一种用于操作系统安全内核的多级分层文件系统的研究与实现

刘文清, 卿斯汉, 刘海峰
(中国科学院软件研究所信息安全技术工程研究中心, 北京 100080)

摘 要: 本文对一种用于操作系统安全内核的多级分层文件系统, 给出了其安全策略和关键技术, 并对其安全性、兼容性和效率进行了分析, 该多级分层系统已工程实现于我们开发实现的安全操作系统 SecLinux 中.

关键词: 多级分层文件系统; 强制存取控制; 安全内核; 安全操作系统
中图分类号: TP309 文献标识码: A 文章编号: 0372-2112 (2002) 05-0763-03

Study and Implementation of a Multilevel File System in Security Kernel of OS

LIU Wen-qing, QING Si-han, LIU Hai-feng
(*Engineering Research Center for Information Security Technology, The Chinese Academy of Science, Beijing* 100080, *China*)

Abstract: This paper presents a multilevel file system applicable to security kernel of operating system. It provides security policies and key techniques as well as analysis of security, compatibility and efficiency. The file system has been implemented in the secure operating system "SecLinux".

Key words: multilevel file system; MAC; security kernel; secure operating system

1 引言

多级安全强制存取控制是 GB17859-1999 第三级、第四级的特征标志, 也是我国当前安全操作系统研究的首选目标. 但一般的树型结构平面文件系统不易于实现多级安全强制存取控制, 也不易于限制隐蔽存储通道[1]. 因此, 在开发操作系统 SecLinux 的安全内核时, 设计了一种多级分层文件系统. 安全内核技术是目前安全操作系统设计的唯一最常用方法, 但它只是一个笼统的概念, 工程实现的工作量、难度和复杂度都十分巨大. 多级分层文件系统则是我们自主开发 SecLinux 操作系统安全内核的一个重要组成部分, 它基于 BLP 安全模型[2~4]、MBLP 安全模型[5], 并紧紧结合了 Linux 的体系结构和文件系统结构. 本文首先给出了该多级分层文件系统的具体存取控制策略, 然后面向 Linux 的文件系统结构, 详细给出了如何加入这些存取控制策略, 以及加在哪里以构成完备的存取控制点, 最后讨论了所构成的多级分层文件系统的安全性、兼容性和效率. 另外, 为实现本文多级分层文件系统, 引用了多级目录、隐蔽文件名等概念.

2 存取控制策略

多级分层文件系统的存取控制策略是以 BLP 安全模型、MBLP 安全模型的简单安全公理、* -特性公理、兼容性公理和激活性公理为基础的. 具体包括:

①若主体读(*r*)或执行(*x*)访问客体, 主体的安全级必须支配客体的安全级, 记为 *R*1;

②若主体写(*w*)访问客体, 主体的安全级必须等于客体的安全级, 记为 *R*2;

③若主体创建文件类型的客体时, 客体的安全级必须等于其所在父目录的安全级, 记为 *R*3;

④若主体创建目录类型的客体时, 客体的安全级必须支配其所在父目录的安全级, 记为 *R*4;

⑤若主体创建客体时, 新客体的安全级等于主体的安全级, 记为 *R*5;

⑥若主体删除客体时, 主体的安全级必须等于客体的安全级, 记为 *R*6;

⑦若主体搜索(*x*)一路径名, 主体的安全级必须支配路径名中每一个目录分量的安全级, 记为 *R*7.

⑧若主体列表(*r*)一目录下的文件或目录时, 主体的安全级必须支配文件或目录的安全级, 记为 *R*8.

多级分层文件系统的存取控制策略具体实现在文件系统的每个系统调用中, 即在 open()、creat()、read()、write()、mkdir()、rmdir()、link()、unlink()、stat()、rename() 等系统调用中分别加入相应的存取控制策略, 如表 1 所示.

系统调用是用户程序进入内核, 存取系统资源的唯一入口, 对文件系统的每个系统调用都基于存取控制策略进行检查, 就等于控制了用户对文件的存取.

由于这些检查处于核心态, 是完全与用户隔离的, 即安全内核对用户的访问请求进行的存取控制判定, 是不受用户干扰的, 这就使得多级分层文件系统的存取控制机制是完备的、不可绕过的.

