在 LINUX 核心实施强制访问控制*)

Enforcing Mandatory Access Control in Linux Kernel

张志文　周明天

(电子科技大学计算机学院　成都610054)

Abstract　The widespread use of computers and networks has emphasized the necessity for secure operating systems. The security research in operating system has shown that mandatory access control provided by operating system is essential for supporting secure applications. This paper presents the details of enforcing mandatory access control in Linux kernel, and the future work.

Keywords　Mandatory access control, Secure operating system, Linux

1. 引言

由于计算机和网络的广泛使用,迫切需要安全操作系统来作为其基础平台。为了确保计算机处理、存储信息的机密性,完整性和有效性,必须依靠高安全级的操作系统。

早期安全操作系统的研究在理论上取得了重大的成果,它们主要基于安全核的思想来构造安全操作系统,比如 DEC 公司的安全核,加利福尼亚大学的 Data Secure Unix 及 Multics 的安全核等,但因缺乏相应的应用支持而不能得到推广应用;近年来多采用操作系统安全增强技术,在一个成熟的 C2级商品操作系统基础上,通过对安全机制的再设计,将原有操作系统源码进行修改,形成新的高安全级别如 B1安全级的操作系统。

国内对安全操作系统的研究也是近年来的热点。利用当前开放源码的 LINUX,使得我们有机会构造自己的主流高安全级别的操作系统[2]。

我们已经研究了在 LINUX 核心中增强 DAC 访问控制机制[2]、系统审计机制、最小特权机制,为了进一步提高 LINUX 操作系统的安全性,使之达到 B1安全级,以满足高安全性的应用环境,要求操作系统支持强制访问控制(MAC)。本文详细介绍如何在 LINUX 核心中设计与实现强制访问控制(MAC),进一步增强 LINUX 的安全性。

2. 强制访问控制(MAC)

传统的 UNIX 模型允许所有用户共享所有系统资源,对象的拥有者可以指定其访问模式。此模型对低安全级要求的系统是满足的,而对高安全级要求的系统是不能接受的。

因此要对文件系统对象、进程及其他系统资源增加新的安全属性,此新的安全属性称为敏感标签。敏感标签能用于限制用户或进程对文件系统对象及其他对象的访问模式,由于敏感标签不能由用户或进程更改,系统安全策略的实施不依赖于用户的意愿,用敏感标签实施访问控制称为强制访问控制(MAC)。

2.1　MAC 机理

Bell &Lapadula 模型[11~13]使用数学符号和集合论定义了安全状态的概念、访问模式及授予访问的规则。模型用主体(Subject),客体(Object)来描述一个访问。一个实体根据其访问类型的不同,既可以是主体也可以是客体。

2.1.1　MAC 标签　MAC 标签由安全敏感级别(senesitive label)和类别集合(category)两部分组成。安全敏感级别具有严格的顺序关系(例如:绝密、机密、秘密、公开),类别集合没有顺序关系,相当于部门(例如:市场部、财务部、开发部等)。安全敏感级别和类别集合构成序偶对(S,C),这个序偶对表示一个 MAC 标签。

2.1.2　MAC 标签之间的关系

1)支配(dominate):若有两个 MAC 标签 labelA 和 labelB,如果 labelA 的安全敏感级别大于等于 labelB 的安全敏感级别,且 labelA 的类别集合包含 labelB 的类比集合,则称 labelA 支配(dominate)labelB;

2)相等(equal):若有两个 MAC 标签 labelA 和 labelB,如果 labelA 的安全敏感级别等于 labelB 的安全敏感级别,且 labelA 的类别集合等于 labelB 的类别集合,则称 labelA 与 labelB 相等;

3)不相交(disjoint):若有两个 MAC 标签 labelA 和 labelB 是不可比的,则称 labelA 与 labelB 不相交。

2.1.3　安全属性　Bell &Lapadula 模型由两个安全属性组成:

1)简单安全性:主体(Subject)允许从客体(Object)读,当且仅当主体(Subject)安全标签支配客体(Object)的安全标签;

2)*—特性:主体(Subject)允许向客体(Object)写,当且仅当主体(Subject)的安全标签受客体(Object)的安全标签支配;

