的名称，然后请求您输入下一个订户的名称。

6. 根据需要继续输入订户名称，然后按 Enter 键，而无需输入订户名称。

   脚本注册所有订户并继续。

   注意：您仍必须将每台订户计算机指向其父 PMDB。

7. 如果您正在运行 NIS、NIS+ 或 DNS，请选择您是否要用 PMDB 更改来更新 NIS/DNS 表。

   将对 PMDB 中的用户和组进行更新。 表中提供了有关用户及其特性的信息。 如果选择 yes，则通过策略模型更新的 UNIX 用户或 UNIX 组也将在 NIS 密码文件和组文件中进行更新。

   a. 如果您要更新 NIS/DNS 表，请输入 **y**。

      脚本立即询问您 NIS passwd 和组文件的位置。

      a. 输入 NIS 密码文件的完整路径。

         脚本注册完整路径并继续。

      b. 输入 NIS 组文件的完整路径。

         脚本注册完整路径并继续。

   b. 如果您要更新 NIS/DNS 表，请输入 **n** 或按 Enter 键。

      脚本注册您的回答并继续。

8. 输入您要为其提供 PMDB 的特殊属性的用户：
   a. 根据需要输入 CA Access Control 管理员名称，输入后按 Enter 键。

      管理员已获得授权，可以更改 PMDB 的属性。

      注意：必须在 PMDB 中至少定义一个管理员（*root* 为默认值）。
   b. 根据需要输入企业用户管理员名称，输入后按 Enter 键。
   c. 根据需要输入 CA Access Control 审核员名称，输入后按 Enter 键。

      审核员经过授权可查看 PMDB 的审核日志文件。
   d. 根据需要输入企业用户审核员名称，输入后按 Enter 键。
   e. 根据需要输入 CA Access Control 密码管理员名称，输入后按 Enter 键。
   f. 根据需要输入企业用户密码管理员名称，输入后按 Enter 键。

      密码管理员经过授权可更改 PMDB 中的密码。

   脚本注册您的回答并继续。
9. 根据需要输入管理终端名称，然后按 Enter 键而无需输入管理终端的名称。

   脚本注册所有管理终端，然后报告您所做的选择并要求确认。
10. 按 Enter 键确认您所做的选择，或输入 **n** 以使用新输入重新运行脚本。

    如果您确认了所做的选择，则会使用您提供的回答来创建新的 PMDB。

更多信息：

创建和配置订户 PMDB (p. 131)
为订阅计算机定义父 PMDB (p. 134)

创建和配置订户 PMDB

配置主 PMDB 后，如果您要扩展层级结构，则需要创建和配置订户 PMDB。 要在本地主机上执行此操作，可以使用 sepmdadm 命令。

注意：以下过程说明了 sepmdadm 命令的交互形式。 有关使用所有输入的命令-行参数的信息，请参阅 *《参考指南》*。

创建和配置订户 PMDB

1. 在命令行中，输入以下命令：

   ```
   sepmdadm -i
   ```

   CA Access Control 将启动策略模型数据库管理脚本 (sepmdadm)，并显示您可以从中选择选项的菜单。

2. 输入 2 以选择第二个选项（创建子 PMDB 并定义其订户和父项）。

   该脚本配置为提问您相关问题。

3. 按 Enter 键继续。

   脚本继续询问您第一个问题。

   注意：如果 CA Access Control 未运行，则脚本将发出警告，要求您先启动 CA Access Control 再重新运行脚本。

4. 输入您要创建的策略模型的名称。

   该脚本注册策略模型名称并继续。

5. 输入您要指定的第一个订户计算机的名称。

   脚本注册第一个订户的名称，然后请求您输入下一个订户的名称。

6. 根据需要继续输入订户名称，然后按 Enter 键，而无需输入订户名称。

   脚本注册所有订户并继续。

   注意：您仍必须将每台订户计算机指向其父 PMDB (p. 134)。

7. 输入父 PMDB 的名称。

   脚本注册父 PMDB 的名称并继续。

   注意：sepmdadm 只允许您为每个订阅数据库输入一个父项。 不过，您可以为每个数据库定义多个父项。 为此，请修改 pmd.ini 配置文件的 parent_pmd 标记。 有关使用此标记的详细信息，请参阅*《参考指南》*。

8. 如果您正在运行 NIS、NIS+ 或 DNS，请选择您是否要用 PMDB 更改来更新 NIS/DNS 表。

   将对 PMDB 中的用户和组进行更新。 表中提供了有关用户及其特性的信息。 如果选择 yes，则通过策略模型更新的 UNIX 用户或 UNIX 组也将在 NIS 密码文件和组文件中进行更新。

   a. 如果您要更新 NIS/DNS 表，请输入 **y**。

      脚本立即询问您 NIS passwd 和组文件的位置。

      a. 输入 NIS 密码文件的完整路径。

         脚本注册完整路径并继续。

      b. 输入 NIS 组文件的完整路径。

         脚本注册完整路径并继续。

   b. 如果您要更新 NIS/DNS 表，请输入 **n** 或按 Enter 键。

      脚本注册您的回答并继续。

9. 输入您要为其提供 PMDB 的特殊属性的用户：

   a. 根据需要输入 CA Access Control 管理员名称，输入后按 Enter 键。

      管理员已获得授权，可以更改 PMDB 的属性。

      注意：必须在 PMDB 中至少定义一个管理员（*root* 为默认值）。

   b. 根据需要输入企业管理员名称，输入后按 Enter 键。

   c. 根据需要输入 CA Access Control 审核员名称，输入后按 Enter 键。

      审核员经过授权可查看 PMDB 的审核日志文件。

   d. 根据需要输入企业用户审核员名称，输入后按 Enter 键。

   e. 根据需要输入 CA Access Control 密码管理员名称，输入后按 Enter 键。

      密码管理员经过授权可更改 PMDB 中的密码。

   f. 根据需要输入企业用户密码管理员名称，输入后按 Enter 键。

   脚本注册您的回答并继续。

10. 根据需要输入管理终端名称，然后按 Enter 键而无需输入管理终端的名称。

    脚本注册所有管理终端，然后报告您所做的选择并要求确认。

11. 按 Enter 键确认您所做的选择，或输入 **n** 以使用新输入重新运行脚本。

    如果您确认了所做的选择，则会使用您提供的回答来创建新的 PMDB。

为订阅计算机定义父 PMDB

要将端点计算机建立为 PMDB 的订户，除了在 PMDB 中注册订户名称外，您必须执行其他操作。 您还需要在订户计算机上完成一个过程。

为订阅计算机定义父 PMDB

1. 在订户计算机的命令行中，以交互模式启动 sepmdadm 命令：

   ```
   sepmdadm -i
   ```

   CA Access Control 将启动策略模型数据库管理脚本 (sepmdadm)，并显示您可以从中选择选项的菜单。

2. 输入 3 以选择第三个选项（定义本地主机的父 PMDB 和密码 PMDB）。

   该脚本配置为提问您相关问题。

3. 按 Enter 键继续。

   该交互脚本继续询问您第一个问题。

   注意：如果 CA Access Control 正在运行，则脚本将发出警告，要求您先停止 CA Access Control 再重新运行脚本。

4. 输入父 PMDB 的名称。

   脚本注册父 PMDB 的名称并继续。

5. 输入父密码 PMDB 的名称。

   脚本注册父密码 PMDB 的名称，然后报告您所做的选择并要求确认。

6. 按 Enter 键确认您所做的选择，或输入 **n** 以使用新输入重新运行脚本。

   如果您确认了选择，则使用这些输入来设置订户计算机。

注意：sepmdadm 只允许您为每个订阅数据库输入一个父项。 不过，您可以为每个数据库定义多个父项。 为此，请修改 seos.ini 配置文件的 parent_pmd 标记。 有关使用此标记的详细信息，请参阅*《参考指南》*。

UID/GID 同步

作为管理员，您可能会接收到按 UID 表明用户和按 GID 表明组的消息。必须确保 UID 和 GID 在任何地方的意义都相同。

默认情况下，PMDB 尝试在每个地方都对新的用户和组使用相同的 UID 和 GID，但您可以帮助从一开始提供必需的条件。 以相同的密码文件和相同的组文件开始，请确保 pmd.ini 文件中的 synch_uid 标记设置为 yes。 如果您的本地数据库是 PMDB 的订户，而且该 PMDB 是订户数据库的新用户和新组的唯一来源，那么您可以依赖本地数据库、PMDB 和 PMDB 订户的 UID 之间以及 GID 之间的兼容性。

如果您新建用户的 UID 已经在 PMDB 或某个其他订户计算机中使用，则该订户的单独更新将失败，但在没有这种冲突的其他所有订户计算机上，更新会成功。

同步 passwd 和组文件的另一种方法是显式指定每个新用户的 UID 以及每个新组的 GID。

同步用户和组

为了确保各种数据库中的用户和组的列表始终能正确对应，您需要相同列表的初始设置。 由于密码文件和组文件非常重要，因此应在它们开始收集本地用户和组信息之前，先将它们同步。

同步用户和组

1. 将您的 /etc/passwd file and /etc/group 文件复制到策略模型目录中。

   这是一次性的过程，它会破坏策略模型目录 (p. 123)中任何以前的密码文件和组文件。

   注意：如果您使用 shadow 文件并需要同步密码，建议您使用 secrepsw 实用程序。 有关详细信息，请参阅 *Reference Guide*。

2. 将 /etc/passwd 文件和 /etc/group 文件复制到每个订户计算机中，这样一来，它们与您自己计算机上的对应文件相同。

3. 在 PMDB 所在的计算机上，请确保将 pmd.ini 文件中的 synch_uid 标记设置为 yes。

   默认情况下，标记 synch_uid 的值为 yes。 如果您需要某个订户数据库具有独立的默认 UID 和默认 GID（即，不必尝试与 PMDB 的 UID 和 GID 相匹配），可将 synch_uid 设置为 no。

显式指定 UID

将相同的 UID 或 GID 发送到 PMDB 及其所有订户的另一种方式是，在创建新用户时显示设置 UID 或 GID。

要显式指定 UID，请对每条 newusr 命令都使用 userid 或 groupid 参数。

示例：使用指定的 UID 创建新用户

如果您要将 1234 显式建立为 terry_jones 的 UID（并假设数据库中还没有其他数据库使用该 UID），请输入命令：

```
newusr terry_jones unix (userid(1234))
```

如果指定的 UID 已在 PMDB 中使用，则 PMDB 将不会自行更新，但该命令仍将传播到其他订户数据库。 在其他已经使用特定 UID 的数据库中，订户的单独更新将失败；但在没有这种冲突的数据库中，更新会成功。

策略模型更新订户的方式

更新订户时，策略模型执行以下操作：

1. 当向策略模型中添加订户名称，或者从中删除订户名称时，策略模型将试图对其进行完全限定。
2. PMDB 后台程序 sepmdd 试图在标记 _QD_timeout_ 定义的时间内更新订户数据库。
3. 如果超出最长等待时间，但该后台程序无法成功更新某个订户，则它会忽略该订户，并尝试更新列表中的其余订户。
4. 完成订户列表的首次扫描后，sepmdd 会接着执行第二次扫描，在这次扫描期间，它会尝试更新在第一次扫描期间未成功更新的订户。第二次扫描期间，它将尝试更新订阅者，直到连接系统调用超时（大约 90 秒）。

注意：在 seos.ini 文件和 pmd.ini 文件中可能都会找到标记 _QD_timeout_。 如果在这两个文件中都存在该标记，则 sepmdd 将使用 pmd.ini 文件中的值。

注意：每当 PMDB 将更新传播到订户的同时遇到错误时，sepmdd 后台程序都会在策略模型错误日志文件 (p. 141)中创建一个条目。 该文件（默认情况下为 ERROR_LOG）位于PMDB 目录 (p. 123)中。

更新策略模型数据库

在 PMDB 所在计算机上的操作不会自动更新 PMDB 本身。 要更新 PMDB，需将其指定为目标数据库。

要指定目标数据库，请使用 selang 命令 shell 中的 hosts 命令：

```
hosts pmd_name@pmd_host
```

所有 selang 命令立即更新指定的策略模型数据库。 这些命令然后自动传播到此计算机和所有订户计算机上的活动数据库中。

示例：指定目标 PMDB

要将目标数据库设置为 myPMD_host 上的 policy1，请使用以下命令：

```
hosts policy1@myPMD_host
```

如果您现在输入 newusr 命令，则新用户将被添加到 policy1 数据库以及此计算机和所有订户计算机上的活动数据库中。

清理更新文件

sepmd 实用程序将自动写入它在 updates.dat 文件中接收到的每项更新。 为防止该文件变得过大，建议您定期删除文件中已处理的更新。

要清理更新文件，请使用以下命令：

```
sepmd -t pmdbName auto
```

sepmd 计算尚未传播的第一个更新条目的偏移量，并删除在它之前的所有更新条目。

注意：有关 sepmd 实用程序的详细信息，请参阅*《参考指南》*。

加密更新文件

在创建 PMDB 后、启动 sepmdd *前*，您可以指定对保存到 updates.dat 文件的信息进行加密。

要加密该更新文件，请将 pmd.ini 文件 [pmd] 部分中的 UseEncryption 标记设置为 yes。

要解密 updates.dat 文件，请使用带有 -de 开关参数的 sepmd 实用程序。

注意：有关 sepmd 的详细信息，请参阅*《参考指南》*。

排除订户

您可以忽略订户以使他们不接收来自父 PMDB 的更新。

要排除本地主机，请在 pmd.ini 文件中将标记 exclude_localhost 设置为 yes。

要向排除订户列表中添加其他订户，请设置标记 exclude_file（*文件的名称*）。

要让某个订户接收更新，请从已排除列表中删除该订户。

传播密码

当用户使用 sepass 实用程序更改密码时，新密码通常被发送到计算机的父 PMDB。 父 PMDB 在 seos.ini 文件 [seos] 部分的 parent_pmd 和/或 passwd_pmd 标记中定义。 然而，如果用户使用 sepass 实用程序更改密码，那么您还可以指定将该用户的新密码发送到单独的 PMDB，并由该 PMDB 传播。

要将新用户的密码发送到单独的 PMDB，请在 newusr、chusr 或 editusr 命令中使用 pmdb 参数。

示例：为密码传播指定单独的 PMDB

要指定将使用 sepass 创建的用户 Tony 的新密码发送到单独的 PMDB，并由该 PMDB 传播，请输入以下命令：

```
editusr tony pmdb(pw_pmdb@name1.yourorg.com)
```

删除订户

如果您不再希望将更新传播给某个订户，则应当将其删除。 也可以将订户排除，使其不接收更新 (p. 138)。

删除订户

1. 将计算机从订阅列表中删除：

   ```
   sepmd -u PMDB_name computer_name
   ```

   从策略模型订阅列表中删除计算机。

2. 关闭在您从订阅列表中删除的计算机上的 seosd：

   ```
   secons -s
   ```

   后台程序 seosd 被关闭。

3. 在您从订阅列表中删除的计算机上，删除在 seos.ini 文件的 [seos] 部分中 parent_pmd 标记的值。

   计算机将停止接受来自 PMDB 的更新。

4. 重新启动 seosd。

   您从订阅列表中删除的计算机上的活动数据库不再是指定 PMDB 的订户。

注意：从 PMDB 取消订阅数据库后，PMDB 不会再发送命令。

筛选更新

如果希望 PMDB 更新不同订户数据库上的不同数据子集，您需要定义向订户数据库发送哪些记录。

筛选更新

1. 配置 PMDB 以便用作订户子集的父项 (p. 134)。
2. 修改父 PMDB 的 pmd.ini 文件中的 *filter* 标记，以指向您在同一台计算机上设置的筛选文件。

   然后将对订户数据库的更新限于通过该筛选器的记录。

注意：当您在本地 UNIX 环境执行 join 或 join- selang 命令时，CA Access Control 会将命令更改为 change group (cg)。 要在本地 UNIX 环境中筛选 join 或 join- 命令，请在筛选文件中加入以下行：

MODIFY UNIX GROUP *GroupName* USERS NOPASS

无法在本地 UNIX 环境中按用户名筛选 join 或 join- 命令。 此规则在其他任何环境中均不适用于 join 或 join- 命令。

策略模型筛选器文件

筛选器文件由每行具有六个字段的行组成。 字段包含如下信息：

- 允许或禁用的形式。

  例如，READ 或 MODIFY

- 受影响的环境：

  例如，AC 或本地环境

- 记录的类。

  例如，USER 或 TERMINAL

- 规则涵盖的类中的对象。

  例如：User1、AuditGroup 或 TTY1

- 记录授予或取消的属性。

  例如，筛选行中的 OWNER 和 FULL_NAME 意味着具有这些属性的任何命令都会被筛选。 必须按照*《参考指南》*所述准确输入每个属性。
- 这类记录是否应该转发到订户数据库：

  PASS 或 NOPASS

以下规则适用于筛选器文件中的每一行：

- 可以使用星号 (*) 表示任意字段中的所有可能值。
- 如果有多行含有相同的记录，则使用适用的*第一行*。
- 用空格分隔字段。
- 在具有多个值的字段中，使用分号分隔值。
- 以 # 开头的行被视为注释行。
- 不允许有空行。

示例：筛选文件

以下示例介绍筛选器文件中的行：

CREATE	AC	USER	*	FULL_NAME;OBJ_TYPE	NOPASS
访问形式	环境	类	记录名（* =全部）	属性	处理

在此示例中，如果我们将具有该行的文件命名为 TTY1_FILTER 并编辑 PMDB TTY1 的 pmd.ini 文件，以使筛选为 /opt/CA/AccessControl//TTY1_FILTER，则 PMDB TTY1 不会向其订户传播使用 FULL_NAME 和 OBJ_TYPE 属性创建新用户的任何记录。

策略模型错误日志文件

按时间先后顺序组织的策略模型错误日志看上去与以下内容类似：

错误文本	错误类别
20 Nov 03 11:56:07 (pmdb1): fargo nu u5 0 Retry ERROR: 登录过程失败 (10068) ERROR: 无法接受来自非父项 PMDB 的更新 (pmdb1@name.company.com) (10104)	配置错误
20 Nov 03 19:53:17 (pmdb1): fargo nu u5 0 Retry ERROR: 连接失败 (10071) 主机不可访问 (12296)	连接错误
20 Nov 03 11:57:06 (pmdb1): fargo nu u5 560 Cont ERROR: 创建 USER u5 失败 (10028) 已经存在 (-9) 20 Nov 03 11:57:06 (pmdb1): fargo nu u5 1120 Cont ERROR: 创建 USER u5 失败 (10028) 已经存在 (-9)	数据库更新错误