收稿日期: 2001-03-06; 修回日期: 2001-11-26
基金项目: 国家自然科学基金(60083007); 国家重点基础研究发展规划资助项目(G1999035810)

表 1 多级分层文件系统中系统调用的存取控制策略

系统调用	存取控制策略	备 注
open( )	R7, R5, R2, R3, R1	具体控制策略随存取模式和文件是否存在而不同
creat( )	R7, R5, R2, R3	当要创建的文件已存在时, 用 R3 进行隐蔽通道检查
read( )	无	在 open( )、creat( ) 中已做了相应的安全检查, 可以保证“支配读”
write( )	无	在 open( )、creat( ) 中已做了相应的安全检查, 可以保证“相等写”
link( )	R7, R6, R5, R3	
unlink( )	R7, R6	客体的安全级要支配主体的安全级, 以避免隐蔽通道
execve( )	R7, R1	
chdir( )	R7, R1	
fchdir( )	R1	
chmod( )	R7, R2	主体必须是客体主或有 OWNER 特权
fchmod( )	R2	主体必须是客体主或有 OWNER 特权
chown( )	R7, R2	主体必须是客体主或有 OWNER 特权
stat( )	R7, R1	
fstat( )	R1	
lseek( )	无	在 open( )、creat( ) 中已做了相应的安全检查, 可以保证“支配读、相等写”
access( )	R7, R1, R2	具体控制策略随 fmode 而不同
rename( )	R7, R3	
Mkdir( )	R7, R4, R2	当要创建的目录存在时, 用 R2 进行隐蔽通道检查
rmdir( )	R7, R2	
readdir( )	R8	见 3.3 隐蔽文件名的实现
等等		

3 关键技术

3.1 文件、目录安全级信息的存放

在多级分层文件系统中, 每个文件和目录的安全级一般都统一放在文件和目录的 I 结点结构中, 与相应文件或目录在一起, 这样利于管理和存取.

其中, 对于内存索引结构 inode, 可在其中增加一个结构 i_sec, 用于存放这类客体的安全级信息. 由于其仅存在于内存, 对它的扩充和修改, 一般不会引起兼容性问题.

在外存(磁盘)中如何存放这类客体的存取控制信息, 是一个值得讨论的问题. 一般有以下两种方法可供选择:

(1)开发专用安全文件系统, 扩充磁盘 I 结点结构, 存放这类客体的存取控制信息. 如 USL UNIX SVR4.2 中就是采用了这种方法, 它新开发了一个新的安全文件系统——sfs 文件系统.

(2)在现有文件系统(如 ext2)的磁盘 I 结点结构的基础上, 扩充存放安全级信息.

显然, (1)中方法工作量较大, 并可能会影响系统效率. (2)中方法存在局限性, 运用不合理可能会带来兼容性问题.

3.2 多级目录

根据存取控制策略 R3, 多级分层文件系统不允许不同安全级的用户, 在同一目录下创建不同安全级的文件, 但系统中不同安全级的用户有时运行程序(如 *vi*、*cc*)时, 就需要在同一目录(如/tmp、/usr/tmp)下创建临时文件. 如果允许在同一目录下创建不同安全级的文件, 就会破坏存取控制策略的限制或者要求这些程序都是可信的, 并会产生隐蔽通道, 这是不可接受的. 因此, 我们在多级分层文件系统中引入了多级目录的概念.

• 多级目录的结构

多级目录含有一些特殊的子目录, 称为有效目录(Eff. Dir). 有效目录是当某个进程第一次访问多级目录时由安全内核自动创建的, 对用户是透明的. 有效目录名即是与该进程相关的安全级标识(Level). 如图 1 所示, /tmp 为一个多级目录.

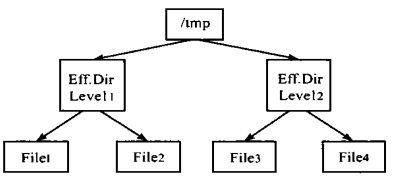

图 1 /tmp 多级目录

与每个进程有关的是其多级目录状态, 它决定了进程对多级目录的访问方式. 进程的多级目录状态包括两种:“实状态”(real mode)和“虚状态”(virtual mode).

•“虚状态”下多级目录访问

如果进程处于“虚状态”, 那么安全内核将把对多级目录的访问自动改变为对多级目录下相应有效目录的访问. 如果当前多级目录下, 没有一个有效目录的安全级等于进程的安全级, 那么安全内核将自动创建一个具有进程安全级的有效目录, 且名字对应于进程的安全级.

如图 1 中, 在“虚状态”下, 具有安全级 level1 的进程执行命令 ls/tmp, 则系统显示 file1 和 file2. 具有安全级 level2 的进程执行命令 cd/tmp 时, 则命令执行成功后, 实际上将处于/tmp/level2 目录之下.

“虚状态”是所有用户进程的缺省状态, 用户登录系统后, 代表其工作的进程多级目录状态就自动被安全内核设置为“虚状态”, 除非以后被用户显式地改为“实状态”.

•“实状态”下多级目录访问

当进程处于“实状态”时, 对多级目录的访问与一般目录相同.“实状态”主要用于系统管理员对多级目录的维护和整理.

