文章编号: 1002-1175(2005)06-0707-05

基于 LSM 框架的审计系统的设计与实现*

王富良 贺也平 李丽萍

(中国科学院信息安全技术工程研究中心, 北京 100080; 中国科学院软件研究所, 北京 100080;
中国科学院研究生院, 北京 100049)

(2004 年 11 月 17 日收稿; 2005 年 2 月 24 日收修改稿)

Wang FL, He YP, Li LP. Design and implementation of LSM based secure auditing system. ***Journal of the Graduate School of the Chinese Academy of Sciences***, 2005, 22(6): 707 ~ 711

摘 要 LSM 是 Linux 系统的通用访问控制框架, 在安胜安全操作系统 V4.0 中, 我们在这一访问控制框架的基础上做了适当的扩展并设计实现了安全审计系统. 该安全审计系统与安胜安全操作系统 V2.0 的审计系统相比, 性能得到了很大的提高. 另外, 隐蔽通道会绕过系统的安全策略来进行非法的数据流传输, 我们在审计系统中进行了实时检测和报警.

关键词 安全操作系统, 安全审计, LSM 框架

中图分类号 TP393

1 引言

一个安全操作系统的审计系统的职责是对系统中有关安全的活动进行记录、检查及审核. 它的主要目的是检测和阻止非法用户对计算机系统的入侵, 并显示合法用户的误操作. 审计作为安全系统的重要组成部分, 在 DoD 的 TCSEC(Trusted Computer System Evaluation Criteria)中要求 C2 级以上的安全操作系统必须包含审计功能[1]. 在计算机信息系统安全保护等级划分准则(GB 17859-1999)中也有相应的要求[2]. 在通用安全评价准则 CC 中, 安全审计类 FAU 是安全功能组件中的一个重要组成部分[3].

本审计系统是国产"结构化保护级"安胜安全操作系统 V4.0 的审计部分, 是在"安全标记保护级"安胜安全操作系统 V2.0 的审计系统的基础上发展而来的. 在借鉴了安胜 V2.0 审计部分[4] 的基础上, 扩展了 LSM 构架重新进行审计系统设计, 使该审计系统结构更为清晰, 更加易于扩展, 并在很大程度上提高了系统的性能.

2 基于 LSM 框架的审计系统实现

2.1 通用访问控制框架 LSM

LSM(Linux Security Modules)是在 Linus Torvalds 指导下开发的一个基于 Linux 系统的通用访问控制框架. 在该访问控制框架中提供了两类检查点: 访问控制检查点和安全信息管理检查点.

LSM 采取了如图 1 的方法来控制系统对核心"内部"客体(如: 进程、节点、文件、IPC 等等)的存取访问. 每当系统通过了 DAC 策略检查而试图对一个内部客体进行访问时, LSM 借助于插入到核心代码中的"hook 函数"来仲裁对该客体的访问[5].

图 1 举例说明 LSM 框架的工作原理.

*国家自然科学基金项目(60083007)和国家重点基础研究发展规划(973)项目(G1999035802)资助

LSM 具有如下优点:

(1)真正通用,不同的安全模型的实施仅仅是加载不同的安全模块;

(2)概念上简单,最小的扩散,有效[5].

在我们的安胜安全操作系统 V4.0 中,我们利用 LSM 框架来实现安全操作系统对动态多安全策略的支持.那么,是否可以利用 LSM 框架来设计实现我们的安全审计系统呢?