策略模型错误日志采用二进制格式，您只有通过输入以下命令才能查看它：

ACInstallDir/bin sepmd -e *pmdname*

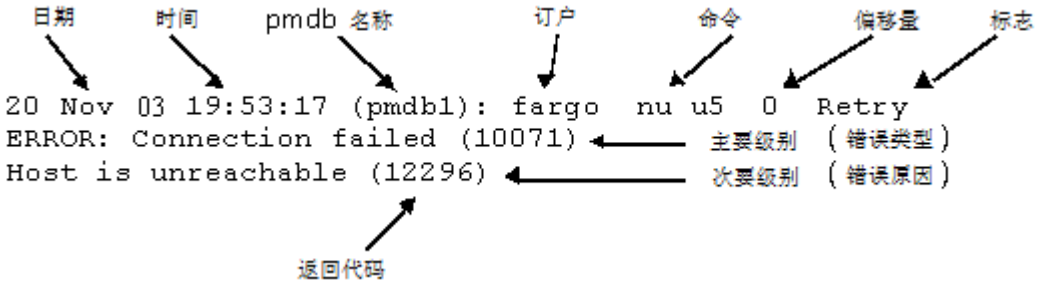
注意：不要手动删除错误日志（例如，使用 UNIX rm 命令）。只能使用以下命令删除日志：

ACInstallDir/bin sepmd -c *pmdname*

重要说明！ 在 CA Access Control r5.1 及更高版本中，错误日志的格式与早期版本的格式不兼容。sepmd 无法处理早期版本的错误日志。当您升级到具有该格式的版本时，旧的错误日志将复制到 ERROR_LOG.bak 中；在您启动 sepmdd 时将创建新的日志文件。

示例：PMDB 更新错误消息

以下示例显示典型的错误消息：



- 第一行总是由日期、时间和订户组成。 接下来显示产生错误的命令，然后是偏移量（十进制格式），指示更新文件中失败更新的位置。 最后，标志指示 PMDB 是自动重试更新，还是忽略该更新而继续。
- 第二行显示主要级别消息（发生的错误类型）的示例及消息的返回代码。
- 第三行显示次要级别消息（发生错误的原因）示例及消息的返回代码。

示例：错误消息

一个命令可能会产生并显示多个错误。 而且，一个错误可能包括主要级别消息、次要级别消息或同时包括这两种消息。

下列错误只有一个消息级别：

```
Fri Dec 29 10:30:43 2003 CIMV_PROD: 发布失败。 返回代码 = 9241
```

sepmd pull 尝试释放可用的订户时出现该消息。

策略模型备份

当备份 PMDB 时，您将策略模型数据库的数据复制到其他目录。 其中包括：

- 策略信息
- 策略模型的订户列表
- 配置设置
- 注册表项
- updates.dat 文件

您不能从使用其他平台、操作系统或 CA Access Control 版本的备份文件还原 PMDB。 确保将策略模型备份到运行相同的平台、操作系统和 CA Access Control 版本的主机上。

使用 sepmd 备份 PMDB

备份 PMDB 时，数据将从策略模型数据库复制到指定的目录。 您应将备份的 PMDB 文件存储至一个安全的位置，最好是受 CA Access Control 访问规则保护的位置。

您可以使用 sepmd 实用程序在本地主机上备份 PMDB。 您还可以使用 selang 命令在远程主机上备份 PMDB。

注意：您可以采用递归方式备份 PMDB。 递归备份可将一个层级结构中的所有 PMDB 备份到您指定的主机并修改 PMDB 订户，从而当备份移到该主机时订阅仍可进行。 当主 PMDB 和子 PMDB 部署在同一主机上时，您仅能使用递归备份。

使用 sepmd 备份 PMDB

1. 使用以下命令锁定 PMDB：

   `sepmd -bl pmdb_name`

   PMDB 将锁定，并且无法向其订户发送任何命令。

2. 请执行下列操作之一：

   - 使用以下命令备份 PMDB：

     `sepmd -bh pmdb_name [destination_directory]`

   - 使用以下命令采用递归方式备份 PMDB：

     `sepmd -bh pmdb_name [destination_directory] [backup_host_name]`

   注意：如果您不指定目标目录，备份将被存入以下目录：

   ACInstallDir/data/policies_backup/pmdb_name

3. 使用以下命令解锁 PMDB：

   `sepmd -ul pmdb_name`

   PMDB 将解锁，并且可向其订户发送命令。

使用 selang 备份 PMDB

备份 PMDB 时，数据将从策略模型数据库复制到指定的目录。 您应将备份的 PMDB 文件存储至一个安全的位置，最好是受 CA Access Control 访问规则保护的位置。

您可以使用 selang 命令在本地或远程主机上备份 PMDB。 您还可以使用 sepmd 实用程序在本地主机上备份 PMDB。

注意：您可以采用递归方式备份 PMDB。 递归备份可将一个层级结构中的所有 PMDB 备份到您指定的主机并修改 PMDB 订户，从而当备份移到该主机时订阅仍可进行。 当主 PMDB 和子 PMDB 部署在同一主机上时，您仅能使用递归备份。

使用 selang 备份 PMDB

1. （可选）如果要使用 selang 从远程主机连接 PMDB，请使用以下命令连接 PMDB 主机：

   ```
   host pmdb_host_name
   ```

2. 使用以下命令移至 PMD 环境：

   ```
   env pmd
   ```

3. 使用以下命令锁定 DMS：

   ```
   pmd pmdb_name lock
   ```

   PMDB 将锁定，并且无法向其订户发送任何命令。

4. 使用以下命令备份 DMS 数据库：

   ```
   backuppmd pmdb_name [destination(destination_directory)]
   [hir_host(host_name)]
   ```

   注意：如果您不指定目标目录，备份将被存入以下目录：

   ACInstallDir/data/policies_backup/pmdbName

5. 使用以下命令解锁 PMDB：

   ```
   pmd pmdb_name unlock
   ```

   PMDB 将解锁，并且可向其订户发送命令。

策略模型还原

当还原策略模型时，CA Access Control 会将 PMDB 备份文件复制到指定的目录。 原始 PMDB 文件中的所有内容都被复制到新的 PMDB 目录中，包括：

- 策略信息
- 策略模型的订户列表
- 配置设置
- 注册表项
- updates.dat 文件

如果目标目录中存在现有的 PMBD，CA Access Control 会在将还原文件复制到该目录之前删除现有文件。

您不能从使用其他平台、操作系统或 CA Access Control 版本的备份文件还原 PMDB。 确保将策略模型备份到运行相同的平台、操作系统和 CA Access Control 版本的主机上。

还原 PMDB

当您还原 PMDB 时，CA Access Control 将 PMDB 备份文件中的数据复制到您指定的目录里。 CA Access Control 必须在您执行还原的终端上运行。

注意：如果您在不同的终端上备份和还原 PMDB, PMDB 将不会在还原的 PMDB 数据库中自动更新终端资源。 您必须将新的终端资源添加到还原的 PMDB 中。 要添加新的终端资源，请停止还原的 PMDB，运行 *selang -p pmdb* 命令，然后再启动还原的 PMDB。

要还原 PMDB，请在想要还原 PMDB 的终端上运行以下内容之一：

- sepmd -restore 实用工具
- selang restore pmd 命令

注意：有关 sepmd 实用程序的详细信息，请参阅 *《参考指南》*。 有关 selang 命令的详细信息，请参阅 *《selang 参考指南》*。

双重控制

双重控制是一种操作方式，它将更新 PMDB 的过程分成两个阶段：

- 创建由一个或多个命令组成的事务。

  制定者（任何具有 ADMIN 属性的用户）输入更新 PMDB 的一个或多个命令。 为事务指定一个唯一的 ID 号，并将其置于一个文件中，等待在执行前被处理。

- 授权执行事务。

  检查者（不是同一用户，而是具有 ADMIN 属性的任何*其他*用户）锁定事务中的命令、检查命令并授权或拒绝这些命令。 如果授权事务，则将在 PMDB 中执行这些命令。 如果拒绝事务，则将删除事务，并且不更新 PMDB。 检查者无法授权事务中的某些命令而拒绝其他命令；必须将事务作为一个整体进行处理。

  注意：只有 find 和 show 命令不需要检查者的授权。

使用 sepmd 实用程序中的参数，制定者可以列出、检索和编辑或者删除未处理的事务；而检查者可以锁定事务，以便授权或拒绝它们，并且可以解除事务锁定，以便在稍后进行处理或者由不同的检查者处理。

当 sepmdd 后台程序收到 start_transaction 命令时，该程序会向子进程发送一个唯一的编号。 该子进程还使用此标识号来标记任何其他命令，该编号将添加到新事务中，并保留在 sepmdd 后台程序的内存中。 当 sepmdd 收到 end_transaction 命令时，将调用授权算法。 授权算法将检查事务中没有与事务制定者相关的命令，以及该命令中没有已被正在等待执行前处理的另一事务锁定的对象。

对于不同事务中的相同对象，您只有处理事务后才能进行使用。 如果检查通过，则会锁定关联对象，为事务分配唯一的序列号，并将数据保存到文件中。 每个事务都保存在不同的文件中。

注意：有关 sepmd 实用程序或 sepmdd 后台程序的详细信息，请参阅*《参考指南》*。

激活双重控制

双重控制允许您在两类用户之间分配更新 PMDB 的责任：制定者和检查者。

要激活双重控制，请将 *pmd.ini* 文件和 seos.ini 文件的 [pmd] 部分中的 is_maker_checker 内标识设置为 yes：

```
is_maker_checker=yes
```

注意：在设置这些标记值*之前*，请先创建策略模型制定者。

创建或编辑事务

激活双重控制时，制定者需要在检查者处理之前创建事务。

创建事务

1. 请确保以下内容为真：
   - 您（作为制定者）有 ADMIN 权限。
   - 没有与您相关的命令。（您无法输入更改您自己的命令。）
   - 命令中没有对象已经属于检查者尚未处理的另一个事务。
   - 命令中的所有对象都存在。
   - 您未在编辑另一制定者已调用的现有事务。（您只能编辑自己的事务。）
2. 连接到制定者 PMDB：

   ```
   hosts maker@
   ```

   主机命令将您连接到 PMDB（制定者）。 激活双重控制后，PMDB 的名称始终是“制定者”。当您输入主机命令后，将显示消息，报告是否成功连接到主机。
3. 启动事务：

   ```
   start_transaction transactionName
   ```

   输入和更新事务时，第一步是使用 start_transaction 命令。 您可以说明事务，也可以为事务指定您所需的名称，最多 256 个字母数字字符。
4. 输入您的事务。

   这是一组命令。 例如：

   ```
   newusr mary owner(bob) audit(failure,loginfailure)
   chres TERMINAL tty30 defaccess(read) \
   restrictions(days(weekdays) time(0800:1800))
   ```
5. 结束事务：

   ```
   end_transaction
   ```

   事务完成。系统将显示分配给事务的唯一 ID 号。 命令置于一个文件中，在检查者为准备处理而将命令锁定之前，您仍然可以访问和更改这些命令。

   注意：如果您希望以后能够编辑事务，请确保记录事务 ID 号。

编辑事务

- 在您输入 end_transaction 命令时将显示一个 ID 号。 这是识别事务的唯一编号。 如果您需要在以后覆盖事务，则除了向文件添加事务名称后的 ID 号以外，该过程与创建新事务相同。 您可以在文件中输入您所要进行的任何更改。 例如：

  ```
  hosts maker@
  start_transaction transactionName transactionId
  ```

  然后，您可以输入适当的命令来更新事务：

  ```
  chusr mary category (FINANCIAL)
  end_transaction
  ```

- 使用以下参数查看未处理的特定事务。

  请确保您位于 *ACInstallDir/bin* 路径中（其中，*ACInstallDir* 是 CA Access Control 的安装目录，默认情况下为 /opt/CA/AccessControl/）。

带参数的命令	说明
sepmd -m l	列出调用参数的用户未处理的事务。
sepmd -m la	列出所有制定者正在等待处理的所有事务。
sepmd -m lo	列出除了调用参数的用户事务以外的所有制定者的事务。 列表中的每个事务都包括制定者名称、事务的 ID 号以及事务的说明（如果制定者输入了事务）。

- 使用以下命令，检索特定于标准输出的事务：

  ```
  sepmd -m r transactionId
  ```

- 使用该命令删除特定的事务：

  ```
  sepmd -m d transactionId
  ```

检查和处理事务

激活双重控制时，检查者需要处理由制定者创建的事务。

检查事务

1. 请确保以下内容为真：
   - 您（作为检查者）有 ADMIN 权限。
   - 其他检查者没有锁定事务。
   - 没有与您相关的命令。（您无法处理调用您自己的命令。）

2. 浏览至 *ACInstallDir*/bin 路径

   其中 *ACInstallDir* 是 CA Access Control 的安装目录，默认情况下为 /opt/CA/AccessControl/

3. 查看在执行前正在等待处理的事务：

   ```
   sepmd -m la
   ```

   或者查看除了您自己创建的事务之外的所有事务：

   ```
   sepmd -m lo
   ```

   每个事务都包括制定者名称、事务的 ID 号以及事务的名称或说明。

4. 在处理它们之前将其锁定：

   ```
   sepmd -m r transactionId
   ```

   注意：已锁定的事务无法更改。

5. 处理事务：

   ```
   sepmd -m p transactionId code
   ```

   code

   可以是以下各项之一：

   - **0** - 拒绝事务。

     在这种情况下，将删除事务中的所有命令，并且不会在 PMDB 中实施更改。

   - **1** - 授权事务。

     在 PMDB 中立即实施事务中的命令。

   - **2** - 事务解除锁定。

     事务返回等待事务的队列，并可以在以后由不同的检查者执行。

   将出现一条说明哪些命令成功和哪些命令失败的消息。

注意：有关制定者和检查者的详细信息，请参阅*《参考指南》*中的 sepmd 实用程序以及 *《selang 参考指南》* 中的 start_transaction 命令。

使用 seagent 和 sepmdd 后台程序

seagent 后台程序负责接受来自远程计算机的请求，并将其应用到 PMDB；seagent 后台程序还会将请求发送到 seosd。 sepmdd 后台程序是 PMDB 后台程序。 本节介绍在 PMDB 环境中这些后台程序如何一起工作。

seagent 后台程序

seagent 后台程序等待连接 seoslang 和 seoslang2 TCP 服务（默认值分别为 8890 和 8891）。 当连接请求到达时，seagent 再生一个子进程来处理连接的通信，然后继续等待新连接。

当用户在 *selang* 中输入主机命令时，seagent 在用户连接到的计算机上再生一个子进程。 然后，该子进程接收来自命令语言接口的命令，并将其传递给 sepmdd 后台程序。

sepmdd 后台程序

sepmdd 后台程序执行以下功能：

- 管理 PMDB
- 管理订户数据库
- 将更改从 PMDB 传播到订户数据库

sepmdd 后台程序在 seagent 必须访问 PMDB 时由 seagent 自动启动。 通常不需要显式运行 sepmdd。

注意：sepmdd 在 AC 环境中在逻辑用户 _seagent（*不*在 root 用户下）下运行。 要允许或限制 sepmdd 访问资源（例如，限制访问 PMDB 目录），请创建 _seagent 的相关规则。

使用卷影文件

通常，sepmdd 在更新本地环境时不使用 shadow 文件。 不过，您可以设置一个 shadow 文件。 要执行此操作，请将 pmd.ini 文件的 [pmd] 部分中的 UseShadow 标记设置为 yes。

如果将 UseShadow 标记设置为 yes，则 sepmdd 将使用与 PMDB 处于相同目录中的默认 shadow 文件。 如果要更改 shadow 文件的位置，请使用 pmd.ini 文件的 [pmd] 部分中的 YpServerSecure 标记指定新位置。

如果（使用 YpServerSecure）将卷影文件的位置更改为本地主机的卷影文件（例如，/etc/shadow），sepmdd 则会将标记 UseSystemFiles 设置为 yes。

重要说明！ 不要自行更改 UseSystemFiles 标记。 sepmdd 或 seagent 后台程序会自动将其更改。

注意：有关 seagent 或 sepmdd 后台程序的详细信息，请参阅 *《参考指南》*中的 seagent 和 sepmdd 实用程序。

大型机密码同步

CA Access Control 支持运行 CA Top Secret、CA ACF2 或 RACF 安全产品（和 CA Common Services CAICCI 程序包）的大型机与运行 CA Access Control 的 Windows 或 UNIX 计算机之间的密码同步。 同步是使用标准 CA Access Control 密码策略模型方法完成的。

大型机用户执行的任何密码更改都将传播至该密码策略模型层级结构中的所有计算机。

第 12 章： 一般安全功能

此部分包含以下主题：

保护空闲工作站

当终端处于打开和活动状态时，信息非常容易遭到破坏。 由于站点上的所有终端都已登录并准备工作，因此，如果入侵者此时恰巧在这样一台终端（例如，在午休时间）上，则无需破解密码或使用复杂的设备即可窃听网络通信。 尽管在还原到桌面之前提示输入密码的屏幕保护程序非常有用，但安全管理员无法确保所有用户都在使用安全的屏幕保护程序。

CA Access Control 提供一种屏幕锁定实用工具 selock，该实用工具通过每当终端和工作站空闲超过指定时间时即将其锁定来对其进行保护。 当用户返回工作时，系统提示用户指定密码。 如果未在一分钟内指定正确的密码，则终端保持锁定状态。 selock 实用工具可以查找取消锁定屏幕的用户的密码，即使那些用户在 selock 处于活动状态时更改他们的密码也是如此。

注意：有关屏幕锁定实用程序 selock 的详细信息，请参阅*《参考指南》*。

您应该选择使用适合您要求的 selock 选项：

- 安全性较低，但更方便

  使用 -timeout 选项将超时设置为较大值，如 10 分钟，然后使用 -lock-timeout 选项将锁定超时设置为更大的值，如 60 分钟。 这样，可通过切换到*保护程序*模式防止 selock 过多地打断您的工作。 该设置还会仅在终端长时间内不活动时锁定屏幕。

- 安全性较高，但更不方便

  使用 -timeout 选项将超时设置为较小值，如 1 分钟，然后使用 -lock-timeout 选项将锁定超时设置为介于 0 到 2 分钟之间的较小值。 这可以始终在您停止访问终端后立即隐藏您的工作，并且需要有密码才能还原访问。 要确保在保护程序模式启动后 selock 总是需要密码条目来激活您的终端，请使用 -lock-timeout 选项将锁定超时设置为零。

