计算机科学2001Vol.28№ 7

安全操作系统的标识鉴别机制研究与实现

Study and Realization of Identification & Authorization in Secure OS

袁 勋 卿斯汉

(中国科学院软件所 北京 100080)(中国科学院信息技术安全工程中心 北京 100080)

Abstract In secure operating system, mechanism of Access and Control is the kernel of security policy realization. As we know, Identification & Authorization is the base and precondition. It is absolutely necessarily in many kinds of secure OS evaluation home and aborad[1~3]. This paper will describe the theory bases and realization of the mechanism in detail and expand it to the network.

Keywords Access control, Security label, Identification, Authorization

1 背景

操作系统是计算机资源的直接管理者,操作系统的安全是整个计算机系统安全的基础。一般地说,安全操作系统指的是使用指定的安全特性,控制对信息的存取,使得只有适当授权的用户或代表他们工作的进程才有读、写、建立或删除信息的存取权。由这个基本的目标得出六项基本需求,分别是:安全策略,即必须有一个显式和良好定义的安全策略得以有系统实施;作标记,即存取控制标签必须对应于对象;标识,即每个主体都必须予以标识;审计,即审计信息必须有选择地保持并加以保护,使得影响安全的活动可被跟踪到应付责任的当事人;保证,即计算机系统必须含有可以被独立地评价的硬/软件机制,保证需求的实施;连续保护,即实现这些基本需求的可信任机制自身必须被保护,避免篡改和非授权改变。

根据美国国防部可信计算机系统评价准则(TCSEC,橘皮书),B2级以上的操作系统中,要求对所有的主体(如用户,进程等)和对象(如文件,IPC 客体等)采取强制存取控制机制(MAC)。具体来说,就是对系统内的所有主体,客体都附上许可标签或敏感性标签,用于约束和控制它们的行为,或以它们为操作对象的行为。

标识与鉴别机制正是作标记和标识的前提和基础。这一机制首先负责管理用户的安全级标签,包括创建、修改、删除等,在用户登录系统时通过一系列的安全性检查保证用户适当安全级的使用,并通过用户将安全级标签传递到进程,为强制存取控制机制的实施做好了前期准备。

2 依据和实现基础

根据 Bell-LaPadula[4]安全模型的要求,安全操作系统中的每一个客体都拥有一反映其信息安全性的安全级别和敏感性标签,每一个主体都拥有一标明其对信息的访问程度的许可标签。分别定义如下:

安全级别 系统用来保护该信息的程度。

敏感性标签 客体的安全级别的外在表示,系统利用此敏感性标签来决定一进程是否拥有对此客体的访问权限。

许可级别 当前进程(主体)的安全级别,用来决定此进程对信息的访问程度,故也将之称为进程的安全级别。

许可标签 当前进程的安全级别的外在表示,系统利用一进程的安全级别来决定此进程是否拥有对要访问的信息的相应权限。

通过标识与鉴别机制以及强存取控制机制,我们可以实现多级安全策略,即可以建立多个不同安全级别(类别,范畴)的分类信息,系统可以控制用户只能访问那些允许他访问的信息。安全级等级是分类的,而在一个分类中,又可以建立不同的范畴,每一个级别分类加上适当的范畴(classifications+categories)构成一个安全标签。级别是线性可比较的概念值,它的大小对安全级之间的比较是决定性的。范畴则代表一个集合,它表示在某一级别中所拥有的不同身份。范畴之间的比较只有包含关系。所以,安全级之间的关系包括三种:相等、分配、无关。相等是指两安全级的级别相等且范畴相等。当两安全级的级别一个大于等于另一个,并且其范畴集合包含另一个的范畴集合时,称前一安全级支配后一安全级。其余情况称两种安全级无关。

如果这标签是赋予客体的,就称之为敏感性标签,

表明此客体受保护的程度；而如果此标签是赋予主体的，就称之为许可标签(clearance label)，其中标签的类别部分用来告诉系统用户(主体)访问信息的可信级别以及用户的可信程度，而范畴部分是系统用来决定该用户是否属于信息对应的范畴。

敏感性标签和许可标签在组成上是相同的，只是一个相对于客体而言，另一个相对于主体而言。因此我们常常将之统称为安全标签或安全级别。

根据上面的定义，可以采取不同的强制性安全策略。以 CLASS(S)、CLASS(O)表示主体与客体的安全级，用＞＝表示安全级支配关系，＝表示安全级的相等关系，考察以下三种策略：

策略1：(1)if CLASS(S)＞＝CLASS(O)
then Read(S,O) or Execute(S,O)；
(2)if CLASS(O)＞＝CLASS(S)
then Write(S,O) or Append(S,O)；