对多级目录的具体实现, 多级分层文件系统增加了两个系统调用:

①mkmld(char * mldname, int dmode)

创建多级目录 mldname, 安全内核将其标识为多级目录, 并将多级目录标识与其安全级标识放在一起.

②mldmode(int mode)

获取或设置当前进程的多级目录状态, 安全内核可将相应进程的多级目录状态置为“实状态”或“虚状态”.

3.3 隐蔽文件名的实现

文件名的安全级与文件内容的安全级是相同的, 因此, 一个目录中的信息可能具有不同的安全级. 安全内核要对该目

录中的所有内容都实施访问控制, 不允许用户进程通过读目录内容而查访该目录下的文件名, 否则会产生一个存储隐蔽通道.

具体实现就是修改 readdir( )系统调用, 加入存取控制策略 *R*8. 这就使得对某个用户有些文件名、目录名是不可见的(隐蔽的), 不同安全级的用户列表同一目录下的文件时, 结果可能会不一样.

4　系统分析

4.1　系统安全性

在多级分层文件系统中, 若想生成一个对别的用户隐蔽的文件, 就可将文件建立在安全级不被该用户安全级支配或安全级无关的目录中. 不能读取这个目录的用户, 就不能访问该目录的内容及它以下各层的内容. 当然, 若在目录上置以特定的安全级类别, 即使使用了特洛伊木马, 也不能将信息传递到该类别以外的目录去. 这样, 多级分层文件系统通过安全级的合理设置, 实现了文件信息的安全存取控制和域隔离.

另外, 当进程不能对目录进行写访问时, 就不允许进程生成该目录中的客体, 因此, 多级分层文件系统限制了隐蔽存储通道的产生.

4.2　系统效率

在保证多级分层文件系统安全性的前提下, 效率也进行了充分考虑, 比如安全级支配关系判定充分使用内存缓冲区, 判定结果借助高速缓冲机制, 尽可能地减少与磁盘打交道的次数, 等等, 这就使安全性开发对系统效率的影响降到最低.

4.3　系统兼容性

实现多级分层文件系统的操作系统安全内核, 在外部接口几乎相似于原操作系统, 改变的只是在一些系统调用中增加了安全策略检查和增加了一些新的系统调用. 由于系统的原有系统调用的接口参数及其语义都没有任何改变, 因此, 系统原有的应用程序可以不经过修改就能在多级分层文件系统中使用, 可以说多级分层文件系统具有较好的向上兼容性.

5　结束语

国产自主操作系统一个不容忽视的关键就是其安全性, 文件系统则是操作系统不可分割的重要组成部分. 本文中多级分层文件系统的设计和建立具有很好的条理性, 故能成功实现于我国高安全级别的操作系统的设计之中. SecLinux 是基于 Linux 的核心资源开发完成的一个安全操作系统, 设计目标是达到我国 GB17859-1999 的第三级"安全标记保护级", 目前原型系统已分别通过公安部计算机信息系统安全产品质量监督检验中心和国家信息安全测评认证中心的严格检验和认证.

参考文献:

[1] 莫瑞加瑟. 计算机安全的技术与方法 [M]. 吴亚非, 等译. 北京: 电子工业出版社, 1992.

[2] D E Bell, L J La Padula. Secure computer system: mathematical foundations [R]. ESD-TR-73-278, Ⅰ, AD 770 768. Electronic System Division, Air Force Systems Command, Hanscom AFB, Bedford, Massachusetts, 1973.

[3] D E Bell, L J La Padula. Secure computer system: a mathematical model [R]. ESD-TR-73-278, Ⅱ, AD 771 543. Electronic System Division, Air Force Systems Command, Hanscom AFB, Bedford, Massachusetts, 1973.

[4] D E Bell, L J La Padula. Secure computer system: a refinement of the mathematical foundations [R]. ESD-TR-73-278, Ⅲ, AD 780 528. Electronic System Division, Air Force Systems Command, Hanscom AFB, Bedford, Massachusetts, 1974.

[5] 刘文清, 等. 一个修改 BLP 安全模型的设计及在 SecLinux 的应用 [J]. 软件学报, 2002, 13(4): 567- 573.

[6] Handbook for the computer security certification of trusted systems [R]. NRL Technical Memorandum 5540: 062A, 1996.

[7] Managing security on the trusted DG/UX™ system [Z]. Data General Corporation, Westboro, Massachusetts 01580: 1994.

[8] Fraim L J SCOMP. A solution to the multilevel security problem [J]. computer, 1983, 16(7): 26- 34.

作者简介:

刘文清　男, 1967 年 7 月出生于河南虞城, 博士生, 副研究员, 主要研究领域为操作系统安全, 网络安全.

卿斯汉　男, 1939 年生于湖南邵阳, 研究员, 博士导师, 主要研究领域为信息系统安全理论与技术.