- selock 命令可以是 X 启动 shell 的一部分，这样，该命令便可以在用户每次登录系统时自动启动。 该脚本必须在用户 ID 下而不是 root ID 下运行。 将 selock 命令集成到启动脚本的方式取决于站点的特定环境。

  注意：有关启动脚本的详细信息，请参阅 UNIX 系统的文档。

保护模式

selock 提供三种操作模式：

监控模式

监控模式是 selock 的初始模式。在这种模式下，selock 监控键盘和鼠标活动。 如果 selock 检测到在超时时间内没有键盘或鼠标活动，而且 transparent 参数是关闭状态，则 selock 自动切换到屏幕保护程序模式。 从监控模式切换到保护程序模式不需要输入密码。

保护程序模式

在保护程序模式下，selock 使整个屏幕空白并显示位置会改变的系统图标。 空白屏幕和位置会改变的图标具有两个操作优势：

- 降低了未经授权人员查看屏幕内容的风险
- 减少了屏幕额外曝光的情况

您可以使用 selock 选项控制图标的外观和调整其位置。 当 selock 检测到任何键盘或鼠标活动时，它会立即从保护程序模式返回到监控模式，将屏幕显示还原为它切换到保护程序模式之前的状态。 从监控模式切换到保护程序模式不需要输入密码。

如果 selock 在 lock-timeout 参数指定的时间内一直保持保护程序模式，它将自动切换到锁定模式。selock 不会显示有关从保护程序模式切换到锁定模式的任何指示。

锁定模式

在使用默认设置的锁定模式下，selock 会继续在黑色背景上显示移动的图标。 当 selock 检测到任何键盘或鼠标活动时，会显示一个对话框，其中包含要求输入用户密码的提示。

当用户输入的密码正确时，selock 会切换回监控模式。如果用户输入的密码不正确，密码条目对话框会关闭，而 selock 仍保持锁定模式。

如果您将 -transparent 选项设为 *on*，selock 将锁定屏幕，但会显示内容并更新正在进行的进程。 屏幕的背景会更改，以指示屏幕已锁定。使用锁定模式时，不会调用保护模式。

将工作站设置为在空闲时锁定

通过 selock 实用程序，您可以锁定空闲工作站，以防止在工作站空闲期间受到未经授权的访问。

将工作站设置为在空闲时锁定

1. （可选）设置 DISPLAY 环境变量。

   要使 selock 命令工作，您必须设置 DISPLAY 环境变量。 不过，您可以改用 selock 命令直接指定目标显示。

2. 将 selock 命令置于用户的登录脚本（.login 文件）中。

   您也可以将 selock 命令置于 /etc/login 或 /etc/cshrc 文件中。

注意：两类用户始终可以对锁定的屏幕取消锁定。 默认情况下，这两类用户是当前用户和 root 用户。 但是，如果您在 unlocking_user 标记（位于 seos.ini 文件的 [selock] 部分）中指定了任何其他用户的名称，则可以用该用户替换 root 用户。 执行 selock 时，可以使用 -user 选项将当前用户替换为任何其他用户。

示例：启动文件中的空闲工作站锁定命令

下面是典型的启动命令，适合置入 X 启动文件中：

selock -display $DISPLAY -timeout 5

该命令在终端处于停怠状态五分钟后激活 selock。

建议您将下面一行置于全局 xstartup 脚本中。 xstartup 脚本通常位于目录 /usr/lib/X11/xdm/Xstartup 中。

```
selock -display $DISPLAY -user $USER -timeout 3 &
```

该语句强制正在使用 X 终端的所有用户使用终端锁定程序。

更改屏幕锁定图标

selock 使用的默认系统图标是 CA Access Control 徽标，该图标位于文件 ACInstallDir/data/admin/Selogo.xpm 中。

要选择您喜欢的图标，请替换该文件。

注意：图标文件必须为 XPM 版本 3.3 格式。

用 API 保护资源

如果您定义的资源不属于 CA Access Control（即内部资源），您可以通过使用 CA Access Control API 对其进行保护。 每个 API 都有两层：

函数库

使编程人员可使用 CA Access Control 授权引擎。

用户出口

使系统管理员可以定制满足站点要求的 CA Access Control 行为。

注意：有关 CA Access Control API 的详细信息，请参阅 *《SDK 指南》*。

防止堆栈溢出：STOP

堆栈溢出使黑客可以在远程或本地系统上像 root 用户（超级用户）一样多次地执行任意命令。 他们利用操作系统或其他系统中的缺陷来进行该操作。 这些缺陷使用户覆盖程序堆栈，更改要执行的下一个命令。

堆栈溢出不仅是一个错误，它还可能会创建一个块，使用有意义的地址来覆盖返回地址，从而对未经授权的代码进行传输控制（通常在同一个块中）。

堆栈溢出保护 (STOP) 是防止黑客创建和利用堆栈溢出进入系统的一种功能。

注意：强制使用 Linux 本地堆栈随机化（ExecShield 随机化）时，不会激活 Red Hat Linux 和 SuSE Linux 上的 STOP 功能。

在 Linux s390 RHEL 4 上，本地堆栈随机化无法工作，并且必须停用它后 STOP 才能变为活动状态。 要停用本地堆栈随机化，请输入以下命令：

```
echo 0 > /proc/sys/kernel/exec-shield-randomize
```

启动和停止 STOP

首次安装 STOP 时，在默认情况下将激活堆栈溢出保护。 要取消激活堆栈溢出保护，您必须更改 seos.ini 文件 [seos_syscall] 部分中的标记，然后重新启动 CA Access Control。 要执行该操作，请使用以下 seini 命令：

```
seini -s SEOS_syscall.STOP_enabled 0
```

您也可以手动更改 seos.ini 文件。

要重新启用 STOP，请将该标记的值更改为 1，并重新启动 CA Access Control 。

注意：当 STOP 在 Sun Solaris 7 系统上处于活动状态时，dbx 程序无法正常工作。 如果需要在受 STOP 保护的系统上使用 dbx，则必须先禁用 STOP。

定义资源的日期和时间访问规则

您可以使用 CA Access Control 为资源访问指定“工作日”和“工作时间”限制。 该功能可用于 TERMINAL 访问、SURROGATE 请求和用户定义的资源。 例如，下列规则在周末和每天 08:00-19:00 时间段以外的时间内完全禁用 terminal ws3:

```
chres TERMINAL ws3 restrictions(days(weekdays) time(0800:1900))
```

在这些时间段外不接受该工作站的登录请求。

您可以使用 CA Access Control 来防止在工作时间外替换为高度授权的用户的请求。 假如用户 AcctMgr 是允许执行财务事务的财务经理，而您已经限制 AcctMgr 只在工作时间和工作日登录。 入侵者或未经授权的人员可能试图通过调用 **su** AcctMgr 命令来访问 AcctMgr 的帐户。 使用下列命令可禁止在指定时间外将用户名替换为 AcctMgr：

```
chres SURROGATE USER.AcctMgr restrictions(days(weekdays) time(0800:1900))
```

可以对受保护的任何资源实施同样的方法，包括用户-定义的用于实施内部-应用程序的抽象类。

B1 安全级别认证

CA Access Control 包括下列 B1“橙皮书”功能：

- 安全类别
- 安全标签
- 安全级别

安全级别

启用安全级别检查后，CA Access Control 将执行安全级别检查以及其他授权检查。 安全级别是可以分配给用户和资源的 1 到 255 之间的正整数。 用户请求访问分配了安全级别的资源时，CA Access Control 会将该资源的安全级别与该用户的安全级别进行比较。 如果该用户的安全级别等于或大于该资源的安全级别，则 CA Access Control 将继续进行其他授权检查；否则，将拒绝该用户访问该资源。

如果 SECLABEL 类处于活动状态，则 CA Access Control 将使用与资源和用户的安全标签关联的安全级别，而忽略在资源和用户记录中显式设置的安全级别。

要通过安全级别检查来保护资源，请为资源记录分配安全级别。 newres 或 chres 命令的级别参数为资源分配安全级别。

若要允许用户访问受到安全级别检查保护的资源，请为用户记录分配安全级别。 newusr 或 chusr 命令的级别参数为用户分配安全级别。

启用安全级别检查

下面的 setoptions 命令启用安全级别检查：

```
setoptions class+ (SECLEVEL)
```

禁用安全级别检查

下面的 setoptions 命令禁用安全级别检查：

```
setoptions class- (SECLEVEL)
```

安全类别

启用安全类别检查后，CA Access Control 将执行安全类别检查以及其他授权检查。 用户请求访问分配了一个或多个安全类别的资源时，CA Access Control 会将资源记录中的安全类别列表与用户记录中的类别列表进行比较。 如果分配给该资源的每个类别均显示在用户的类别列表中，则 CA Access Control 将继续进行其他授权检查；否则，将拒绝该用户访问该资源。

如果 SECLABEL 类处于活动状态，则 CA Access Control 将使用与资源和用户的安全标签关联的安全类别列表，而忽略用户和资源记录中的类别列表。

要通过安全类别检查来保护资源，请为资源记录分配一个或多个安全类别。 newres 或 chres 命令的类别参数为资源分配安全类别。

要允许用户访问受安全类别检查保护的资源，请为该用户的记录分配一个或多个安全类别。 newusr 或 chusr 命令的类别参数为用户分配安全类别。

启用安全类别检查

下列 setoptions 命令启用安全类别检查：

```
setoptions class+ (CATEGORY)
```

禁用安全类别检查

下列 setoptions 命令禁用安全类别检查：

```
setoptions class-(CATEGORY)
```

定义安全类别

通过定义 CATEGORY 类中的资源来定义安全类别。 下列 newres 命令定义安全类别：

```
newres CATEGORY name
```

其中，*name* 是安全类别的名称。

要定义安全类别“*Sales*”，请输入以下命令：

```
newres CATEGORY Sales
```

要定义安全类别“*Sales*”和“*Accounts*”，请输入以下命令：

```
newres CATEGORY (Sales,Accounts)
```

列出安全类别

要显示数据库中定义的所有安全类别的列表，请使用如下 show 命令：

```
find CATEGORY
```

屏幕上显示安全类别列表。

删除安全类别

要删除安全类别，请从 CATEGORY 类中删除安全类别的记录。 下列 rmres 命令删除安全类别：

```
rmres CATEGORY name
```

其中，*name* 是安全类别的名称。

要删除安全类别“*Sales*”，请输入以下命令：

```
rmres CATEGORY Sales
```

安全标签

安全标签代表特定安全级别与零个或多个安全类别之间的关联。

当启用安全标签检查时，CA Access Control 将执行安全标签检查以及其他授权检查。 用户请求访问分配了安全标签的资源时，CA Access Control 会将资源记录的安全标签中指定的安全类别列表与用户记录的安全标签中指定的安全类别列表进行比较。 如果为资源的安全标签分配的每个类别均显示在用户的安全标签中，则 CA Access Control 将继续进行安全级别检查；否则将拒绝用户访问该资源。 CA Access Control 会将资源记录的安全标签中指定的安全级别与用户记录的安全标签中指定的安全级别进行比较。 如果用户的安全标签中分配的安全级别大于或等于资源的安全标签中分配的安全级别，则 CA Access Control 将继续执行其他授权检查；否则，将拒绝该用户访问该资源。

当启用了安全标签检查时，将忽略用户和资源记录中指定的安全类别和安全级别；仅使用安全标签定义中指定的安全级别和安全类别。

要通过安全标签检查来保护资源，请为资源记录分配安全标签。 newres 或 chres 命令的标签参数为资源分配安全标签。

要允许用户访问受安全标签检查保护的资源，请为该用户的记录分配安全标签。 newusr 或 chusr 命令的标签参数为用户分配安全标签。

启用安全标签检查

下面的 setoptions 命令启用安全标签检查：

```
setoptions class+(SECLABEL)
```

禁用安全标签检查

下面的 setoptions 命令禁用安全标签检查：

```
setoptions class-(SECLABEL)
```

定义安全标签

通过定义 SECLABEL 类中的资源来定义安全标签。 以下 newres 命令定义安全标签：

```
newres SECLABEL name category(securityCategories) level(securityLevel)
```

其中：

名称

指定安全标签的名称。

securityCategories

指定安全类别的列表。 要指定多个安全类别，请使用空格或逗号来分隔安全类别的名称。

securityLevel

指定安全级别。 使用介于 1 至 255 之间的整数。

要定义包含安全类别 Sales 和 Accounts 且安全级别为 95 的安全标签管理器，请输入以下命令：

```
newres SECLABEL Manager category(Sales,Accounts) level(95)
```

列出安全标签

要显示数据库中定义的所有安全标签的列表，请使用如下 show 命令：

```
find SECLABEL
```

屏幕上出现安全标签列表。

删除安全标签

通过删除 SECLABEL 类中的安全标签记录来删除安全标签。 以下 rmres 命令删除安全标签：

```
rmres SECLABEL name
```

其中，*name* 是安全标签的名称。

要删除安全类别“*Manager*”，请输入下列命令：

```
rmres SECLABEL Manager
```

第 13 章： 审核事件

此部分包含以下主题：

设置审核规则

为了安全审核，CA Access Control 将根据在数据库中定义的审核规则，保留拒绝或授权访问事件的审核记录。

每个访问者和每种资源都具有 AUDIT 属性，可以设置为以下一个或多个值：

FAIL

记录访问者对资源的访问失败。

SUCCESS

记录访问者对资源的成功访问。

LOGINFAIL

记录访问者的每一次登录失败。（该值不适用于资源。）

LOGINSUCCESS

记录访问者的每一次成功登录。（该值不适用于资源。）

ALL

记录与访问者的 FAIL、SUCCESS、LOGINFAIL 和 LOGINSUCCESS 或资源的 FAIL 和 SUCCESS 相同的信息。

无

不记录任何有关访问者或资源的信息。

TRACE

记录的信息与使用 ALL 以及所有系统事件相同。（该值不适用于资源。）

无论您何时在数据库中创建或更新访问者或资源记录，都可以指定 AUDIT 属性。 此外，您还可以指定是否应该发送以及向何人发送记录事件的电子邮件通知。

审核日志中的记录将根据这些审核规则进行汇集。 是否记录某个事件的决策基于以下情况：

- 如果资源或访问者具有 AUDIT(ALL)，则将记录访问者的所有登录事件以及与 CA Access Control 所保护资源有关的所有事件，而无论访问失败还是成功。
- 如果对 CA Access Control 所保护资源的访问成功，且访问者或资源具有 AUDIT(SUCCESS)，则将记录该事件。
- 如果对 CA Access Control 所保护资源的访问失败，且访问者或资源具有 AUDIT(FAIL)，则将记录该事件。

此外，如果您将用户设置为可跟踪用户，该用户的每次跟踪记录被写入时，对应的审核记录也会被写入审核日志中。

定义 CA Access Control 写入审核日志的审核事件

CA Access Control 将成功和失败的访问记录写入到审核日志。 要定义 CA Access Control 将哪些访问事件写入到审核日志，请更改要审核的资源或访问者的 AUDIT 属性值。 您还可以使用该方法指定 CA Access Control 将每个跟踪事件记录到审核日志。

使用 AUDIT 属性来指定 CA Access Control 写入到审核日志的审核事件。 使用 selang 或 CA Access Control 端点管理 来为以下资源和访问者设置 AUDIT 属性：

AUDIT 值	CA Access Control 记录的内容	适用对象
FAIL	失败访问	用户和资源
SUCCESS	成功访问	用户和资源
LOGINFAIL	失败登录	用户
LOGINSUCCESS	成功登录	用户
ALL	等同于 FAIL、SUCCESS、LOGINFAIL、LOGINSUCCESS 和 INTERACTIVE	用户和资源
TRACE	等同于 ALL 和所有系统事件	用户

AUDIT 值	CA Access Control 记录的内容	适用对象
INTERACTIVE	UNIX 计算机上的用户会话	用户
无	无登录	用户和资源

注意：如果未设置用户的审核属性，组或配置文件组的 AUDIT 值则可以影响 CA Access Control 针对用户所使用的审核模式。

用户会话日志记录的工作原理

通过用户会话日志记录，您可以跟踪端点上的用户活动、重放会话和查看用户在该会话期间输入的命令。

会话日志记录器记录列在 /etc/shells 文件中所有程序的输入。 例如，如果 /etc/shells 中列有 /usr/bin/passwd 且您使用 passwd 来更改密码，那么 seaudit 实用工具会在您显示会话日志时显示更改的密码。 建议您在实施会话日志记录之前先查看 /etc/shells 文件。

以下过程说明了用户会话日志记录的工作原理：

1. 安装启用了“键盘记录器”选项的 CA Access Control。

   自定义 CA Access Control 参数文件来启用键盘记录器。

   注意：您可以在安装后在 seos.ini 文件中启用键盘记录器。

2. 启动 CA Access Control。

   确认键盘记录器后台进程 KBLAudMngr 正在运行。 使用 issec 实用工具查看 CA Access Control 后台进程的状态。

3. 将 INTERACTIVE 属性分配给您想要跟踪的用户以便启用会话日志记录。 例如：

   - selang:

     ```
     eu user1 audit(interactive)
     ```

   - CA Access Control 端点管理:

     在“用户属性”窗口中选中“审核”选项卡上的“交互”框。

   CA Access Control 启用了该用户帐号的会话日志记录。

4. 当用户登录端点时，CA Access Control 开始记录用户会话。 当用户注销该端点时，会话结束。
5. CA Access Control 将已纪录的会话保存在 kbl.audit 日志文件中。 该文件位于以下目录：

   ```
   /opt/CA/AccessControl/log
   ```

6. 将 seaudit 实用工具与 -kbl 命令结合使用可显示 kbl.audit 日志文件的内容。 例如：

   ```
   ./seaudit -kbl -sid 65223 -rp
   ```

   注意：有关 seaudit -kbl 命令的详细信息，请参阅*《参考指南》*。 建议您将 CA Access Control 端点与 CA Enterprise Log Manager 相集成，以便从企业中的主机收集用户会话并生成报告。 有关与 CA Enterprise Log Manager 集成的详细信息，请参阅*《实施指南》*。

CA Access Control 如何为用户确定审核模式

用户的审核模式指定 CA Access Control 将哪些审核事件发送至该用户的审核日志中。 以下过程说明 CA Access Control 如何确定用户的审核模式：