该策略与 Bell-Lapadula 模型的简单安全特性及＊一特性要求一致，但它违背了"不向上破坏"的准则。

策略2：(1)if CLASS(S)＞＝CLASS(O)
then Read(S,O) or Execute(S,O)；
(2)if CLASS(S)＝CLASS(O)
then Write(S,O)；
(3)if CLASS(O)＞＝CLASS(S)
then Append(S,O)；

该策略解决了"不向上破坏"问题，但留下了隐信道，例如高安全级用户可以开启或关闭某一高安全级文件的写许可权，而低安全级进程可通过多次添加尝试获得关于该文件写许可权的信息。虽然可以采用"盲目添加"(总是返回成功)方法，但这是不合理的，因而也是不可取的。

策略3：(1)if CLASS(S)＞＝CLASS(O)
then Read(S,O) or Execute(S,O)；
(2)if CLASS(S)＝CLASS(O)
then Write(S,O) or Append(S,O)；

该策略比 Bell-Lapadula 模型的要求严格，同时避免了策略1、2中的弊端，是较为合理的策略。

3 标识与鉴别的实现

标识与鉴别的实现分为建立标识与鉴别文档、安全级管理、鉴别过程，下面分别介绍。

3.1 建立标识与鉴别文档

为了维持系统中的标识与鉴别信息，对不同的对象将采取不同的方式。对用户而言，系统将为所有用户建立安全文档，每个用户在文档中拥有一项，以记录格式存在。以 GNU C 为例，记录结构可以表示如下：

```
typedef unsigned int uid_t;
typedef unsigned int level_t;
typedef unsigned int audit_t;
struct Identification_and_Arthorization{
    char username[32];
    uid_t useruid;
    level_* user_levelset_pointer;
    audit_t userauditmask[256];
    ...
};
```

其中 username 是用户名；useruid 是用户 id 号；user_levelset_pointer 指针指向用户的安全级集合空间。它记录该用户的安全级范围，同时标明用户的缺省安全级；userauditmask 是用户审计事件掩码。

安全文档库以二进制文件形式存在，包括库文件和索引文件以及单独的以用户名为文件名的安全级别文件。所有这些安全信息文件的自身安全级都是高度受保护的，处于系统中除了特权用户外不可随意更改、替换的安全级区域内，这样他们自身的安全就得到了保证，而他们的维护由系统中特定的管理员来负责，其中将使用有关的特权命令。

文件的安全级信息保存在文件的索引节点(inode)中。由于文件在任何时候只拥有唯一的安全级，所以只需在 inode 结构中扩充一项指示当前该文件的安全级即可。同理，只有系统中特权用户才能修改文件的安全级。

设备虽然在安全操作系统中也视为文件，但与一般文件不同的是，设备是公共资源，将被众多不同的用户使用，所以应具有一个安全级集合，以表征在一定安全级范围内访问者才有权使用。而普通文件都所属于唯一的用户，无论是系统用户或者一般用户，他们只具有该用户创建该文件时的安全级。所以，为设备也专门建立安全文档，作为一项存储在设备档案中。访问者的安全级只有被设备最高安全级支配，并且支配设备最低安全级时，访问才被允许。

进程、消息队列、信号量集合和共享存储区等动态的对象，安全级在这些对象产生时由其创建者直接加入他们相应的数据结构中，保存在内存中。

3.2 安全级管理

3.2.1 用户和进程的安全级 用户的安全级是系统管理员在创建用户时设置。在用户的生存周期内，可以添加、删除、修改其安全级范围以及缺省安全级，以反映该用户的安全级别的变动。用户可具有多个安全级，以不同安全级登录系统的用户将能读、写不同的文件，使用不同的设备。

进程的安全级是在用户登录后创建进程时确定，这个安全级在其代表用户的安全级范围内，在进程的生存周期内不可以改变。

3.2.2 系统设备的安全级 为每个系统设备也建立安全文档，说明它的安全属性。它同用户安全级一样是一个范围，并且可以更改。

3.2.3 用户创建客体的安全级 用户创建客体的安全级具有唯一性，它的确定和赋值，是根据客体的类型按以下组织规则进行的。

a. 文件、特别文件(设备)、有名管道的安全级。文件、有名管道的安全级为创建该客体进程的安全级，且客体的安全级必须等于其父目录的安全级，除非其父是一个将被多种安全级主体访问的多级目录。

b. 目录的安全级，每个目录同普通文件一样，在它们的生存周期内具有一个安全级，所不同的是目录的结构须满足兼容性。也就是说，一个进程创建一个目录，目录的安全级即为创建其进程的安全级，且目录的安全级须大于或等于其父目录的安全级。