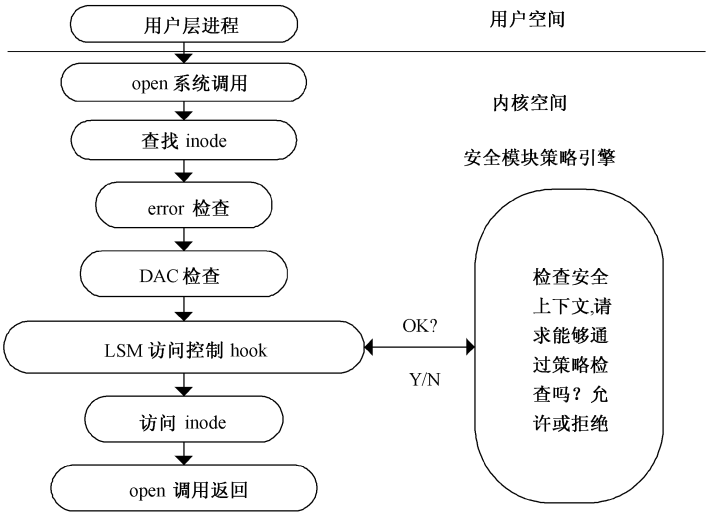

图 1 LSM hook 函数在 open 系统调用中的工作示意图

2.2 利用 LSM 框架设计安全审计系统面临的问题

由于 LSM 主要目的是提供一个通用的访问控制框架,因此,对 hook 函数的添加点,主要是集中在内核对客体的访问控制以及系统的安全信息管理上.而审计系统主要是用来对系统中有关安全的活动进行记录、检查及审核,并不完全适用于这个框架.在 LSM 提供的 hook 点上进行安全审计,会出现如下几个问题.

第一,审计信息不完全,由于访问控制点的 hook 函数在对客体的访问之前调用,而这时审计系统需要记录的许多信息尚未完全生成.审计系统应当在系统对客体的访问后根据系统状态采集需要审计的信息.

第二,系统提供的 hook 函数对审计系统来说是不完备的.由于审计系统是安全操作系统的最后一道防线,需要为系统管理员、入侵监测系统等提供足够的信息.单纯在访问控制点和安全信息管理点上进行信息采集是不够的.对于系统中的安全相关操作,即使不涉及到访问控制,也应当纳入审计系统的审计范围之内.并且,对于特定的操作系统,特定的需求,在某些特定点处进行审计信息采集也是很有必要的.

第三,对访问控制而言,对某一访问控制点处,加入一个 hook 函数就能够很好地进行访问控制了.而对审计系统而言,系统往往由于各种原因,例如出错处理等存在多个退出点.审计系统应当在各个安全相关的退出点上都要添加 hook 函数.

因此,在实际应用中,要利用 LSM 框架的优点,采用这个框架来实现审计系统,必须针对审计系统的特点对该框架进行扩展.

2.3 审计系统在扩展的 LSM 框架上的设计实现

鉴于以上分析,为了使 LSM 能更好地支持审计系统,必须在 LSM 框架中加入第三类的 hook 点——审计信息 hook 点.在我们系统的实现中,由于要采用 LSM 构架支持多策略模型,而让 DTE, RBAC, MAC 等安全策略模块都采用扩充了的 LSM 框架,无疑会降低系统的效率(对于大多数的审计 hook 点,以上安全策略模块基本上用不到).所以应当针对审计系统的要求,对 LSM 框架进行了针对性的扩展.

经过对原有的 LSM 框架和 hook 点的分析,并结合安全审计系统的自身特点,我们在如下几个方面扩展 LSM 构架以使其适应安全审计的需要.第一,在系统中定义安全审计操作集合 audit_operations.在 audit_operations 中包含审计 hook 函数.这样就避免了与其他的安全策略共同使用同样的安全操作集,提高了其他安全策略模块的效率.在系统中添加全局安全审计操作集合变量 audit_ops.对应于系统原有的 security_ops.第二,针对审计系统的要求,添加审计 hook 点.审计 hook 点分为两类:一类是审计入口 hook 检查点,这类 hook 点主要用于判断本次访问操作是否应该被审计,其 hook 函数主要集中在安全敏

感的系统调用的入口点和客体访问的入口点处; 另一类审计 hook 点是审计记录 hook 点, 这类 hook 点的作用是采集审计信息, 用于审计记录, 这类 hook 点位于系统调用的出口点上. 对于一个系统调用具有多出口点的情况, 只要是系统安全相关的, 都要添加 hook 点. 第三, 参照系统对 LSM 的注册函数, 为全局变量 audit_ops 提供注册函数.

图 2 举例说明基于 LSM 框架的审计系统的工作原理(比较于图 1).