1. CA Access Control 检查 USER 或 XUSER 类中的用户记录是否有针对 AUDIT 属性的值。

   如果用户的记录有针对 AUDIT 属性的值，CA Access Control 则使用此值作为该用户的审核模式。

2. CA Access Control 检查是否将用户分配到配置文件组。 如果将用户分配到配置文件组，CA Access Control 则检查 GROUP 类中配置文件组的记录是否有针对 AUDIT 属性的值。

   如果将用户分配到配置文件组且配置文件组的记录有针对 AUDIT 属性的值，CA Access Control 则使用此值作为该用户的审核模式。

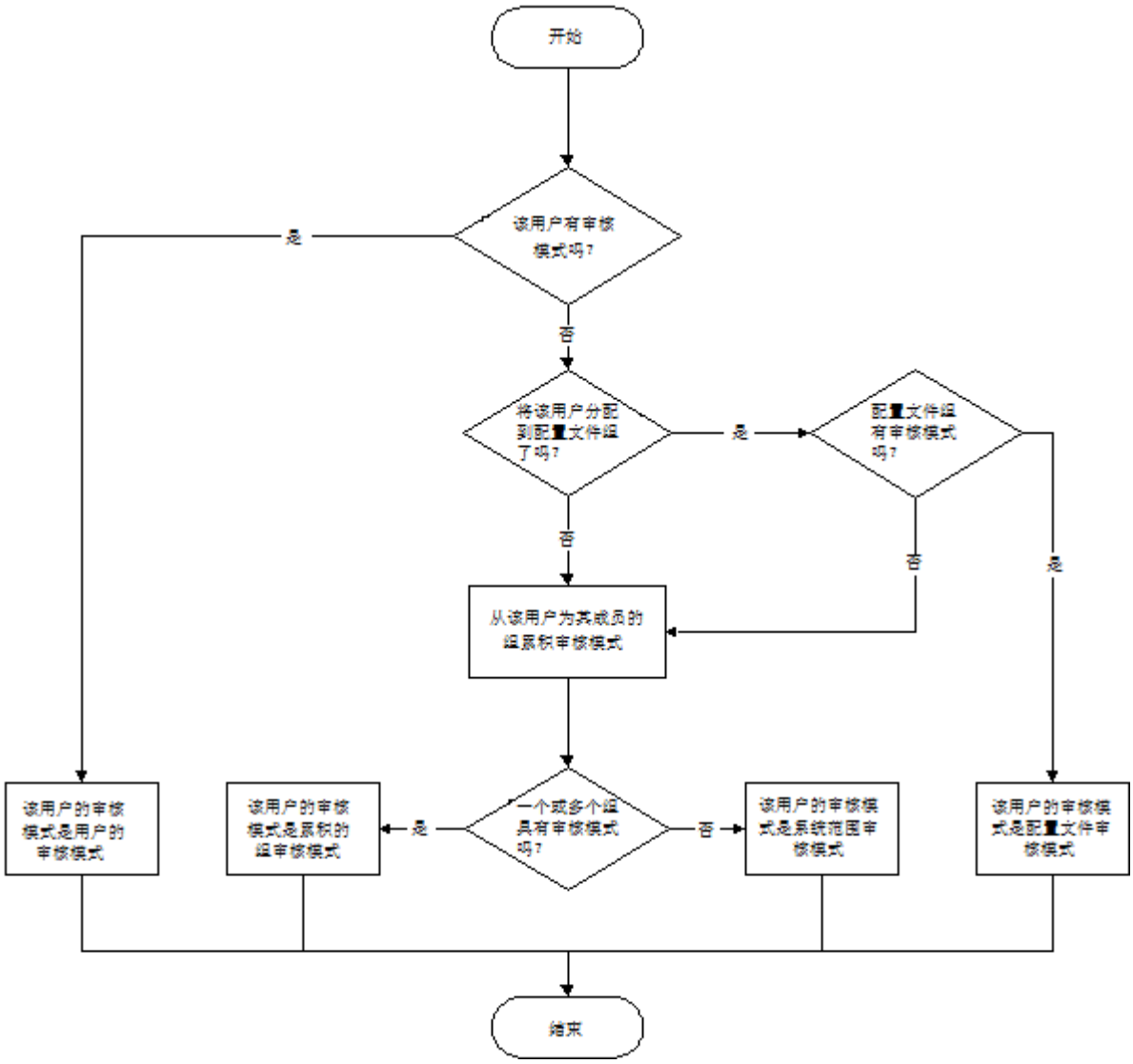
3. CA Access Control 检查用户是否为组中成员。 如果用户是组成员，CA Access Control 则检查 GROUP 或 XGROUP 类中组的记录是否有针对 AUDIT 属性的值。

   如果用户是组成员且组的记录有针对 AUDIT 属性的值，CA Access Control 则使用此值作为该用户的审核模式。 如果用户不是组成员或组的记录没有针对 AUDIT 属性的值，CA Access Control 则将系统范围审核模式分配给该用户。

   注意：如果用户是多个组中的成员且这些组有不同的审核模式，那么用户的审核模式会累积。 即该用户的审核模式是成员所在组的所有审核模式的总和。

注意：如果 CA Access Control 使用组的 AUDIT 属性值来确定用户的审核模式，而且在用户登录时更改了组的审核模式，那么已登录用户的审核模式也发生变化。 该用户不需注销使组审核模式中的更改生效。

以下图表显示 CA Access Control 如何确定用户的审核模式：



示例：按组审核

用户 Jan 同为组 A 和组 B 的成员。 组 A 有 FAIL 的审核模式，组 B 有 SUCCESS 的审核模式。 由于 Jan 是两个组的成员，因此 Jan 有 FAIL 和 SUCCESS 的累积审核模式。

更多信息：

CA Access Control 如何使用配置文件组确定用户属性 (p. 36)

用户和企业用户的默认审核模式

当您创建用户（USER 对象）时，CA Access Control 将默认 AUDIT_MODE 分配给该对象。 AUDIT_MODE 属性的默认值为 Failure、SuccessLogin、SuccessFailure。

当您创建企业用户（XUSER 对象）时，默认情况下，CA Access Control 不会将默认 AUDIT_MODE 值分配给该对象。

注意：(UNIX) 要针对 USER 对象更改 AUDIT_MODE 属性的默认值，请在 lang.ini 文件的 [newusr] 部分中编辑 DefaultAudit 值。

为某些用户更改为默认审核值

r12.0 SP1 CR1 之前，针对以下访问者默认审核模式为“无”：

- 在相应的 USER 类记录中没有定义的 AUDIT 值，没有与定义的 AUDIT 值的配置文件组相关联的用户。
- 没有在数据库（由 _undefined 用户记录表示）定义的任何用户。

  注意：如果使用企业用户，CA Access Control 则不会将任何用户看作未定义。 针对 _undefined 用户的属性在这种情况下不适用。

从 r12.0 SP1 CR1， 这些访问者的默认审核模式为 Failure、LoginSuccess 和 LoginFailure。 要获得早期行为，请将这些用户的 AUDIT 属性值设置为“无”。

对于 GROUP 记录，更改 AUDIT 属性的值

如果具有两个函数的 GROUP 记录：

- 为一组用户定义审核策略的配置文件
- 针对第二组用户的容器

r12.0 SP1 CR1 之前，GROUP 记录也为第二组用户定义审核策略。 为避免由于更改操作而可能引发的问题，请为第二组用户创建单独的 GROUP。

警告模式

*警告模式*是一个可应用到资源的属性、可应用到类的选项。 如果警告模式已应用到资源或类，且访问违反了访问规则，则 CA Access Control 会写入审核日志条目和返回代码 W，但允许访问资源。 如果类处于警告模式，则该类中的所有资源都处于警告模式。

仅在 CA Access Control 处于完全增强模式时警告模式才有效。

注意：完全强制模式是 CA Access Control for UNIX 所支持的唯一模式。 CA Access Control for Windows 还支持仅审核模式。

当您创建或修改访问策略时，可以使用警告模式。 如果使用警告模式，则在实施策略之前，您可以查看审核日志以预览该所需策略的结果。 您可以使用 seaudit 命令显示审核日志。

如果类具有属性 *warning*，则您可以将该类置于警告模式。 如果资源组或类处于警告模式，则当访问违反访问规则时，CA Access Control 将允许该访问，并在引用资源（非资源组或类）的审核日志中写入条目。

资源和类的警告模式设置是独立的：如果您将资源置于警告模式，即使它属于某个类并且您从该类中删除了警告模式，该资源仍将处于警告模式。

注意：如果资源或类具有属性 *warning*，则您可以将其置于警告模式；并非所有资源或类都有该属性。

将资源置于警告模式

将资源置于警告模式可监控访问规则的效果，而无需强制执行这些规则。

注意：除了将单独资源置于警告模式外，您还可以将类置于警告模式 (p. 172)。

将资源置于警告模式

1. 在 CA Access Control 端点管理 中编辑要置于警告模式的资源。

   将显示相应的“修改”页面。

2. 单击“审核”选项卡。

   将显示针对该资源的“审核模式”页面。

3. 选择“警告模式”，然后单击“保存”。

   您修改的资源现在处于警告模式。

注意：在警告模式中，当允许访问但该访问违反访问规则时，CA Access Control 始终将警告记录写入审核日志：您无需通过对资源设置审核属性来执行该操作。

使用 sereport 实用程序（报告编号 6）查看处于警告模式的所有资源。

示例：将文件置于警告模式

以下 selang 示例将文件 c:\myfile 置于警告模式：

```
chres FIlE c:\myfile warning
```

示例：清除文件的警告模式

以下 selang 示例将文件 c:\myfile 的警告模式清除：

```
chres FIlE c:\myfile warning-
```

myfile 现在不处于警告模式，因此 CA Access Control 将对 myfile 强制执行访问规则。

示例：将终端置于警告模式

以下 selang 示例将终端 myterminal 置于警告模式：

```
chres terminal myterminal warning
```

CA Access Control 允许任何授权用户从终端 myterminal 进行访问，但会对任何通常被拒绝从该终端访问的用户记录审核记录。

将类置于警告模式

您可以将类中的所有记录置于警告模式，而不是将记录逐个置于警告模式。 您可以使用警告模式来监控访问规则的结果，而无需强制执行这些规则。

将类置于警告模式

1. 在 CA Access Control 端点管理 中，执行如下操作：
   a. 单击“配置”。
   b. 单击“类激活”。
   将显示“类激活”页面。
2. 为要置于警告模式的类选中“警告”列中的复选框。
3. 单击“保存”。
   此时将显示一条确认消息，通知您已成功更新 CA Access Control 选项。

找出处于警告模式的资源

实施 CA Access Control 时，您应使用警告模式作为临时措施。 如果您确定用户具有访问其所需资源的相应权限，则应关闭警告模式，之后 CA Access Control 将开始强制执行相关联的规则。

要找出处于警告模式的资源，可以创建一个显示所有处于警告模式的资源的报告。

要创建报告，请输入以下命令：

```
sereport -r 6
```

CA Access Control 将创建报告。

注意：有关 sereport 实用程序的详细信息，请参阅 *《参考指南》*。

找出处于警告模式的类

实施 CA Access Control 时，您应使用警告模式作为临时措施。 如果您确定用户具有访问其所需资源的相应权限，则应关闭警告模式，之后 CA Access Control 将开始强制执行相关联的规则。

要找出处于警告模式的类，您可以使 CA Access Control 显示此数据。

要显示此数据，请输入以下 selang 命令：

```
setoptions cwarnlist
```

CA Access Control 将显示一个表，其中显示处于警告模式的类。

注意：有关 setoptions 的详细信息，请参阅 *《selang 参考指南》*。

如何执行系统维护

您可能需要在特定时间执行系统维护来升级系统、安装新应用程序等等。 在系统维护期间，您应当在警告模式下设置 CA Access Control 规则。 一旦您确定维护不会影响用户访问其所需要的资源，则应关闭警告模式，之后 CA Access Control 将开始强制执行相关联的规则。

要在执行系统维护时使用警告模式，请执行以下操作：

1. 使用以下 selang 规则在开始维护之前将适当的类设置为警告模式：

   ```
   setoptions class(NAME) flags(W)
   ```

2. 执行维护。
3. 在执行维护后运行 seretrust 实用工具。

   seretrust 实用程序可生成重新托管在数据库中定义的程序和安全文件所需的 selang 命令。
4. 运行 selang 命令来重新信任在数据库中定义的程序。
5. 使用以下 selang 规则从类中删除警告模式以便启用策略实施：

   ```
   setoptions class(NAME) flags-(W)
   ```

6. 查看 CA Access Control 审核日志文件。

   审核日志包含受维护影响的资源的警告。

注意：有关 seretrust 实用程序的详细信息，请参阅 *《参考指南》*。

审核日志

审核记录存储在称为审核日志的文件中。 审核日志的位置在 seos.ini 文件中指定。 可以使用 seaudit 实用程序或 CA Access Control 端点管理 列出审核日志中的记录事件，按照时间限制或事件类型等来筛选事件。

注意：有关 seaudit 的详细信息，请参阅*《参考指南》*。

审核日志存储在本地，但是您可以通过使用日志传递工具，使用 CA Access Control 来分发审核信息。 考虑将旧的审核日志存档到磁带中，以便以后浏览事件。

默认情况下，由于 seosd 程序由 root 用户执行，因此，授权后台程序 seosd 使用 root 用户所有权来创建审核日志。 出于同样的原因，将使用仅向 root 用户授予的读取/写入权限来创建审核日志。

为了使其他用户可以读取审核日志，而无需使用 su（替换用户）替换为 root 用户，CA Access Control 在 seos.ini 文件中包括两个条目，用于指定将那种组所有权分配给日志文件。

- 一个条目用于审核日志。

  假设您站点上的审核者是 auditforce 组的所有成员。 您希望这些用户可以浏览本地审核日志文件。 编辑 seos.ini 文件，以便将 [logmgr] 部分中的 audit_group 标记设置为 auditforce。 之后，CA Access Control 将授予 auditforce 组对本地审核日志的读取权限。 今后，在您的工作站创建的任何本地审核日志都将 auditforce 组作为其所有者。

  日志传递后台程序将参考同一标记，以了解谁应该对后台程序生成和收集的审核日志拥有访问权限。 注意，审核日志像任何其他文件一样受 Access Control 约束，而 CA Access Control 规则可以阻止用户访问这些日志。
- 另一个条目用于错误日志，它以同样的方式指定该文件的组所有权。

系统审核员

系统审核员是分配了 AUDITOR 属性的用户。 定义为系统审核员的用户可以执行审核任务，例如更改分配到用户和资源的审核属性。

可以从中央位置执行审核任务。 要在一台主机中收集网络上各种工作站的审核信息，审核员可以使用日志传递工具。

设置日志传递工具

设置日志传递

1. 创建日志传递配置文件。

   除非您使用 seos.ini 文件中的 RouteFile 标记另行指定，否则 CA Access Control 会将日志传递配置文件命名为 *ACInstallDir*/log/selogrd.cfg，

   其中 *ACInstallDir* 为 CA Access Control 的安装目录，默认为 /opt/CA/AccessControl/。

   您可以在目录 *ACInstallDir*/samples/selogrd.init 中找到日志传递配置示例文件。 您也可以创建由以下三行组成的文件，作为非常简单的日志传递配置文件：

   规则
   ```
   host destination
   ```
   。

   对于 *destination*，输入应接收审核记录的主机的名称。 系统将记录所有类、资源、访问者和结果。

   注意：有关配置文件语法的详细信息，请参阅*《实用程序指南》*中的 selogrd 实用程序。

2. 在要传递审核信息的所有主机上启动发射器后台程序 (selogrd)，并在要收集审核信息的所有主机上执行收集器后台程序 (selogrcd)。

   注意：有关使用这些后台程序的详细信息，请参阅*《参考指南》*。

文件通知

除了编译日志，日志传递工具还可以将通知发送到主机的显示屏、电子邮件地址或其他目标。 您可以基于工作站自有审核日志中的信息，或者收集器后台程序带到工作站的日志中的信息发送通知。

要设置此类通知，您需要使用日志传递配置文件*和* selang 命令。 例如，假设您希望针对 root 用户的 setuid 请求每次成功提出时都通知用户 John。

1. 发出以下 selang 命令：

   ```
   chres SURROGATE USER.root notify(John)
   ```

   该 chres 命令指定，每当有人将用户替代为 root 用户时，都将创建专门的审核日志记录，而 seosd 后台程序将通知用户 John。 该后台程序还创建一种称为*通知记录*的专门审核记录。

2. 当您为一种或多种资源指定了通知后，您可以向日志路由配置文件中添加以下三行。

   ```
   Rule2
   notify default
   。
   ```

   该行使日志传递发射器为通知审核记录创建邮件消息。

   注意：有关配置文件格式和设置日志传递后台程序的详细信息，请参阅*《参考指南》*。

日志路由

CA Access Control 使用日志传递后台程序 selogrd 将选中的本地审核日志记录分发到特定主机；将审核日志记录重新设置为电子邮件消息格式、ASCII 文件格式或用户窗口格式；以及基于审核的事件传输通知消息。

为了确定审核记录传递，selogrd 使用配置文件 selogrd.cfg。 该文件列出要传递（或不传递）的审核日志记录以及要传递（或不传递）到的位置。 有关该文件的完整说明，请参阅*《参考指南》*。

日志传递配置

要在 seosd 启动时自动启动 selogrd 或 selogrcd，请将 seos.ini 的 [daemons] 部分中的 selogrd 或 selogrcd 标记设置为 yes。 这样，当运行 seload 时，seload 将为您启动这些后台程序。

例如，seos.ini 的 [daemons] 部分中的相应内标识看起来如下：

```
selogrd = yes
selogrcd = yes
```

日志传递工具使用 RPC 传递审核记录，因此，将日志审核收集器置于防火墙后会禁止对 UDP 端口的简单阻塞，因为，这样无法知道 portmapper 为服务器后台进程分配了哪个端口。 要解决该问题，可以使用标记 ServicePort 为服务器后台程序分配预定义的端口。

如果防火墙允许网络外部的端口 111（portmapper 端口），则只应更改服务器中的 seos.ini 文件。 如果防火墙不允许在受保护的网络中与 portmapper 进行通信，则客户端和服务器必须就特定端口达成一致。

通过将客户端和服务器的 seos.ini 文件中的 ServicePort 标记设置为同一值，可以确保实现这一点。 您可以指定一个数字（表示后台程序绑定到指定端口）或服务名。 如果指定服务名，则客户端和服务器都必须使用相同的服务解析。 例如，如果指定服务名 seoslogr，则需要向客户端和服务器的 /etc/services 文件添加以下内容：

```
seoslogr 2022/udp # Audit log-routing
```