这两个限制试图保证没有信息从高安全级别的客体(Object)流向低安全级别的主体(Subject),因此可以防止信息从高安全级流向低安全级。

2.1.4　系统安全状态　设系统中有主体集合 S,客体集合 O,访问模式集合 A,敏感级别 L。对客体有读访问的主体,可以读和拷贝客体内容;有写访问的主体,可以修改客体内容。因此系统可以用有限状态机来描述,每个状态 v=(b,f)是 V=(B×F)的元素,这里 B 为(S×O×A),当前系统可能的访问集合 F 为 $L^t \times L^o$,是一序偶对(fs,fo),函数 fs 返回主

*)本文的工作受到国家计委重大项目的支持。张志文　博士生,主要研究领域为操作系统安全、网络安全。周明天　教授,博士生导师,主要研究方向为计算机网络,网络与信息安全,并行分布处理,分布对象技术。

体的MAC标签，函数fo返回客体的MAC标签。请求集合由R表示，状态转换函数T为R×V->V，将系统从一个状态转换到另一个状态。系统Σ(V,T,Vo)是由状态V转换函数T及初始状态Vo组成的有限状态机。

我们说一个状态是简单安全的，当且仅当状态中对客体具有读访问的主体，其MAC标签支配客体的MAC标签；一个状态满足*--特性，当且仅当状态中主体可读客体的MAC标签，受主体可写的客体的MAC标签支配。

一个系统状态是安全的，当且仅当所有的可达状态满足简单安全性和*一特性。

2.1.5 校验规则 校验规则的目的是确保系统的状态转换是安全的。当前主体Si对客体Oj以方式X访问，可以看作是三元组(Si,Oj,X)，所有这些三元组构成当前的访问集。对当前访问(Si,Oj,X)，当且仅当满足：

1)简单安全特性；

2)*一特性；

3)自主访问控制(DAC)；

时，才被授予；否则拒绝。

3. MAC在LINUX核心中的设计与实现

操作系统的安全级别越高，越难以使用和管理，我们在LINUX核心中设计与实现安全机制时，其首要目标是考虑它的易用性与兼容性。另外，尽可能减少对系统性能的影响。对系统中没加MAC标签的进程或文件，其行为与运行在通用C1级或C2级的LINUX操作系统相似，不影响其程序的执行和对文件的访问，实现了向下兼容。

3.1 MAC信息存放

要实现MAC，首先要解决的问题是MAC信息的存放。为了尽量减少对系统性能的影响，对客体(如文件，目录)的MAC信息采用分布式存放，即对文件，目录，采用一个专用的磁盘块来永久存放MAC信息；在核心中，通过扩展INODE的数据结构，动态存放MAC信息。打开文件时，在读入INODE时，从磁盘块中读入MAC信息，存放在INODE数据结构中，这样，只增加读一次磁盘块的开销，以后就可以直接使用核心INODE的MAC信息，可以最大限度地减少对系统性能的影响。对主体(指进程)的MAC信息通过修改进程PCB来存放，用户登录系统或执行程序时，就设置相应的MAC信息，在进程的生命期内，可以直接访问进程的MAC信息，进一步减小对系统性能的影响。

3.2 MAC核心数据结构

在MAC的核心实现中，增加了几个主要的数据结构，用于处理MAC的访问控制，即kern_label_t，kern_cat_t，和kern_lev_t，核心中的一个MAC标签由kern_cat_t和kern_lev_t组成，可以分别对安全敏感级别和类别集合进行操作。它们在核心中的关系如图3-1所示。

3.3 软件结构

在核心中完成MAC机制的模块主要有：MAC管理模块和MAC策略模块。

3.3.1 MAC策略模块 MAC策略模块由一个函数集构成，它们是MAC的安全策略函数。向MAC管理模块提供服务。为了保证系统的安全，每当主体引入客体或者向客体读/写信息等事件发生时，都要引起系统状态的改变，需要进行特定的校验，如果系统允许某事件发生，则该事件发生后，将使系统转移到一个新状态。MAC策略模块就是确保系统的状态转换是安全的。

3.3.2 MAC管理模块 MAC管理模块分散在系统调用当中，其主要功能是监控每个事件，获取适当的安全信息，调用MAC策略校验事件，为事件授权。由用户或进程发出的系统调用，如果其中有影响系统安全状态的事件，则MAC管理模块必须根据获取的安全信息决定调用哪个安全策略函数来校验这个事件，检验事件是否合法，如果合法，就意味着受检事件不会使系统向非安全状态转移，让其执行；否则，就拒绝该事件发生。