为了使系统能够正常进行，当多个拥有不同安全级的进程将访问同一个目录时，我们设置此目录为多级目录。在多级目录下，进程下仅可以创建等于该目录安全级的文件，还可以创建高于该目录安全级的文件。对于多级目录，它使用户只能看到低于或等于其进程安全级的目录项，以防止隐通道。

c. 进程、消息队列、信号量集合和共享存储区，这组类型的客体不具有文件系统表示。利用相关的系统调用创建客体时，客体的安全级为创建其进程的安全级。

3.3 鉴别过程

鉴别机制用于保证只有合法用户才能存取系统中的资源，用户在创建伊始由用户管理员指定其拥有的许可性标签，在其生存周期间可以修改，并在取消用户时删除。用户登录系统时，在一般的非安全系统的用户管理和登录的基础上，不仅检查用户的登录名和口令、赋予用户唯一标识用户号(uid)、组号(gid)，还检查用户的安全级，赋予用户进程安全级和其他安全特性。具体流程图如图1。

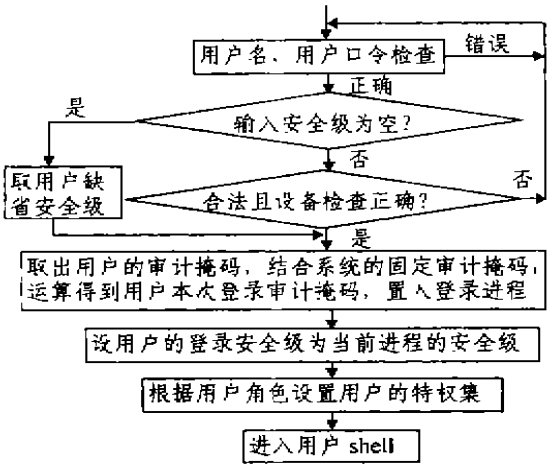

图1

用户登录时，选择输入安全级。标识与鉴别机制检查该安全级是否在安全文档中定义的该用户安全级范围之内。若是，则认可，否则拒绝该次登录。若用户没有选择安全级，系统从该用户的安全文档中取出其缺省安全级作为本次登录安全级。对于本次输入的合法安全级，系统将设为该用户下一次登录的缺省安全级。

当标识与鉴别机制确认用户的登录名口令和安全级后，便允许用户登录，登录后，用户本次使用的安全级将作为许可标签置入用户进程，并在这次登录过程中保持不变，以约束用户的行为。系统审计模块，将根据用户自身拥有的审计掩码和系统的审计掩码，经过一定的计算形成用户进程的审计掩码，设置相应的审计特性以对该用户本次登录的一定行为进行审计，除此之外，用户登录安全级还必须在其登录设备允许的安全级范围内。通过设备安全模块的检查后，用户登录安全级将被设为该设备的工作安全级。特权管理模块也将在这里为用户设置一定的特权，如果用户是某一管理角色，他将具有某些超越其他安全检查的特权，若用户只是普通用户，它的特权将设为最小。这样用户就不可能超越他拥有安全属性进行不允许的操作了。

4 扩展

在本地操作系统上实现的标识与鉴别机制还可以扩展到网络服务，例如 Telnet 服务，Ftp 服务等。我们将对为网络用户设置的安全文档进行扩充，增加对用户登录资格、用户远程登录设备、登录源主机、登录源 IP 地址、网络服务使用端口等网络特性的控制，对用户身份进行全面标识。这样就可以确保登录系统的用户来自一台合法主机，并以合法安全级使用合法服务。

结论 标识与鉴别是安全操作系统的重要组成部分，是实现所有安全机制的基础与前提。标识与鉴别机制以及安全标签的使用，使得用户只能在一定安全级别范围的许可内进行操作，同时结合了其他存取控制机制，使安全策略在安全操作系统中得到精确的实现。

参考文献

1 Department of Defense. Trusted Computer System Evaluation Criteria. DoD5200. 28-STD, Washington, DC, Dec. 1985

2 The International Organization for Standardization, Common Criteria for Information Technology Security Evaluation ISO/IEC 15408:1999(E), 1999

3 国家质量技术监督局. 计算机信息系统安全保护等级划分准则. GB 17859-1999

4 Mayer F L. An Interpretation of a Refined Bell-La-Padula Model for the Mach Kernel. 1988

5 Managing Security on the Trusted DG/UX™ System. AVLLON PRODUCT LINE, 1994

6 刘文清，等. 基于 Linux 开发安全操作系统的研究. 计算机科学，2001，28(2)