如果客户端或服务器使用 NIS 来解析服务，则必须更新 NIS 服务映射。

审核日志传递加密

您可以加密审核日志记录。 使用加密时，selogrd 后台程序会先加密审核日志记录，然后再将其发送到收集器（selogrcd 或审核日志传递器）。 收集器则会依次将收到的记录解密。

CA Access Control 为 selogrd 提供了两种加密方式：CA Access Control 标准加密和通过 adcipher 的审核日志加密。 为了进行加密，selogrd 按照 seos.ini 文件中 [selogrd] 部分的指定，使用共享库对象的功能。

标准加密使用共享库 libcrypt；审核加密使用由 CipherName 标记指定的文件中的功能。 默认情况下，该文件名为 adcipher，它是指向所需共享库的符号链接。 CA Access Control 安装过程将四个共享库置于 CA Access Control/lib 目录中，这四个共享库分别为 lib1des、lib3des、libIDEA 和 libblowfish。

CA Access Control 在共享库中维护标准加密密钥，而审核加密使用由 KeyFile 标记指定的单独文件（默认值：adcipher.bin）。

使用 UseEncryption 内标识确定加密类型：

- 要使用 CA Access Control 标准加密，请指定 UseEncryption=native
- 要通过 adcipher 使用审核日志加密，请指定 UseEncryption=eTrust，并为 CipherName 和 KeyFile 标记输入适当的值。
- 要禁用 selogrd 加密，请指定 UseEncryption=no。

使用 RefuseUnencrypted 标记接受或拒绝未加密的审核。 该标记与 UseEncryption 标记一起使用，如果 UseEncryption 设置为 no，则该标记为冗余：

- 要拒绝未加密的审核，请指定 RefuseUnencrypted=yes
- 要接受已加密和未加密的审核，请指定 RefuseUnencrypted=no

注意：selogrcd 后台程序使用 seos.ini 文件中的相同标记。

要更改加密密钥，请使用本章中说明的 sechkey 实用工具。

重要说明！ 如果向审核收集器发送记录，请确保 selogrd 和收集器使用相同的共享加密文件和加密密钥。

通过电子邮件发送审核日志记录

selogrd 可以直接将记录发送到电子邮件目标。 您可以通过邮件程序实用程序将电子邮件定向（旧方法），也可以使用 SMTP 将电子邮件直接发送到邮件交换服务器。

要将审核日志记录直接发送到邮件交换服务器，请设置 seos.ini 文件的 [selogrd] 部分中的 UseSmtpMail 标记。

您还可以指定以下内容：

- 使用 StmpTimeLimit 标记指定邮件服务器没有应答时的超时时间
- 使用 SmtpMailFrom 内标识指定“发件人：”邮件头字段
- 使用 SmtpMailServer 内标识指定邮件服务器主机地址

注意：该方法不使用 UNIX 邮件实用程序，而是与邮件服务器建立直接连接，并使用 SMTP 协议发送邮件。

配置 SNMP 陷阱

对于使用 Internet 网络管理协议 SNMP（简单网络管理协议）的系统，可以将 selogrd 配置为使用 CA Access Control 审核记录创建 SNMP 陷阱。

要实施 SNMP 陷阱，首先要找到 CA Access Control 库中提供的 SNMP 共享对象，然后使用这些共享对象正确配置 selogrd。

注意：如果您希望使用 selogrd 的 SNMP 扩展，并且 CA Access Control 未安装在默认位置 (/opt/CA/AccessControl/)，则必须先设置一个环境变量，然后才能运行 selogrd。 环境变量如下所示，其中 *ACInstallDir* 是安装 CA Access Control 的目录：

- 在 AIX 中，将 LIBPATH 设置为 *ACInstallDir*/lib
- 在 Solaris 中，将 LD_LIBRARY_PATH 设置为 *ACInstallDir*/lib
- 在 LINUX 中，将 LD_LIBRARY_PATH 设置为 *ACInstallDir*/lib
- 在 HP 中，将 SHLIB_PATH 设置为 *ACInstallDir*/lib

共享对象（通常可在目录 *ACInstallDir*/lib 中找到）称为 snmp.*xx* 和 libsnmp.*xx*，其中 *xx* 扩展名因平台而异。 可能的扩展名有：

- **.o** - AIX 平台
- **.sl** - HP 平台
- **.so** - 所有其他平台

如果您希望使用 selogrd 的 SNMP 扩展名，并且 CA Access Control 未安装在默认位置，则必须先设置以下环境变量，再运行 selogrd：

- 在 AIX 中，将 LIBPATH 设置为 *ACInstallDir*/lib
- 在 Solaris 中，将 LD_LIBRARY_PATH 设置为 *ACInstallDir*/lib
- 在 LINUX 中，将 LD_LIBRARY_PATH 设置为 *ACInstallDir*/lib
- 在 HP 中，将 SHLIB_PATH 设置为 *ACInstallDir*/lib

其中，*ACInstallDir* 是安装 CA Access Control 的目录。

将 selogrd 配置为使用共享对象

1. 创建名为 *ACInstallDir*/etc/selogrd.ext 的对象。
2. 向文件 *ACInstallDir*/etc/selogrd.ext 中添加一行（其中包括 snmp.so 的适当路径），从而定义 SNMP 共享对象的位置。 （指定该共享对象即可自动链接其他对象。）例如：

   ```
   snmp /opt/CA/AccessControl//lib/snmp.so
   ```

3. 最后，必须配置 selogrd.cfg 文件，以指定应该触发 SNMP 陷阱的操作类型以及触发 SNMP 陷阱时应该通知的位置。 配置与其他审核通知的配置非常相似，将传送系统指定为 snmp。

   例如，假设您希望在启动和关闭 CA Access Control 时激活 SNMP 陷阱，并将这些 SNMP 陷阱的通知发送到 AuditPC。 通过将以下部分添加到 selogrd.cfg 配置文件，可以完成此操作：

   ```
   snmpRule
   snmp AuditPC
   include Class(START).
   include Class(SHUTDOWN).
   。
   ```

同样，您可以通过其他操作或访问类型激活 SNMP 陷阱，或者将 SNMP 陷阱发送到其他位置。

迁移用户跟踪筛选器

如果将用户设置为可跟踪，则每次写入该用户的跟踪记录时，将向 seos.audit 文件写入匹配的审核记录。 在先前版本的 <eAC > 中，这些审核记录由 trcfilter.init 文件筛选。 在 CA Access Control r12.0 SP1 及更高版本中，用户跟踪记录生成的审核记录由 audit.cfg 文件筛选，该文件筛选所有其他审核记录。

您必须手工将审核记录筛选从 trcfilter.init 迁移到 audit.cfg。 如果不迁移筛选器，则不会筛选用户跟踪所生成的审核记录。

注意：跟踪记录仍旧由 trcfilter.init 筛选。 不将跟踪筛选器从 trcfilter.init 迁移到 audit.cfg。

迁移用户跟踪筛选器

1. 在 trcfilter.init 中，查找需要迁移的用户跟踪筛选器。

   在 seos.ini 文件的 seosd 部分中的 trace_filter 设置确定该文件的位置。

2. 在 audit.cfg 中，请输入以下内容，其中 *usertracefilter* 是来自 trcfilter.init 的用户跟踪筛选器。

   ```
   TRACE;*;*;*;*;usertracefilter
   ```

3. （可选）对每个要迁移的用户跟踪筛选器重复步骤 1 和 2。

示例：迁移用户跟踪筛选器

在此示例中，以下用户跟踪筛选器位于 trcfilter.init 文件中：

```
*ExampleFilter
```

要迁移此用户跟踪筛选器，请在 audit.cfg 文件中另起一行键入以下内容：

```
TRACE;*;*;*;*;*ExampleFilter
```

第 14 章： 管理权限的范围

此部分包含以下主题：

全局权限属性

全局授权属性在用户记录中设置。 每个全局授权属性都允许用户执行某些类型的功能。 本节介绍每个全局授权属性的功能和限制。

ADMIN 属性

通过 ADMIN 属性，用户可以执行 CA Access Control 中几乎所有的命令。 在数据库中定义的具有 ADMIN 属性的用户可以定义和更新数据库中的用户、组和资源。 这是 CA Access Control 中最强大的属性，但是它也有限制：

- 如果数据库中只有一个用户具有 ADMIN 属性，则无法删除该用户，而且也无法从记录中删除 ADMIN 属性。
- 具有 ADMIN 属性但没有 AUDITOR 属性的用户不能更改对用户、组或资源所做的审核的类型（审核模式）。 如果您有 ADMIN 属性且需要更改用户、组或资源的审核特性，请为自己分配 AUDITOR 属性。
- 具有 ADMIN 属性的用户无法删除超级用户（UNIX 上的 root 帐户或 Windows 上的 Administrator 帐户），但他们可以将 root 用户设置为非 ADMIN 用户。

AUDITOR 属性

具有 AUDITOR 属性的用户可以监视系统使用情况。 对于具有 AUDITOR 属性的用户，其显式权限包括以下内容：

- 用户可以显示数据库中的信息。

  审核员可以执行 selang 命令 *showusr*、*chgrp*、*chres* 和 *showfile*。

- 用户可以设置现有记录的审核模式。

  审核员可以执行 selang 命令 *chusr*、*chgrp*、*chres* 和 *chfile*。

OPERATOR 属性

具有 OPERATOR 属性的用户拥有对所有文件的 READ 访问权限。 使用该访问权限，他们可以列出数据库中的任何内容，且可以运行备份作业。 操作员可使用 showusr、showgrp、showres、showfile 和 find 命令列出数据库记录。 通过 OPERATOR 属性，用户还可以使用 secons 实用程序。

注意：有关 secons 实用程序的详细信息，请参阅 *《参考指南》*。

PWMANAGER 属性

PWMANAGER 属性为常规用户提供使用 chusr 或 sepass 命令更改其他用户的密码的权限。

注意：要通过 PWMANAGER 更改 ADMIN 用户的密码，请设置 setoptions 命令的 *cng_adminpwd* 选项。 有关详细信息，请参阅 *《参考指南》*。

PWMANAGER 属性不包括更改宽限登录次数、其他用户的密码间隔或常规密码规则的权限。

PWMANAGER 的权限还包括 showusr 和 find 命令的使用。

注意：如果用户将 nochngpass 属性设置为 yes，则 PWMANAGER 无法更改该用户的密码。

SERVER 属性

与许多其他安全模型一样，CA Access Control 不允许常规用户询问：“用户 A 是否可以访问资源 X？”，常规用户可以询问的唯一问题是：“我是否可以访问资源 X？”，不过，它应该允许向许多用户提供服务的进程（例如数据库服务器服务或内部应用程序）代表其他用户询问权限。。

SERVER 属性允许进程询问用户权限。 具有 SERVER 属性设置的用户可以发出 SEOSROUTE_VerifyCreate API。

注意：有关服务器属性和 CA Access Control API 的详细信息，请参阅*《SDK 指南》*。

IGN_HOL 属性

通过 IGN_HOL 属性，用户可以在假期记录中定义的任何时间段内登录。 HOLIDAY 类中的每个记录定义一个或多个时间段，在这些时间段内，用户需要额外权限才能登录。 通过 IGN_HOL 属性，用户可以随时登录，不受假期记录中定义的时间段的限制。

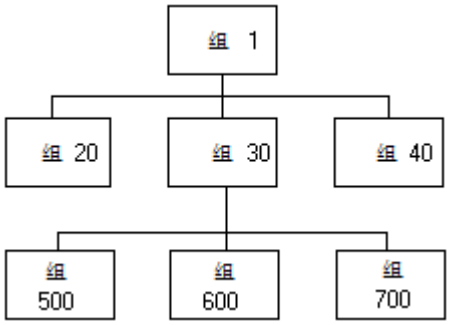
注意：有关 HOLIDAY 类的详细信息，请参阅*《参考指南》*。

组授权

在讨论组授权属性之前，有必要理解父子关系的概念。

父子关系

从属和超级组的概念（也称为父子关系），在讨论组管理权限时非常重要。 一个组可以是一个或多个组的父组（超级组）。 *子*组或从属组只能拥有一个父组。 为组分配一个父组是可选操作。 请考虑下图：



组 1 是 20、30 和 40 三个组的父组。 组 30 是 500、600 和 700 三个组的父组。 组 600 只有一个父组 30。 组 1 没有父组。

组授权限属性

包括像资源记录和访问者记录在内的所有记录都有所有者。 拥有一个记录意味着拥有查看、编辑和删除该记录的权限。

一个组可以拥有其自己的记录。 然而，在拥有记录的组中，只有某些特权用户才能管理记录。 这些特殊用户在自己的用户记录中设置了组授权属性。 组授权属性包括：

- GROUP-ADMIN
- GROUP-AUDITOR
- GROUP-OPERATOR
- GROUP-PWMANAGER

join 命令（只有正确授权的用户才可以发出）设置这些属性。 join 命令用于将用户置于组中，并指定用户的组授权属性（如果有）。

拥有权限的组成员可能或不可能被授权管理定义组成员的用户记录，这取决于谁拥有这些记录。

更多信息：

所有权 (p. 188)

GROUP-ADMIN 属性

拥有组管理授权属性的用户可以创建某组记录。 要创建记录，组管理员必须指定记录的所有者。

记录的所有者必须是该用户在其中拥有组授权属性的组。 如果该组是其他组的父组，则所有者还可以来自与其中的一个子组。 整组记录被称为组范围。 提供的授权示例说明了组范围的概念。

具有 GROUP-ADMIN 属性的用户对他们组范围中的记录拥有以下访问权限：

访问	说明	命令
读取	显示记录的属性。	showusr、showgrp、showres、showfile
创建	在数据库中创建新记录。 您必须指定所有者。	newusr、newgrp、newres、newfile
修改	更改记录的属性。	chusr、chgrp、 chres、chfile
删除	删除数据库中的记录。	rmusr、rmgrp、rmres、rmfile
连接	将用户加入组或者将用户与组分离。	join、join-

GROUP-ADMIN 属性也有限制：

- GROUP-ADMIN 用户无法阻止自己对资源的访问，因此：
  - GROUP-ADMIN 用户无法分配高于自己安全级别的安全级别。
  - GROUP-ADMIN 用户无法分配他们没有的安全类别或安全标签。
- GROUP-ADMIN 用户无法从数据库中删除超级用户（UNIX 上的 root 帐户或 Windows 上的 Administrator 帐户）。
- 几种限制与本章中的“全局权限属性”所述的全局权限属性有关：
  - GROUP-ADMIN 用户无法删除数据库中唯一的 ADMIN 用户记录。
  - GROUP-ADMIN 用户无法删除数据库最后一个 ADMIN 用户记录中的 ADMIN 属性。
  - 没有 AUDITOR 属性的 GROUP-ADMIN 用户无法更新审核模式。只有具有 AUDITOR 属性的 GROUP-ADMIN 用户可以更新审核模式。
  - GROUP-ADMIN 用户无法为任何用户设置全局授权属性 ADMIN、AUDITOR、OPERATOR、PWMANAGER 和 SERVER。

GROUP-AUDITOR 属性

具有 GROUP-AUDITOR 属性的用户可以列出组范围中任何记录的属性。组审核者还可以为组范围中的任何记录设置审核模式。

GROUP-OPERATOR 属性

具有 GROUP-OPERATOR 属性的用户可以列出组范围中任何记录的属性。

GROUP-PWMANAGER 属性

具有 GROUP-PWMANAGER 属性的用户可以更改其记录位于组范围中的任何用户的密码。

所有权

数据库中的每个记录（包括访问者记录和资源记录）都有一个所有者。当您向数据库中添加记录时，您可以使用 owner 参数来显式指定其所有者，也可以通过 CA Access Control 将定义记录的用户指定为记录的所有者。

如果以下内容的*任何*一条为真，访问者即拥有记录：

- 他们被定义为记录的所有者。
- 他们是定义为记录所有者的组的成员*并且*已加入具有 GROUP-ADMIN 属性的组。
- 他们是资源所属的资源组记录的所有者。

如果您删除拥有数据库中记录的用户或组，则这些记录不再具有所有者。

拥有记录的用户对他们所拥有的记录享有以下访问权限：

访问	说明	命令
读取	显示记录的属性。	showusr、showgrp、showres、showfile
修改	更改记录的属性。	chusr、chgrp、 chres、chfile
删除	删除数据库中的记录。	rmusr、rmgrp、rmres、rmfile

访问	说明	命令
连接	将用户加入组或者将用户与组分离。	join、join-

如果您不想让某用户或组对特定记录具有所有权权限，则为该记录以及该记录所属的任何资源组记录分配所有者 *nobody*。

所有权权限的限制如下：

- 数据库中最后一个 ADMIN 用户的所有者无法删除该用户的记录。
- 不具有 ADMIN 属性的所有者无法更新审核模式。 只有具有 AUDITOR 属性的用户才可以更新审核模式。
- 超级用户（UNIX 上的 root 帐户或 Windows 上的 Administrator 帐户）的所有者不能从数据库中删除 root。
- 所有者无法为他们所拥有的用户设置全局权限属性 ADMIN、AUDITOR、OPERATOR 和 PWMANAGER。
- 所有者无法阻止自己对资源的访问，因此：
  - 所有者无法分配高于自己安全级别的安全级别。
  - 所有者无法分配他们没有的安全类别或安全标签。

文件所有权

通过 CA Access Control，文件所有者可以通过在 FILE 类中定义记录来保护文件。 文件的所有者对该文件的记录具有完全权限，所以所有者可以将具有所有参数的 newfile、chfile、showfile 和 authorize 命令用于保护该文件的记录。

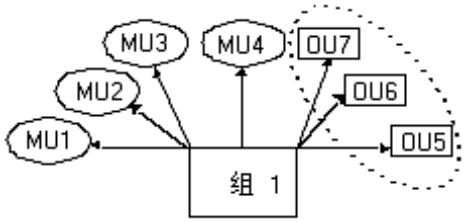
在 UNIX 上，当用户创建文件时，UNIX 会将该用户指定为该文件的所有者。 通过 CA Access Control，UNIX 文件所有者可以定义 FILE 记录，除非该功能被显式禁用。 如果您不需要文件所有者定义 FILE 记录，请确保将 seos.ini 文件 [seos] 部分中的 use_unix_file_owner 标记设置为 no。（这是默认设置）