以上模块之间的关系如图3-2所示。

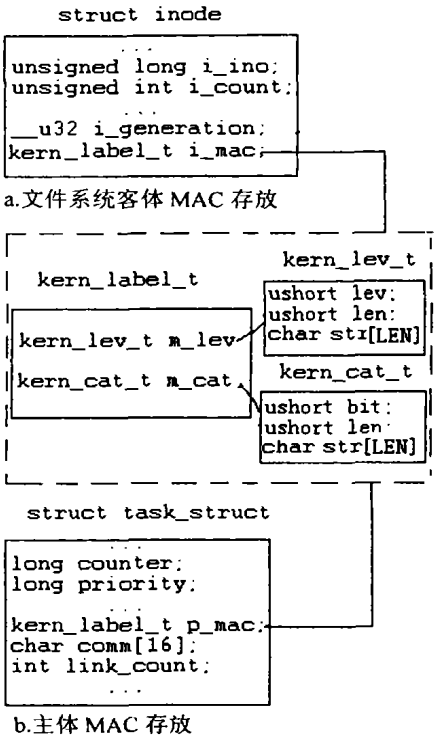

图3-1 核心中主体和客体的MAC表示

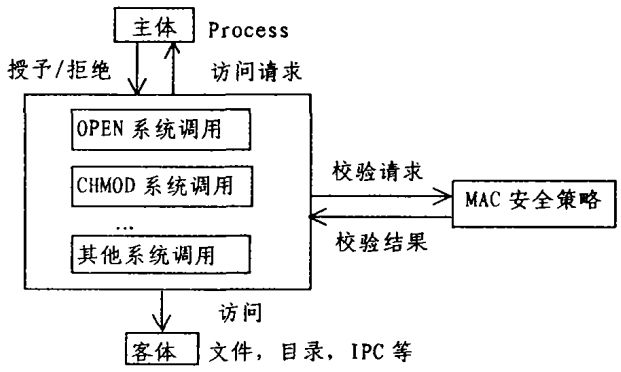

图3-2 MAC机制实现

4. MAC核心数据初始化

当前所开发的系统的强制访问控制是在系统启动时由核心模块初始化的，通过直接操作磁盘块完成，其目的是更好地控制系统的安全性，减少系统管理开销。通过在核心中动态建立两张表，使用HASH技术，来确定当前系统所支持的MAC敏感级别和类别集合，可以大大地减少系统管理的复杂性，例如，如果系统中有某个部门要撤消，只需修改系统所支持的

MAC 类别集合就可以了，而不用去修改每个具有此 MAC 类别集合的文件、目录；如果系统已经不支持低级的安全敏感级别，只需将处于系统不支持的低安全级的文件、目录或进程的安全敏感级别映射成系统当前最低的安全级别即可，而无需一一修改。MAC 核心数据的初始化过程如图4-1所示。

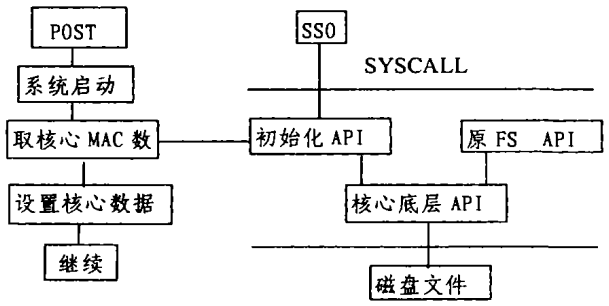

图4-1 MAC 核心数据初始化

结束语 Bell&LaPadula 模型只考虑了系统的机密性，而没有考虑数据完整性和有效性的保护。例如，如果没有自主访问控制，最低安全级别的主体可以删除其类别集合的所有数据，因此，MAC 机制一般不单独使用。另外，MAC 还不足以充分保证系统的安全性，信息泄露有可能通过间接路径（如隐蔽通道）发生。

对 LINUX 系统的安全增强，国内外已有的研究工作尚不尽人意，它们或者是兼容性不太好，或者是系统开销过大。我们的系统充分体现了与已有系统应用的兼容性，安全管理的易用性，以及系统运行的高效性，下表以系统调用 OPEN 为例，示出了 MAC 开销。目前，我们的系统已经通过了 B1安全级的初测试。进一步的工作将研究计算机网络上的多级安全策略，与基本操作系统一起构成适合于网络应用的完善的安全操作系统产品。