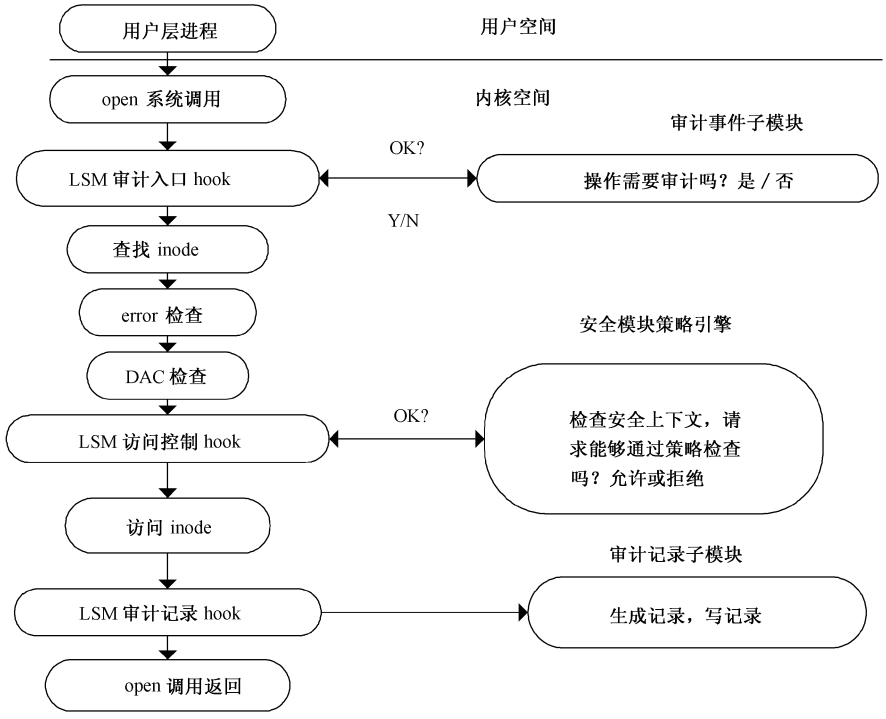

图 2　**LSM** 审计系统在 **open** 系统调用中的工作示意图

2.4　采用 LSM 构架的审计系统的优点

采用 LSM 构架实现的审计系统除去逻辑模型清晰以外, 还具有如下的优点: (1) 支持动态审计注册与注销. LSM 构架中, 审计系统作为一个安全策略模块可以动态地加载和卸载, 具有极大的灵活性. 在系统不需要较高级别的审计模块时, 审计模块可以被动态卸载, 卸载掉审计模块后的系统的 audit_ops 指向NULL(空值), 此时的系统比普通的 Linux 系统在效率上降低很小, 几乎可以忽略不计. (2)审计系统代码结构清晰. 由原来散列在系统内核各处集中于审计记录模块处, 便于源代码的维护与管理. (3)易于系统扩展. 如果需要添加新的审计点, 只需添加、修改审计 hook 点即可, 对内核的改动降低到最小.

3　隐蔽通道的审计与审计报警机制

3.1　隐蔽通道及其国标要求

隐蔽通道的概念最早是由 Butler W. Lampson 在"论限制问题"一文中提出来的. 隐蔽通道在国内文献中也叫做泄漏路径、隐通道, 公认的定义是"允许进程以危害系统安全策略的方式传输信息的通信信道"[6] (见中国国家标准 GB17858—1999《计算机信息系统安全保护等级划分准则》).

GB17859 对结构化保护级安全操作系统中的隐蔽通道分析的要求为: 操作系统开发者应根据实际测量和工程估算, 分析系统中存在的隐蔽信道, 并采取相应措施进行防范. 根据此项要求, 安胜安全操作系统 V4.0 共标记出系统存在的隐蔽通道 18 条, 并根据每条的特点给出了相应的处理[7]. 由于效率和实用以及兼容方面的整体考虑, 我们对系统中的 14 条隐蔽通道采用审计监测的方法进行了处理. 对其余

的4 条,使用隐蔽通道专项技术进行了处理.

3.2 隐蔽通道的审计与报警

针对安胜安全操作系统V4.0 存在的隐蔽通道,我们在审计系统中开发了专门的实时报警子模块进行了处理.关于本节中涉及到的安胜安全操作系统的隐蔽通道的详细知识,请参阅参考文献[7].

要利用隐蔽通道技术进行非法的信息传输,必须达到一定的带宽.带宽太低的隐蔽通道很难被利用.我们在系统中针对这一特点进行了实时跟踪分析.在安胜安全操作系统 V4.0的审计系统中,审计操作员可以通过命令来调整监测与带宽相关的隐蔽通道频率阈值、操作次数阈值等参数,从而使对隐蔽通道的监测效果达到最佳.

为了便于审计,我们将 14 条可审计的隐蔽通道进行了分类处理.对于利用内核变量传输信息的隐蔽通道,我们为内核中每一个可被隐蔽通道利用的变量分别安装了读监测器和写监测器.读、写监测器在系统中同时、独立地采集关于某一内核变量的使用信息.根据读、写监测器所收集到的信息,审计系统根据审计管理员所设置的参数值进行实时分析,在此基础上可以分辨出系统中是否发生了隐蔽通道,以及利用隐蔽通道进行非法通信的各进程.