授权示例

下图说明了组授权属性、父子关系、所有权、成员资格和组范围的概念。 这些图只包含用户和组，但所有权概念也适用于资源和文件记录。

单个组授权

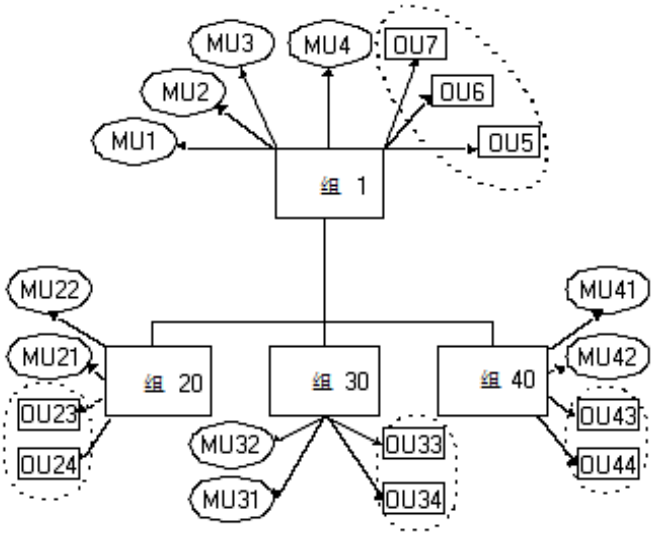
在下图中，四个用户 MU1、MU2、MU3 和 MU4 都是 Group 1 的成员。 Group 1 还拥有自己的三个用户 OU5、OU6 和 OU7。 成员 MU4 具有 GROUP-ADMIN 属性。÷



椭圆表明用户 MU4 执行的命令的组范围。 它包括组 1 拥有的所有用户，即 OU5、OU6 和 OU7。

父组和子组

在下图中，四个用户 MU1、MU2、MU3 和 MU4 都是 Group 1 的成员。 Group 1 还拥有自己的三个用户 OU5、OU6 和 OU7。 成员 MU4 在其记录中设置了 GROUP-ADMIN 属性。



组 1 是 20、30 和 40 三个组的父组。 其中的每个从属组都有两个属于该组成员的用户以及两个该组所拥有的用户。

四个椭圆表明用户 MU4 执行的命令的组范围。 它包括 Group 1 拥有的所有用户以及从属于 Group 1 的组所拥有的用户。 MU4 的组范围中的用户是 OU5、OU6、OU7、OU23、OU24、OU33、OU34、OU43 和 OU44。

如果有组从属于拥有用户、组或资源的组 20、30 或 40，则这些组所拥有的记录也属于用户 MU4 执行的命令的组范围。

子管理

安全管理员（具有 ADMIN 属性的用户）可以向常规用户授予特定管理权限。 这些常规用户则称为子管理员。 子管理员只有管理指定的 CA Access Control 类或对象的权限。 例如，可以授权子管理员只管理用户和组对象。 可以通过向子管理用户授予某类中特定对象的管理权限，来设置更高级别的子管理。

用户、组和资源的子管理员可以使用 selang 执行与这些资源相关的管理任务。

如何将特定管理权限授予常规用户

由于管理员（具有 ADMIN 属性的用户）几乎可以执行 CA Access Control 中的所有操作，因此您可能需要将特定管理任务指派给子管理员。 要执行此操作，您需要向这些用户授予对 CA Access Control 数据库中的类的权限，通过这些权限可以控制用户需要执行的特定管理任务，如下所示：

1. 识别控制您要指派的任务的一个或多个类。

   例如，CA Access Control 使用 USER 和 GROUP 类来创建访问者资源。如果您要指派访问者管理，那么您需要使用 ADMIN 类的 USER 记录和 GROUP 记录。

2. 将一个或多个子管理员授权到 ADMIN 类的适用资源。

   例如，要让一个子管理员查看和修改用户记录，请向该用户授予对 ADMIN 类的 USER 记录的*读取*和*修改*访问权限。

ADMIN 类

子管理员（类 ADMIN 中记录的 Access Control 列表 (ACL) 列出的用户）所拥有的权限与具有 ADMIN 属性的用户的权限相类似。 不过，ADMIN 类中记录的 ACL 中用户的权限仅限于记录代表的特定类。 例如，ADMIN 类中的 SURROGATE 记录确定哪些用户可以管理 SURROGATE 类的记录。

注意：有关 CA Access Control 类的详细信息，请参阅*《参考指南》*。

ADMIN 类中特定记录的 ACL 中的用户可以执行以下命令：

访问	说明	命令
读取	显示类中记录的属性。	showusr、showgrp、showres、showfile、find
创建	在类中创建新数据库记录。	newusr、newgrp、newres、newfile
修改	更改类中的属性。	chusr、chgrp、chres、chfile
删除	删除数据库中现有的类记录。	rmusr、rmgrp、rmres、rmfile
连接	向组中添加用户和删除组中的用户。该访问权限仅在 GROUP 记录的 ACL 中有效。	join、join-

访问	说明	命令
密码	控制数据库中所有用户的密码及其密码属性。 该访问权限授予的权限与具有 PWMANAGER 属性的用户的访问权限相同。 该访问权限仅在 USER 记录的 ACL 中有效。	chusr

拥有 ADMIN 类权限的用户有以下限制：

- 在 ADMIN 类的 USER 记录的 ACL 中定义的用户无法删除数据库中的最后一个 ADMIN 用户。
- ADMIN 类用户无法为他们所拥有的用户设置全局权限属性 ADMIN、AUDITOR、OPERATOR 和 PWMANAGER。
- ADMIN 类用户无法强制更新审核模式。 只有具有 AUDITOR 属性的 ADMIN 类才可以更新审核模式。
- ADMIN 类用户无法删除超级用户（UNIX 上的 root 帐户或 Windows 上的 Administrator 帐户），但他们可以将 root 设置为 NOADMIN。
- ADMIN 类用户无法阻止自己对资源的访问，因此：
  - ADMIN 类用户无法为资源分配高于自己安全级别的安全级别。
  - ADMIN 类用户无法分配他们没有的安全类别或安全标签。

这些限制是 B1 安全级别认证的一部分。

环境注意事项

您在环境中所占据的位置是决定您是否可以更新数据库中的信息的因素之一。

远程管理限制

您可以通过网络访问远程工作站，以及更新远程工作站上的数据库。要更新远程工作站上的数据库，您和您的终端都需要有权限。

- 您必须被显式定义为远程工作站数据库中的用户。 无论要执行何种命令，都必须在远程工作站数据库中您的用户记录中设置适当的属性。
- 您必须在规则中显式说明本地终端的需求，对本地终端授予访问远程工作站的写入访问权限；否则，您无法在远程工作站执行 CA Access Control 管理。

  借助默认访问字段 (_default) 或 UACC 类的写入访问权限，您可以在远程工作站输入 selang 命令 shell。 不过，您**无法执行任何 selang 命令或者以其他方式访问远程数据库。 使用读取访问权限，您可以登录远程工作站，但无法在该工作站执行 CA Access Control 管理。**

  以下示例说明了 WRITE 和 READ 权限之间的区别：

  1. 要指定默认访问权限为 READ 的新终端，即管理员可以从该终端登录，但却无法处理其中的数据库，请发出以下命令：

     ```
     newres TERMINAL tty13 defacc(read)
     ```

  2. 要授予用户 ADMIN1 权限，以便从新终端处理数据库（即授予 WRITE 权限和 READ 权限），请发出以下命令：

     ```
     authorize TERMINAL tty13 uid(ADMIN1) access(r,w)
     ```

UNIX 环境

对于管理 UNIX 中的用户和组，CA Access Control 中拥有全局或组授权属性的用户对 UNIX 的权限和限制与他们对 CA Access Control 的权限和限制相同。

如果您在*没*运行 seosd 后台进程时使用 selang（例如在安装时），则必须遵守以下规则：

- 必须在 selang 命令中包括 -l 选项。
- selang 的用户必须是 root 用户。 （这种独占的 root 权限符合常规的 UNIX 限制。）

Windows 环境

在本地 Windows 环境中有效

当 CA Access Control 正在运行时，如果您使用 selang 更改本机 Windows 环境中的资源，CA Access Control 代理则会在相应的 Windows 库中更改资源。 您不需要任何其他的 Windows 权限来更改资源。 这意味着，当 CA Access Control 中具有全局或组授权属性的用户在本机 Windows 环境中执行 selang 命令时，与其在 CA Access Control 中执行该操作具有相同的 Windows 权限和限制。

当 CA Access Control 未在运行时，如果您使用 selang 更改本机 Windows 环境中的资源，您必须遵守如下规则：

- 必须在 selang 命令中包括 -l 选项
- 必须具有 ADMIN 属性或子管理权限
- 必须具有足够的 Windows 权限来更改资源

  出现该限制的原因是由于 selang 进程（而不是 CA Access Control 代理）在 Windows 库中更改资源。

例如，用户 Emma 想使用 chfile selang 命令在本机 Windows 环境中更改文件 C:\tmp.txt 的所有者。 如果 CA Access Control 正在运行，Emma 则需要足够的 CA Access Control 权限来更改文件拥有者，但是不需要额外的 Windows 权限。 如果 CA Access Control 未在运行，Emma 则同时需要 CA Access Control 和 Windows 权限来更改文件拥有者。

第 15 章： 提高性能

此部分包含以下主题：

使用全局访问检查

全局访问检查 (GAC) 功能允许您以比其他可能存在的方式快得多的速度访问经常打开的受保护文件（其访问规则不可能更改）。

GAC 使 CA Access Control 管理员可以缓存针对读取、写入、chown、chmod、重命名、撤消链接、utimes、chattr、链接、chdir、创建和所有的规则，因此无需将控制传递到 seosd 即可获得对文件的适当访问权限。 默认设置为所有。 但是执行请求不适于使用 GAC，因为它们可能会带来安全漏洞。

如果没有 GAC，则每当用户或程序尝试访问受保护的文件时，CA Access Control 将运行全面的安全检查。 频繁访问的文件需要反复的深入检查以便确认访问权限。

通过 GAC，CA Access Control 管理员可以理所当然地认为某些经常被访问的受保护文件需要较简短的安全检查。 CA Access Control 管理员可以选择适合于进行较简短检查的文件。 文件必须先通过基于设置规则的全面安全检查，之后 CA Access Control 才允许进行较简短的安全检查。 规则本身由通用文件名和访问列表组成。 根据用户缓存规则。

选择某些文件进行较短检查是有可靠保证的，原因在于：借助适当的 GAC 功能，如果对关于受保护文件的规则进行了实际更改，则会刷新较短安全检查表，并启动最初的全面安全检查。

注意：GAC 限制表明此功能适用于除 root 用户以外的所有用户。

GAC 工作原理

CA Access Control 可监控对指定文件的访问，并在执行期间构建所允许访问的表格。 这些是为了设置 GAC 规则提前指定的文件。

每当 CA Access Control 决定应该授予用户对某个文件某一级别的访问权限时，它会检查是否满足以下两个附加条件：

- 授予的访问权限是无条件的（即不依赖于时间、日期以及要执行的程序，或其他类似的条件）。
- 文件与它预先选定的一个文件掩码设置相匹配。

注意：文件规则定义对文件的访问权限。

如果满足了这些条件，CA Access Control 会生成一个 UID 文件规则访问“三合一”，并将其存储在由此类“三合一”组成的表中。 在发生任何数据库访问规则解释之前先检查该表。 每当用户尝试访问文件时，均会将此表作为筛选机制查询。

该表可以形象地描述为“无需调用”表，因为其包含文件掩码的列表，一旦识别了这些掩码，就不再需要接受访问权限检查。 它也可被描述为“总是授予权限”表，因为总是对文件掩码列表中指定的文件授予访问权限。

每当用户尝试访问文件时，即会查询此表。 如果文件与表中查找到的一个“三合一”匹配，则授予适当的访问权限，而无需将控制传递给 seosd。 这可以跳过访问规则分析。 因此，根据存储在表中的“三合一”，将允许所有符合这种模式的文件访问无需查询访问规则数据库。

每当向数据库中添加新的访问规则时，都会刷新整个表，并从头开始了解过程。

实施 GAC

要设置 GAC，您必须为经常访问的文件组选择掩码，设置包含这些文件掩码的 GAC 文件，然后重新启动缓存进程。

设置 GAC 规则

注意：数据库中的文件规则使用类 FILE 参数和文件掩码创建。 规则应用到与文件掩码匹配的所有文件。 FILE 访问类型包括：全部、chdir、控制、创建、删除、执行、无、读取、重命名、sec、更新、utime、写入。

从数据库中定义的文件规则中，选择您要缓存的文件掩码。 将文件掩码的列表，以与文件掩码在数据库中显示的形式完全相同的方式，输入到 *ACInstallDir*/etc/GAC.init 文件中（其中 *ACInstallDir* 是 CA Access Control 的安装目录，默认情况下为 /opt/CA/AccessControl/）。

应该在单独的行上指定每个此类掩码。 例如，如果数据库包含 /tmp/mydir/* 的文件掩码，而您要将其缓存，请将以下行添加到 *ACInstallDir*/etc/GAC.init 文件中：

```
/tmp/mydir/*
```

注意：无法在 GAC.init 文件中指定特定的文件名。 只使用文件掩码。

启动 GAC

要将 CA Access Control 的当前版本转换成 GAC 兼容版本，请准备具有可以缓存的文件掩码的文件 *ACInstallDir*/etc/GAC。 只能使用文件掩码。

例如，*ACInstallDir*/etc/ 中的文件 GAC.init 仅包含以下一行：

```
/IBBS/REL63/*
```

GAC 约束

GAC 实施已经证明非常有效，尤其是在瞬间访问数百个文件的情况下，但它却有以下约束：

- 默认情况下，GAC 规则不适用于 root 用户（通常为 ADMIN）。 要使该规则适用于 root 用户，请设置 seos.ini 文件 [SEOS_syscall] 部分中的以下标记：

  ```
  GAC_root=1
  ```

  该标记的默认值为 0。 要还原默认值，请将该标记设置为 0 或删除该标记。

- 您不得在表中包括有条件保护的文件规则(例如日期或时间限制、程序通路等)。 如果您确实需要在 GAC.init 文件中指定此类文件规则，则将不再适用日期或时间限制以及其他限制。

- GAC.init 文件中不得包括具有 audit(ALL) 或 audit(success) 属性的文件规则。 如果在 GAC.init 文件中指定了此类文件规则，则不会记录对成功访问进行的审核。
- 筛选进程使用真正的（当前的）UID（即，与执行时的进程关联的 UID）。 这给原始 UID（用户最初用于登录的 UID）和非当前 UID 的 CA Access Control 跟踪带来了漏洞。（UID 使用的 CA Access Control 实施跟踪可提供更具责任的安全保护。）

  让我们来看一个别人可能怎样试图利用该漏洞的示例。 用户 Tony 未被授权访问文件 Accounts/tmp。 因此，Tony 代替（通过 /bin/su）被*授权*访问 Accounts/tmp 的用户 Sandra。 如果 Sandra 已经访问了 Accounts/tmp 文件，则该文件会以她的 UID 出现在“无需调用”表中。 之后，通过使用 Sandra 的 UID 允许 Tony 访问该文件。 这是因为内核代码并不保存 UID 的历史记录。

  但是，如果 Sandra 以前没有访问过该文件，则将使用 seosd 定期检查访问权限，然后拒绝 Tony 访问该文件。 要关闭该漏洞，ADMIN 用户必须保护数据库中的 SURROGATE 对象。 在本示例中，ADMIN 可以向数据库中添加以下规则：

  ```
  newres SURROGATE USER.Sandra default(N) owner(nobody)
  ```

  该命令确保 Tony 无法使用 su 命令来获得 Sandra 的访问权限。
- 如果访问者是 root 用户，则缓存系统没有任何影响。 原因是如果不查询数据库，就不会向 root 用户授予任何访问权限。

GAC 疑难解答

您可以按以下步骤测试 GAC 以查看它是否正在工作：

1. 启用跟踪 (secons --t+)。
2. 访问与 GAC.init 中指定的文件掩码之一相对应的文件。 跟踪期间应报告第一次访问。
3. 尝试再次访问该文件。 跟踪期间不应该记录第二次文件访问。

   如果记录了第二次访问，则 GAC 未正常工作。 检查 GAC.init 以查看它是否包含正确的格式。

使用资源缓存

CA Access Control 提供的另一个提高性能的工具是资源缓存（文件缓存）。

缓存将“记住”对 FILE 类中的资源授权请求的上一次回答（允许或拒绝）。其结果与文件名、用户名和授权响应（访问模式、程序名和结果）一起保存。 请求相同的权限时，会使用存储在缓存表中的最后一个响应来回答请求。 这可以节约时间，因为 CA Access Control 无需重新评估请求；CA Access Control 可以立即返回答案。 当规则被更改时，缓存立即自动同步。

缓存是一个运行时间表。 管理员可以使用两种方式来配置它：

- 在 seos.ini 文件中设置初始化参数。
- 将缓存切换到 ON 或 OFF，并在运行时更改参数。

安全管理员可以在 seos.ini 文件中定义表的大小、清理表的时间间隔以及包含标记的其他内部表参数。

拥有管理权限的用户可以将缓存表切换到 ON 或 OFF，更改缓存参数并将缓存表写入标准输出。

注意：有关 secons 实用程序或 seos.ini 初始化文件中 [seosd] 部分的详细信息，请参阅 *《参考指南》*。

调优建议

使用这些建议可以进一步提高性能：

- 如果三张表（池）中的一张表有最大的记录数，而另一张表没有，则展开完整表的大小。

  注意：三张表是：文件、用户和授权。

  如果某个池具有较低设置，则提高这些设置以展开池。
- 您必须设置时才设置最大大小标记。 表越大，则扫描记录所耗费的时间就越多。

使用网络缓存

网络或 IP 缓存功能存储已接受的传入 TCP 请求，因此，不会被发送到数据库；但是，系统会自动允许它们使用 syscall 功能。 该功能可以提高启动多种传入的 TCP 连接的主机性能。

要激活 IP 缓存功能，请更改 seos.ini 文件 [seosd] 部分中的下列标记，然后重新启动 CA Access Control：

network_cache_timeout

定义清理缓存表的频率。 如果您要对接受请求设置时间限制，则该标记非常重要。

UseNetworkCache

将该标记设置为 yes 以激活 IP 缓存。

启用缓存时，所有接受的 TCP 连接都将保存在内核表中。 记录由对等 IP 地址、对等端口和本地端口组成。 在该缓存中搜索每个新连接。 如果找到一组与 IP 地址、IP 端口和本地端口匹配的数据，则立即允许使用该连接。 这缩短了建立连接的时间。

使用真实路径缓存

文件名解析是一个长过程，因为 CA Access Control 使用文件系统中的信息。 CA Access Control 的内核在截获相应事件时，将节点编号转换成文件全名。 真实路径缓存在内部表中保存文件名。