连续访问次数(次)	100	1000	10000
无 MAC(us)	314	2969	29676
有 MAC(us)	344	3250	32489
相对百分比(%)	109.51	109.45	109.48

环境：CPU PIII 800；Kernel linux-2.4.4

参 考 文 献

1 刘文清，刘海峰，卿斯汉. 基于 Linux 开发安全操作系统的研究. 计算机科学，2001，28 (2)

2 张志文，周明天. 用内核 ACL 增强 LINUX 的安全性. 计算机科学，2002(1)

3 Rusling D A . The Linux Kernel. January 1998

4 IEEE Draft P1003.1e. IEEE Standard Department，1997

5 Remy Card . Linux File Systems. Brueselles Oct. 1994

6 Louis-Dominique Dubeau. Ext2 file system. 1994

7 陈爱民，于康友，管海明. 计算机的安全与保密. 电子工业出版社，1992

8 Bach M J. The design of Unix Operating System. Prentice-Hall，Inc. 1986

9 Trusted Computer System Evaluation Criteria. Department of Defence U.S.A. 1985. DoD 5200.28-STD

10 Secure Computer System: Unified Exposition and Multics Interpretation. The Mitre Corporation . March 1976

11 Bell D E，LaPadula L J . Secure Computer Systems: Mathematical Foundations: [MITRE Technical Report 2547]. Volume I . March 1973

12 Bell D E . Secure computer system: A refinement of the mathematical model. Hanscom AFB. BEDford. MA. REP: [ESD-TR-73-278]. vol. 3ESD/AFSC，1973

13 Padula L J L，Bell D E. Secure Computer Systems: Mathematical Foundations and Model. M74-244，The MITRE Corporation，BEDFORD，Massachussetts，Oct. 1972

14 Mclean J . The Specification and Modelings of Computer Security. Naval Research Laboratory Washingten，D.C. 20375

15 Osborn S. Mandatory Access Control and Role-Based Access Control. Department of Computer Science The University of Western Ontario London，Ontario，Canada NCA-587

（上接第139页）

长度编码(Run length code)等。之所以要使用无损压缩，原因在于从有误差的密文很难正确地恢复出原文。

另外，为了使加密系统更加难以破译，还可以采用超混沌系统或有时延的混沌系统来生成随机序列，由于它们有多个正的李亚普诺夫指数，导致序列更加难以预测。

总之，混沌理论用于信息加密是一个很有应用前景的研究领域，如何将传统的加密算法、思想与混沌理论相结合，仍然是目前主要的研究方向，如混沌密码体制、混沌数字水印等。如文[10]对混沌映射作为流加密算法进行了较为深入的研究，具有一定的指导作用。

参 考 文 献

1 Pecora L M，Carroll T L. Synchronization in chaotic systems. Phys. Rev. Lett，1990，64(2)：821～824

2 Morgül Ö，Feki M. A chaotic masking scheme by using synchronized chaotic systems. Phys. Lett，1999，A 251 (3)：169～176

3 Liao T-L，Tsai S-H. Adaptive synchronization of chaotic systems and its application to secure communications. Chaos，Solitons & Fractals，2000，11 (9)：1387～1396

4 Murali K. Heterogeneous chaotic systems based cryptography. Phys. Lett，2000，A 272 (3)：184～192

5 Baptista M S. Cryptography with chaos. Phys. Lett，1998，A 240 (1-2)：50～54

6 Alvarez E，et al. New approach to chaotic encryption. Phys. Lett，1999，A 263 (4-6)：373～375

7 Alvarez G，et al. Cryptanalysis of a chaotic encryption system. Phys. Lett，2000，A 276 (1-4)：191～196

8 Li Shujun，Mou Xuanqin，Cai Yuanlong. Improving security of a chaotic encryption approach. Phys. Lett，2001，A 290 (3-4)：127～133

9 Schneier B. [美] 著，吴世忠，祝世雄，张文政等译. 应用密码学——协议、算法与 C 源程序. 机械工业出版社，2000

10 Kocarev L，Jakimoski G. Logistic map as a block encryption algorithm. Phys. Lett，2001，A 289 (4-5)：199～206