对于利用内核中的某些操作的特定返回值传输信息的隐蔽通道,我们利用另外的策略进行监测分析.在这一类别的隐蔽通道中,我们也设置了读、写监测器.仅当读监测器根据系统某一操作的特定返回值以及审计管理员所设定的参数认为系统可能发生隐蔽通道时,系统中对应于该读监测器的写监测器才被激活,然后进行相应信息的采集.同样,根据读、写监测器所收集到的信息,审计系统根据审计管理员所设置的参数值进行实时分析,在此基础上可以分辨出系统中是否发生了隐蔽通道,以及利用隐蔽通道进行非法通信的各进程.如果发现某两个进程正在利用隐蔽通道进行非法的信息传输,审计系统的报警机制会向审计操作员发出警报.审计操作员可以设置审计系统使其能够停止系统对参与进程的服务,在最严厉的情况下,系统会对进程所属的用户停止服务.

我们在安胜安全操作系统 V4.0中针对隐蔽通道的审计命令是 adtch.该命令可以设定操作频率,操作阈值,以及发现隐蔽通道后采取的措施(是否停止该进程的服务等).在对隐蔽通道模拟场景的监测中,审计系统能够非常准确地检测到系统中是否有进程利用隐蔽通道技术进行通讯.

4 结论

本系统是中国科学院知识创新工程项目(KGCX-SW-104)"结构化保护级安全操作系统"的审计子系统,是在"安全标记保护级操作系统"的审计系统上发展起来的.基于 LSM 构架,我们对该审计系统的构架进行了重新设计,使其具有更大的灵活性和可扩展性.当本系统的审计子模块卸载后,系统效率的下降可以忽略不计,较原来的审计系统(5%左右)有了很大的提高.对于安胜安全操作系统V4.0 的隐蔽通道,本审计系统的处理效果良好.

References

[1] DoD 5200.28-STD, Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. National Computer Security Center: Ft. Meade, MD, USA, 1985

[2] China State Bureau of Quality and Technical Supervision. National Criteria of People's Republic of China: Classified criteria for security protection of Computer information system. GB 17859-1999, 1999(in Chinese)

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

[4] Liu HF, Qing SH, Liu WQ. Design and realization of auditing in secure OS. *Journal of Computer Research & Development*, 2001, 38(10): 1262~1268 (in Chinese with English abstract)

[5] Zhao ZK, Qing SH, Li LP. Research on the security architecture supporting dynamic and multiple security policies. *Computer Engineering*, 2004, 30(3): 63~66(in Chinese with English abstract)

[6] Qing SH, Liu WQ, Wen HZ. Operating System Security. Beijing: Tsinghua University Press, 2004(in Chinese)
[7] Qing SH, Zhu JF. Covet channel analysis on ANSHENG secure operating system. *Journal of Software*, 2004, 15(9): 1385~1392(in Chinese with English abstract)

附中文参考文献

[2] 中国国家质量技术监督局. 中华人民共和国国家标准: 计算机信息系统安全保护等级划分准则. GB 17859-1999, 1999
[4] 刘海峰, 卿斯汉, 刘文清. 安全操作系统审计的设计与实现. 计算机研究与发展, 2001, 38(10): 1262~1268
[5] 赵志科, 卿斯汉, 李丽萍. 支持动态多策略的安全体系结构应用研究. 计算机工程, 2004, 30(3): 63~66
[6] 卿斯汉, 刘文清, 温红子. 操作系统安全. 北京: 清华大学出版社, 2004
[7] 卿斯汉, 朱继锋. 安胜安全操作系统的隐蔽通道分析. 软件学报, 2004, 15(9): 1385~1392

Design and Implementation of LSM Based Secure Auditing System

WANG Fu-Liang　HE Ye-Ping　LI Li-Ping

(*Engineering Research Center for Information Security Technology, Chinese Academy of Sciences, Beijing* 100080, *China*; *Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*; *Graduate School, Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract　LSM is the access control framework for Linux system. In the ANSHENG secure operating system V4.0, a secure auditing system based on the extended Linux Security Modules was introduced. Compared with the secure auditing system of ANSHENG secure operating system V2.0, the LSM based auditing system has a better performance. Last, the auditing system developed a method to detect covered channels in our ANSHENG secure operating system V4.0.

Key words　secure operating system, secure audit, Linux Security Modules(LSM)