要启用该功能，请将 seos.ini 文件 [SEOS_syscall] 部分中的标记 cache_enabled 设置为 1。 文件名与数据对（索引节点编号和设备号）一起缓存在表中。

注意：有关 seos.ini 初始化文件的详细信息，请参阅 *《参考指南》*。

使用派生同步

创建新进程时，seos.ini 文件 [SEOS_syscall] 部分中的再生同步标记 (synchronize_fork) 管理再生事件行为。 由于再生事件频繁，因此降低该标记的值可提高性能。

注意：有关 seos.ini 初始化文件的详细信息，请参阅 *《参考指南》*。

使用高优先级

CA Access Control 包含一个选项，用于为某些平台上的 seosd 后台程序设置实时优先级。 要激活该功能，请将 seos.ini 文件 [seosd] 部分中的 rt_priority 标记设置为 yes。 实时运行可提高系统性能。

注意：有关 seos.ini 初始化文件的详细信息，请参阅 *《参考指南》*。

绕过进程文件系统

为了降低系统负载，您可以指定当文件属于进程文件系统 (/proc) 时 CA Access Control 是否应该检查文件访问。

要激活该功能，请使用 seos.ini 文件的 [SEOS_syscall] 部分中的 proc_bypass 标记。 该标记存储每次 CA Access Control 必须访问进程文件系统时都要跳过的访问信息。

注意：有关 seos.ini 文件标记的详细信息，请参阅 *《参考指南》*。

绕过真实路径

搜索含有绝对文件路径（而不是相对路径）的文件将增加系统负载；绕过该搜索可加快文件事件的处理速度。

要激活该跳过功能，请将 seos.ini 文件 [SEOS_syscall] 部分中的 bypass_realpath 标记设置为 1。 如果您启用了此标记，则 CA Access Control 不获取真实文件名，例如，它可能是一个符号链接。

注意：有关 seos.ini 文件标记的详细信息，请参阅 *《参考指南》*。

重要说明！ 使用此功能时应非常小心，因为它会影响安全。如果使用相对路径访问文件，则一般规则不起作用。

绕过受托进程授权

通过 CA Access Control，您可以将程序定义为受托程序。 CA Access Control 在表中存储受托程序及其子程序。 无需完全跳过网络，即可访问与受托进程（及其相应端口）相关的所有（传入*和*传出）事件。

要指定这些程序，请使用 SPECIALPGM 类：

- 要绕过指定程序的文件和网络事件，请使用值为 pbf 和 pbn 的属性 PGMTYPE。
- 要绕过指定程序的 setuid 和 setgid 事件，请使用值为 surrogate 的属性 PGMTYPE。
- 要绕过指定程序的所有 CA Access Control 授权检查，请使用具有 fullbypass 值的属性 PGMTYPE。

  CA Access Control 会忽略具有 PGMTYPE(fullbypass) 属性的进程，且不会在 CA Access Control 审核、跟踪或调试日志中显示任何进程事件的记录。
- 要将绕过传播到从指定程序调用的所有程序，请将属性 PGMTYPE 与传播的值结合使用。

  注意：安全权限传播仅与 PBF、PBN、DCM、FULLBYPASS 以及 SURROGATE 权限一起使用。

跳过网络活动端口

要将与特定 TCP/IP 端口相关的所有连接事件（传入*和*传出）指定为无需 CA Access Control 授权就可以建立，您可以定义跳过这些端口。 跳过这些端口可以降低系统负荷并加速事件处理。 跳过的连接事件不会记录在审核和跟踪文件中。

注意：通过 CA Access Control，您仅可以跳过网络连接事件，而不可以跳过使用网络连接的所有后续事件（例如，打开文件）。

受托的传入连接是与传出连接分开指定的：

- 要跳过*传入*连接，请在 seos.ini 文件的 [seosd] 部分修改 *bypass_TCPIP* 配置设置。
- 要跳过*传出*连接，请在 seos.ini 文件的 [seosd] 部分修改 *bypass_outgoing_TCPIP* 配置设置。

注意：有关 seos.ini 初始化文件、更新标记和影响更改的详细信息，请参阅*《参考指南》*。

示例：跳过传入 Telnet 事件

如果您将 bypass_TCPIP 配置设置设为 23 （Telnet 端口），则当您通过 Telnet 登录*到*该工作站时，审核和跟踪文件将不再记录网络事件。 仍会进行记录与其他服务相关联的事件，例如 ssh、login、FTP 以及使用网络连接的后续事件（例如，打开文件）。

示例：跳过传出 FTP 事件

如果您将 bypass_outgoing_TCPIP 配置设置设为 21（FTP 端口），则当您从该工作站进行 FTP 时，审核和跟踪文件将不再记录该网络事件。 仍会记录与其他服务相关联的事件，例如 ssh、login、Telnet 以及使用网络连接的后续事件（例如，打开文件）。

减少审核和跟踪负载

CA Access Control 使用文件系统来保留审核数据和跟踪数据。 当 CA Access Control 写入此文件系统时可能会锁定系统中的大多数进程。 要缩短对文件系统的访问时间，请执行以下操作：

- 仅对您需要的资源和访问设置审核模式。
- 仅在您需要时打开跟踪。
- 在速度最快的可用文件系统上存储审核文件、跟踪文件和 CA Access Control 数据库文件。
- 在速度较快的文件系统上存储旁视数据库目录。

减少数据库负载

定义数据库规则的方式会影响系统性能：

- 针对常用目录的一般规则会产生许多验证，从而导致系统负载加重。

  例如，保护 /usr/lib/* 将导致 CA Access Control 检查系统上的每个操作。 要提高性能，请避免对常用的文件使用一般规则。
- 用户和资源的深层级结构要求系统负载获取并检查所有的依存关系。 要提高性能，请避免在数据库中使用深层级结构。

改进 PMDB 更新

策略模型将命令逐个发送给循环中各自的订户。 为了控制策略模型在每个循环期间向每个订户发送的最大命令数，请使用附录“pmd.ini 文件”的 [pmd] 部分中所述的 updates_in_chunk 标记。

如果增大此标记的值，则策略模型将使用较少的循环来发送命令。 在每次循环之后，策略模型都检查是否有新请求。 如果此标记设置较高，则策略模型不会像通常那样检查新请求。

例如，如果向策略模型中添加新订户（使用 sepmd -n 选项），请增大该标记的值，因为其他订户已经接收到策略模型发出的命令。 策略模型花费较少的时间将命令发送到其他订户，而花费较多的时间将命令发送到新订户，从而缩短了添加订户所花费的时间。

注意：不要将此标识值设置为 100 以上。

提高监视程序的性能

要减少系统负载，请将 Watchdog 后台程序 (seoswd) 设置为定期扫描而不是经常扫描安全的文件。 您可以将 Watchdog 指定为在系统负载较少时进行扫描。

要激活该功能，请使用 seos.ini 文件死亡 [seoswd] 部分中的 IgnoreScanInterval 标记，并为间隔时间和启动时间设置其他标记。

注意：有关这些标记的详细信息，请参阅*《参考指南》*中的 seos.ini 初始化文件。

改进类参数

使用 CA Access Control 的类激活和类授权功能来进一步提高性能。

类激活

CA Access Control 存储有关 CLASS 在数据库中是否活动的信息。 当 CA Access Control 启动时，它会将活动类的列表传递给 SEOS_syscall，因此 CA Access Control 无需经常拦截这些类。 只有在用户更改类的活动状态时 CA Access Control 才会拦截该类。 如果类处于停怠状态，则不会截获对资源的访问。

您可以对以下类使用停怠类绕过：FILE、HOST、TCP、CONNECT 和 PROCESS。

类授权

资源类 SEOS 控制 CA Access Control 授权系统的行为。 SEOS 类具有可修改的属性，这些属性指定类是否处于活动状态。 您可以禁用未使用的类（使用 setoptions 命令）以缩短授权时间。

解析名称

seos.ini 文件 [seosd] 部分中的几个标记（包括 GroupidResolution、HostResolution、ServiceResolution 和 UseridResolution）控制 CA Access Control 执行名称解析的方式。 设置这些标记可适当地提高性能。

您也可以创建一个后背数据库（不使用系统名称解析）。 要提高性能，请选择后备数据库选项。 此功能的标记包括 lookaside_path 和 use_lookaside。

注意：有关这些标记的详细信息，请参阅*《参考指南》*中的 seos.ini 初始化文件。

CA Access Control 必须执行 UID 到 username、GID 到 groupname、ipaddr 到主机名以及端口到服务的转换时，均可能会影响 CA Access Control 的性能。 CA Access Control 如何执行这些转换取决于 seos.ini 文件中某些标记的值，尤其是 under_NIS_server、use_lookaside、GroupidResolution、HostResolution、ServiceResolution、UseridResolution 和 resolve_timeout 标记的值。

当本机操作系统机制执行解析时，对系统性能的影响相对较小。 当将 ipaddr 转换为主机名时，外部机制（例如 DNS）必须执行该转换。 这可能会导致系统性能显著降低。 之所以会发生这种性能下降，是因为 seosd 在等待接收主机名时 CA Access Control 拦截的所有其他进程也必须等到 seosd 完成它的处理。

- 如果将 under_NIS_server 内标识的值设置为 no，seosd 会允许 UNIX 通过从以下源获取数据来转换 UID、GID、IP 地址及端口号：

工作站类型	源
独立	Seosd 使用下列文件进行转换； ■ /etc/passwd，用于 UID 到用户名的转换 ■ /etc/group，用于 GID 到组名的转换 ■ /etc/hosts，用于 IP 地址到主机名的转换 ■ /etc/services，用于服务端口到服务名的转换
NIS 客户端	信息源因操作系统及其版本号而异。 该信息通常来自 /etc 文件和 NIS 服务器。 但是，在一些系统中，/etc 文件不是源，而且执行转换的顺序在系统配置期间会有所更改。 例如，在 Solaris 2.x 系统中，文件 /etc/nsswitch.conf 确定转换顺序。

工作站类型	源
DNS 客户端	用户、组和服务的转换是使用 /etc 文件执行的。通过调用 DNS 服务器来转换主机名，在某些系统上还会读取 /etc/hosts 文件。
NIS 和 DNS 客户端	由 DNS 执行 ipaddr 到主机名的转换。 对于用户、组和服务转换，转换的执行方式与 NIS 客户端的转换方式相同。

- 如果将 under_NIS_server 标记的值设置为 yes，seosd 会执行它自己的转换。 如果 seosd 为它的转换缓存数据，则其数据源如下所示：

工作站类型	源
NIS 服务器	服务器计算机通常既充当服务器又充当客户端，并就任何类型的转换向 NIS 服务器后台程序咨询。 包含 NIS 解析映射源的文件通常位于 /var/yp 中，但是，根据站点配置以及操作系统的类型与版本，该位置可能会有所不同。
DNS 服务器	用于转换的信息源取决于站点的配置。 DNS 无法选择扫描它的解析数据库；因此，CA Access Control 无法使用缓存，而且必须使用后备数据库。 您必须配置该后备数据库，以便实用程序 sebuildla 使用主机列表文件。 有关详细信息，请参阅本章中的 sebuildla。
所有其他工作站	与 DNS 服务器相同。

在 CA Access Control 版本 2 和更高版本中，seosd 还可以使用标记 GroupidResolution、HostResolution、ServiceResolution、UseridResolution 和 resolve_timeout 控制转换进程。 有关这些标记的详细信息，请参阅 *《参考指南》*。

第 16 章： 使用 UNIX Exit

此部分包含以下主题：

UNIX Exit

UNIX exit 是一个指定程序，即一个 shell 脚本或一个可执行程序，在其他定义的 CA Access Control 活动发生的情况下会自动运行。 当加载或卸载 CA Access Control 内核模块时，或发出特定 selang 命令时，CA Access Control 支持使用 UNIX exit。 例如，您可以为添加的每个新用户运行初始化过程。

UNIX exit 可以在以下一个或多个情况下运行：

- 作为 pre-update exit，在更新*用户*或*组*记录的每个 selang 命令*之前*
- 作为 post-update exit，在更新*用户*或*组*记录的每个 selang 命令*之前*
- 作为 pre-load exit，在 *SEOS_load* 加载 CA Access Control 内核*之前*
- 作为 post-load exit，在 *SEOS_load* 加载 CA Access Control 内核*之后*
- 作为 pre-unload exit，在 *SEOS_load* 卸载 CA Access Control 内核*之前*
- 作为 post-unload exit，在 *SEOS_load* 卸载 CA Access Control 内核*之后*

用户或组记录更新 Exit

每当在 UNIX 环境中执行更新用户或组记录的 selang 命令时，都会调用 UNIX exit，无论使用命令行界面 (selang) 还是 GUI（例如 CA Access Control 端点管理）工具。

术语*更新*是指创建、修改或删除用户或组记录。 查询用户或组不会导致运行任何 UNIX exit。 以下为可能导致运行 UNIX exit 的命令：

- newusr
- newgrp
- chusr
- chgrp
- editusr

- editgrp
- rmusr
- rmgrp

从 UNIX 的角度来看，每个 exit 进程都作为 root 用户进程运行；但从 CA Access Control 的角度来看，其以代理身份 _seagent 运行。

提供的 selang exit 脚本的工作方式

CA Access Control 提供了一个脚本，您可以将其用作 master 脚本，根据当前 selang 命令的性质和状态来调用其他程序。 作为 CA Access Control 的一部分提供的 exit 脚本是 *ACInstallDir*/exits/lang_exit.sh（其中 *ACInstallDir* 是 CA Access Control 的安装目录。）下面说明它是如何工作的：

1. CA Access Control 将自动为该脚本的三个参数赋值。

参数	可能的值
CLASS	USER | GROUP
ACTION	CREATE | MODIFY | DELETE
STAGE	PRE | POST

这些参数指示 CA Access Control 正在处理用户还是组；是正在创建、删除还是修改用户或组；以及是将要执行 (PRE) selang 命令还是刚刚已经执行过 (POST)。

该脚本可以将参数值传递给它调用的程序。

参数	可能的值
EXEC_RV	接收用于确定 exit 命令是成功还是失败的 UNIX 命令的返回值。 对于 PRE 命令，该值始终为零。 对于 POST 命令，您可以使用该值决定是运行 exit 还是跳过 exit 命令。 有关如何使用此参数的示例，请参阅 *ACInstallDir*/samples/exis_src。

2. CA Access Control 使用 CLASS 和 STAGE 参数在相应目录中查找程序：

ACInstallDir/exits/USER_PRE/
ACInstallDir/exits/USER_POST/
ACInstallDir/exits/GROUP_PRE/
ACInstallDir/exits/GROUP_POST/

3. 在相应目录中，CA Access Control 将选择文件名以大写字母 S 开头、与相应操作相关并具有以下格式的所有程序：

   Snnaction_string

   其中 *nn* 是一个两位数的十进制数字，定义程序在执行顺序中的位置，*action* 是 CREATE、MODIFY 或 DELETE 命令之一，而 *string* 是描述字符串。

4. CA Access Control 将根据程序名中第二个和第三个字符的数字顺序运行所有相应程序。

示例：UNIX Exit 脚本

您要删除用户，且目录 *ACInstallDir*/exits/USER_PRE/ 包括以下文件：

- S10CREATE_precustom.sh
- S10DELETE_precustom.sh
- S99DELETE_prermusrdir.sh

当您发出命令要删除用户时，不会运行第一个程序，因为您要删除而不是创建用户。 将按照首字母 S 后的两个数字的顺序运行第二个程序，然后运行第三个程序。

可以传递给 selang exit 的参数

编写 exit 时，您可以利用前面提到的三个参数（CLASS、ACTION 和 STAGE）以及所有标准 CA Access Control 数据（例如名称和权限）。 您还可以指定专门供 exit 脚本使用的额外用户或组数据。 要为用户或组存储此类附加数据，可将其括在单引号中并定义为 newusr、chusr、newgrp 或 chgrp 命令中用户或组的 UNIX APPL 属性值。 例如：

```
chusr JONESY unix APPL('HIRED=MAY93,CLEARANCE=2')
```

exit 程序必须能够处理单引号之间的内容。

指定要运行的 selang Exit 程序

要告知 CA Access Control 运行哪些 exit 程序，请修改 seos.ini 文件的 [lang] 部分。 CA Access Control 向 pre-user、post-user、pre-group 和 post-group exit 提供 lang_exit.sh 脚本。 您还可以指定无 exit 或创建自己的 exit。

要指定自己的 exit，请按需要设置 seos.ini 的 [lang] 部分中的下列任一或全部设置。

注意：只有 exit 的完整路径名显示为 exit 标记的值时，才调用 exit。

示例：指定 selang Exit

在下例中，设置 seos.ini 文件标记，以便在组操作之前运行程序 groupcheck，在组操作之后运行程序 flag_exceptions，在用户操作之后运行程序 lang_exit.sh，而在用户操作之前不运行 exit 程序。 如下设置 seos.ini 文件标记：

```
[lang]
pre_group_exit = /opt/CA/AccessControl//exits/groupcheck
post_group_exit = /opt/CA/AccessControl//exits/flag_exceptions
post_user_exit = /opt/CA/AccessControl//exits/lang_exit.sh
```

超时和其他失败

Exit 执行在 15 秒后超时，除非 seos.ini 文件中的 exit_timeout 变量指定其他值。 非零返回值表示失败。

- 如果 *pre*-update exit 超时或返回的返回代码大于或等于 16，那么 CA Access Control 将终止 exit 进程、显示错误消息并放弃执行 CA Access Control update 命令。 任何其他正的返回代码都不会放弃执行该命令。
- 如果 *post*-update exit 超时或返回一个非零值，那么 CA Access Control 将终止 exit 进程并显示错误消息。 尽管已经执行 CA Access Control update 命令，该命令仍然有效。

selang Exit 示例

通过检查下列目录中的脚本，您可以熟悉建议的脚本编写技术。

ACInstallDir/samples/exits-src
ACInstallDir/samples/sample_exits

CA Access Control 内核加载程序 Exit

每当加载或卸载 CA Access Control 内核 (SEOS_load) 时都会调用 UNIX exit。 这样，您可以定义如何在加载或卸载 CA Access Control 内核时处理操作系统和第三方程序。 例如，运行 *SEOS_load -u*，您可以使用内核卸载 UNIX exit 来自动停止并在稍后重新启动防止 CA Access Control 卸载的进程。

对于某些操作系统，CA Access Control 附带一些即装即用的内核加载 exit 和/或内核卸载 exit。

注意：有关识别防止 CA Access Control 内核卸载的详细信息，请参阅*《参考指南》*中的 secons 实用程序。

内核加载 Exit 的工作方式

为了能够控制操作系统和第三方进程，CA Access Control 允许在加载 CA Access Control 内核扩展时自动调用 UNIX exit。

运行 **SEOS_load** 时，CA Access Control 将执行以下操作：

1. 查找以下目录中的程序：

   ACInstallDir/exits/LOAD

2. 选择具有以下格式的文件名的所有程序：

   SEOS_load_*string*.always

   其中 *string* 可以为任何描述性字符串。

3. 按照词典编纂顺序执行在目录 *ACInstallDir*/exits/LOAD 中找到的每个文件：

   SEOS_load_*string*.always -pre

   每个文件都用 *-pre* 参数执行，以便写入 exit 来检测该参数，并在加载内核之前执行必要的操作。

   注意：如果 exit 返回一个非零值，CA Access Control 将终止 exit 进程，显示错误消息，并中止内核加载。

4. 加载内核 (SEOS_syscall)。
5. 按照词典编纂顺序执行在目录 *ACInstallDir*/exits/LOAD 中找到的每个文件：

   SEOS_load_*string*.always -post

   每个文件都用 *-post* 参数执行，以便可以写入 exit 来检测该参数并在加载内核后执行所需的操作。

   注意：如果 exit 返回非零值，则 CA Access Control 将终止 exit 进程并显示错误消息。 CA Access Control 内核加载后一直保持加载状态。

内核卸载 Exit 的工作方式

为了能够控制操作系统和第三方进程，CA Access Control 允许在卸载 CA Access Control 内核扩展时自动调用 UNIX exit。

运行 **SEOS_load -u** 时，CA Access Control 将执行以下操作：

1. 查找以下目录中的程序：

   ACInstallDir/exits/LOAD

2. 选择具有以下格式的文件名的所有程序：

   SEOS_unload_*string*.always

   其中 *string* 可以为任何描述性字符串。

3. 按照词典编纂顺序执行在目录 *ACInstallDir*/exits/LOAD 中找到的每个文件：

   SEOS_load_*string*.always -pre

   每个文件都用 *-pre* 参数执行，以便写入 exit 来检测该参数并在卸载内核之前执行必要的操作。

   注意：如果 exit 返回非零值，则 CA Access Control 将终止 exit 进程、显示错误消息并中止内核卸载。

4. 尝试卸载内核。

   如果内核*无法*卸载：

   a. 选择具有以下格式的文件名的所有程序：

      SEOS_unload_*string*.opt

   b. 按照词典编纂顺序执行在目录 *ACInstallDir*/exits/LOAD 中找到的每个文件：

      SEOS_unload_*string*.opt -pre

      每个文件都用 *-pre* 参数执行，以便写入条件 exit 以检测该参数并在卸载内核之前执行其他可选的必要操作。

      注意：如果 exit 返回非零值，则 CA Access Control 将终止 exit 进程、显示错误消息并中止内核卸载。

   c. 卸载内核。

   d. 按照词典编纂顺序执行在目录 *ACInstallDir*/exits/LOAD 中找到的每个文件：

      SEOS_unload_*string*.opt -post

      每个文件都用 *-post* 参数执行，以便写入条件 exit 以检测该参数并在卸载内核之前执行其他可选的必要操作。

      注意：如果 exit 返回非零值，则 CA Access Control 将终止 exit 进程并显示错误消息。 CA Access Control 内核卸载后一直保持卸载状态。

5. 按照词典编纂顺序执行在目录 *ACInstallDir*/exits/LOAD 中找到的每个文件：

   SEOS_unload_*string*.always -post

   每个文件都用 *-post* 参数执行，以便可以写入 exit 来检测该参数并在加载内核后执行所需的操作。

   注意：如果 exit 返回非零值，则 CA Access Control 将终止 exit 进程并显示错误消息。 CA Access Control 内核卸载后一直保持未加载状态。

第 17 章： 与 LDAP 交互

此部分包含以下主题：

传输用户名称

如果您同时使用 CA Access Control 和 LDAP，则可使用自己编写的脚本在 CA Access Control 和 LDAP 之间传输用户名；提供了三个示例脚本。

重要说明！ 要设置 sebuildla 和所需的 LDAP 配置设置，您必须熟悉 LDAP 并能够执行 ldapsearch 命令。 建议您阅读 ldap(1)、ldapsearch(1) 的说明页面以及 LDAP 客户端文档中关于设置的信息。

其中两个提供脚本（ldap2seos 和 seos2ldap）可将整组用户从 CA Access Control 导出至 LDAP 服务器，并将它们从 LDAP 服务器导入 CA Access Control。

创建新的 UNIX 用户名后，第三个示例脚本 S50CREATE_Ldap_u.sh 会自动将这些名称从 CA Access Control 传输到 LDAP。

这些脚本示例要求访问 TCL shell 环境；它们使用语言客户端 API (LCA) 库扩展名 tcllca.so。

注意：有关 LCA 和 TCL 扩展名的详细信息，请分别参阅*《SDK 指南》*中的“语言客户端 API”一章和附录 LCA 扩展名。

如果您没有 TCL，请参阅 Larry Virden 每月发布到 comp.lang.t_c_l 的 FAQ，MIT 网站和 Terafirm 网站上提供该 FAQ。

您还可以访问 Sun 网站了解有关 TCL 的新闻、文档和资源。

S50CREATE_Ldap_u

S50CREATE_Ldap_u.sh 会在创建新 UNIX 用户时将其上载到 LDAP。

CA Access Control 提供了一个示例 shell 脚本，用于将新 UNIX 用户自动导入 LDAP 服务器。 您需要的脚本可能与该示例有所不同。

要部署示例 shell 脚本（假定您使用的是提供的 exit 脚本），请执行以下操作：

1. 将 S50CREATE_Ldap_u.sh 文件复制到 *ACInstallDir*/exits/USER_POST 目录。 在该目录下，脚本成为 post-user exit。
2. 在 seos.ini 文件的 [ldap] 部分中，将 base_entry 标记设置为 LDAP 基本条目。

   例如，对于位于加拿大的名为 ServerWorld 的组织，基本条目可能为：o=ServerWorld，c=CA。
3. 在同一部分中，将主机名设置为 LDAP 服务器的主机名。 将路径设置为 LDAP 基本目录。 （示例脚本在该目录下的 bin 目录中查找 line 命令实用程序。）

公用名称 (cn) 源自用户的全名。 例如，如果 CA Access Control 数据库仅包含用户名和姓，则该用户名和姓构成公用名称。 实际上将限制您只能使用公用名称，所以建议您不要根据用户名创建公用名称。

随后使用 selang 添加到 UNIX 中的每个用户都被自动上载到 LDAP 服务器。 如果 LDAP 中已经存在该用户，则会生成错误消息。

使用此脚本添加用户时，相关的 LDAP 将回复，如果存在该用户则警告将被收集到 /tmp/add_User2Ldap.tcl.log 文件中。 您可以使用 vi 或任何其他标准 UNIX 编辑器检查此文件是否存在错误。 每次添加新用户时，该文件都会被一组新的回复和警告覆盖。

第 18 章： 配置设置

通过 CA Access Control，您可以远程管理 CA Access Control 端点配置设置。 您可以使用 CA Access Control 端点管理 或 selang 配置环境执行此操作。

此部分包含以下主题：

配置设置

CA Access Control 在以下位置存储其使用的端点和策略模型配置设置：

- Windows 计算机上的 Windows 注册表
- UNIX 计算机上的初始化文件 (.ini)

注意：关于您可进行的配置设置及其含义的信息，请参阅*《参考指南》*。

更改配置设置

要影响 CA Access Control 和任何策略模型的工作方式，需要更改配置设置。

更改配置设置

1. 在 CA Access Control 端点管理 中，执行如下操作：
   a. 单击“配置”。
   b. 单击“远程配置”。
   将显示“远程配置”页面。
2. 在左侧“远程配置区”窗格中，按要求展开配置树，以展示包含您要修改的配置设置的部分，然后单击该部分。
   将显示“区域: *sectionName* 系统标记”页面，其中显示了所有的配置设置。
3. 按要求找到并编辑配置设置，然后单击“保存标记”。
   将保存更改的配置设置。

更改审核配置设置

要影响 CA Access Control 生成和存储审核记录的方式，您需要更改审核配置文件中的设置。 使用 selang 命令来更改审核配置文件中的设置。

更改审核配置设置

1. （可选）如果使用 selang 连接到远程主机，请使用以下命令连接该主机：

   ```
   host host_name
   ```

2. 使用以下命令移至配置环境：

   ```
   env config
   ```

3. 使用 editres config 命令按照需要修改配置设置。

   审核配置设置被更改。

示例：修改审核配置文件

以下示例将行添加到审核配置文件中：

```
er CONFIG audit.cfg line+("FILE;*;Administrator;*;R;P")
```

附录 A： NIS 配置

此部分包含以下主题：

安装说明

注意：本节补充了安装脚本中涵盖的材料。 本附录假设您熟悉网络信息系统 (NIS)、域名服务 (DNS) 和 UNIX 名称解析概念。

安装 CA Access Control 期间，您可以使用两个选项之一将用户 ID 解析为用户名、将组 ID 解析为组名、将主机 IP 地址解析为主机名，以及将服务端口解析为服务名：

- 使用系统函数，这些函数为系统中的 net cashing 后台程序定义绕过功能。
  - 如果您使用 Digital DEC UNIX，而它*不是* NIS 服务器，则默认情况下将使用系统功能进行名称解析。
  - 如果您使用 Digital DEC UNIX，而它*是* NIS 服务器，则安装将提示您选择以下两个选项之一：使用后备数据库或使用系统功能（可为 net caching 后台程序定义跳过功能）。
- 使用旁视数据库，该数据库是由 sebuildla 实用程序创建的。
  - 如果您使用的是配置为在 *NIS 服务器*上运行的 CA Access Control，请使用后备数据库。
  - 安装将在以下平台上默认使用旁视数据库：HP-UX 11.0 及更高版本、Sun Solaris 2.6 及更高版本、IBM AIX 5.1L 及更高版本，以及所有支持的 Linux 平台。

注意：在 IBM AIX 平台上，您必须使用后备数据库；没有使用系统功能的选项。

名称解析

CA Access Control 将拦截访问系统资源的请求并决定允许还是拒绝这些请求。 授权决定基于数据库中定义的访问规则和策略。 截获系统资源访问请求在内核等级中进行。

为了控制主机、组、用户及服务，内核及有关的系统调用将使用代码或数字（即：IP 地址、组 ID、用户 ID 及服务编号），而不使用名称。 CA Access Control 将根据名称定义访问规则。 CA Access Control 会将名称转换为内核可识别的代码。 这一过程称为名称解析。

在独立工作站（运行 Sun Solaris 2.5 或更高版本的工作站除外）中，名称解析是直接通过本地用户、组及主机文件（/etc/passwd、/etc/group 及 /etc/hosts）完成的。 当 CA Access Control 需要解析名称时，它只调用依次读取有关文件的系统功能。

然而，在较大型的网络中，该信息很少在本地存储。 当您使用 NIS 和/或 DNS 时，没有可供在名称解析期间查询的本地文件。 将请求该信息并通过网络从服务器接收。

NIS/DNS 客户端名称解析

CA Access Control 将按照以下方式在仅作为客户端的 NIS 或 DNS 工作站（不是其自己的服务器）中执行名称解析：

1. CA Access Control 将生成连接至相关服务器的网络请求。
2. CA Access Control 内核扩展将拦截该请求。
3. CA Access Control 内核扩展将允许该请求，因为它知道该请求是由 CA Access Control 进程在内部发出的。
4. 建立 NIS 或 DNS 服务器的连接，并检索名称解析所必需的信息。
5. 解析名称后，CA Access Control 将继续进程，决定允许还是拒绝原始访问请求。

对于 CA Access Control 而言，标准的 CA Access Control 配置足以在客户端服务器上轻松地处理名称解析。

服务器名称解析：死锁

CA Access Control 将按照以下方式在服务器（将其本身包括为客户端）上执行名称解析：

1. CA Access Control 将生成连接至相关服务器的网络请求。
2. 内核扩展将截获该请求。
3. 内核扩展将允许该请求，因为它知道该请求是由 CA Access Control 进程在内部发出的。
4. NIS 或 DNS 服务器（位于同一工作站中）将生成接受网络连接的请求。

5. 内核扩展将截获该请求。
6. 内核扩展知道 CA Access Control 进程尚未发出该请求。 它会将该请求置于等候 seosd 决策的请求队列中。
7. seosd 后台程序现在陷入一个死锁之中。 它正在等待完成名称解析所必需的答复，但是在 seosd 对提供该答复的进程授予接受网络连接的权限之前，该进程将无法继续。 第一个请求将生成第二个请求，并创建死锁。

Sun Solaris 名称解析：死锁

在 Sun Solaris 上进行名称解析需要访问 *nscd* 缓存。 nscd 是为最常见名称服务请求提供缓存的进程。 nscd 为 密码、组及主机数据库提供缓存。

缓存不是永久性的。 它将随着对密码、组及主机数据库进行更改或随着生存时间戳到期而变得无效。

Sun Solaris 安装程序可以创建一个死锁，就像上一节中介绍的死锁一样。 这时，CA Access Control 与 nscd 进程之间的交互将导致死锁。

1. 在名称解析期间，CA Access Control 将访问 nscd 缓存。
2. nscd 进程可以判断缓存是否太旧。 在这种情况下，它将尝试通过访问密码、组及主机数据库（位于本地或服务器）来刷新信息。
3. 访问这些数据库的请求将被内核扩展拦截。 由于 CA Access Control 进程没有发出请求，因此它将处于等候 seosd 决策的队列中。 但是不可能作出此这类决策，因为 seosd 仍在处理先前的请求。 第一个请求将生成第二个请求，并创建死锁。

避免死锁：旁视数据库

seos.ini 配置文件中的 under_NIS_server 标记的默认设置为 yes，可以避免死锁。 该标记将告知 CA Access Control 使用其各自的内部名称解析表，而不要使用 NIS、DNS 或 nscd 缓存。 除非指定其他位置，否则这些表将驻留在内存中。

CA Access Control 内部名称解析比 NIS 名称解析快得多，甚至比使用文件快；使用 CA Access Control 内部名称解析甚至在不存在死锁危险的环境中也可提高性能。

注意：后备数据库中没有内部名称解析表的缓存。 CA Access Control 使用打开文件句柄从表中读取数据。

在磁盘上存储解析表

CA Access Control 名称解析表是在 CA Access Control 启动时生成的。 该表应保存在磁盘中，而不应保存在内存中，因为在内存中存储会导致内存过载。 此外，当将信息读入内存时，它是静态的。 因此，CA Access Control 无法了解对用户、组或主机信息所做的任何更改。 在内存中更新表的唯一方法就是重新启动 CA Access Control。

为始终保持数据为最新数据，CA Access Control 提供了后备数据库，可确保内部名称解析表存储在磁盘中。

注意：要实施后备数据库，您需要使用 seos.ini 配置设置。 有关 seos.ini 配置设置的详细信息，请参阅 *《参考指南》*。

设置旁视数据库

后备数据库中的四个表为 userdb.la、groupdb.la、hostdb.la 和 servdb.la。 这四个表分别处理用户、组、主机及服务名称解析请求。 这些表位于由 seos.ini 文件中的 lookaside_path 标记指定的目录中，默认情况下该目录为 /opt/CA/AccessControl//ladb。

包含四个表的旁视数据库

要安装包含四个表的旁视数据库，请执行以下操作之一：

- 如果您要安装 CA Access Control，则请在询问您是否要创建后备数据库时回答“是”。
- 如果您已经安装了 CA Access Control：
  a. 在 **seos.ini** 的 [seosd] 部分中，将以下内标识设置为“是”：
     - under_NIS_server
     - use_lookaside
  b. 运行 sebuildla -a 以创建所有这四个表。

不足四个表的旁视数据库

您也可以创建一个、两个或三个表。 例如，如果您希望使用后备数据库仅解析主机，请完成以下步骤：

1. 安装 CA Access Control 之后，请在 seos.ini 文件的 [seosd] 部分中更改以下标记：
   - 将 under_NIS_server 设置为空。
   - 将 HostResolution 设置为 ladb。
2. 运行 sebuildla -h 以创建包含所有主机的表，包括本地和 DNS 主机。

   或

   运行 sebuildla -e 以创建仅包含本地主机（在 /etc/hosts 中定义）的表。

要创建包含其他表的旁视数据库，请使用 seos.ini 文件中的相应标记，然后运行包含 sebuildla 的相应选件。

注意：有关这些标记的说明，请参阅*《参考指南》*中的 seos.ini 初始化文件。 有关 sebuildla 的详细信息，请参阅*《实用程序指南》*。

重要说明！ 请您在每次添加主机时都运行 sebuildla。

旁视数据库的工作原理

后备数据库中的四个表（groupdb.la、hostdb.la、servdb.la 及 userdb.la）包含组、主机、服务及主机名的解析信息。 这些表位于由 seos.ini 文件中的 lookaside_path 标记指定的目录中，默认情况下该目录为 /opt/CA/AccessControl// ladb。

CA Access Control 内部名称解析比 NIS 名称解析快得多，甚至比查找 th 文件快。

实现旁视数据库

注意：此处概述的问题和解决方案仅供参考。 如果安装时的实际设置正确无误，则对于大多数用户来说无需执行任何操作。

下面从总体上概述了 CA Access Control 实施后备数据库的方式：

- 设置 seos.ini 文件中的相关标记。
- 定义 /opt/CA/AccessControl//exits 目录中的相关符号链接。
- 已发出命令 /opt/CA/AccessControl//bin/sebuildla -a 以便建立 lookaside 数据库。

sebuildla 实用程序将接进本机解析机制（如 /etc 文件和 NIS），从而建立旁视数据库。

后备表中不保留任何安全敏感信息（如密码、主目录的位置或 gecos）。后备数据库表仅包含数字 ID 号和名称。

创建后备数据库后，请使用 sebuildla 实用程序进行更新。 *无需*重新启动 CA Access Control。

更新主机旁视表

您必须更新主机后备表。 要执行该操作，请以固定的时间间隔执行 sebuildla -h（特定于站点）。 请使用 cron 作业执行此操作。

每次使用 selang 更改 UNIX 用户或组数据库时，都必须运行 sebuildla 实用程序。 CA Access Control 提供了这种用途的 exit 脚本，通过相应的参数运行 sebuildla。