计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(5):26−32 [doi: 10.15888/j.cnki.csa.006321]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 Space 系统的叠合式安全机制①

轩　浩[1], 刘金刚[1,2]

[1](首都师范大学 信息工程学院, 北京 100048)
[2](中国科学院 计算技术研究所, 北京 100080)
通讯作者: 轩　浩, E-mail: xuan13hao@gmail.com

摘　要: 基于 Space 系统 (Space Operating System, SpaceOS) 提出了一种叠合式安全机制, 该机制在保证安全性的同时具有实用性的特点. 首先对系统安全域进行划分, 定义了安全机制的要求. 提出了安全机制的设计思想, 通过形式化方法证明其达到安全的要求, 然后通过 Overlay 文件系统说明技术的可行性, 并将该安全机制应用于 SpaceOS 上. 最后对其进行性能测试, 安全性测试以及实用性测试. 测试表明, 叠合式安全机制具有安全性和实用性, 对 SpaceOS 运行速度影响很小, 具有实用价值.

关键词: Space 系统; 安全机制; 形式化方法; Overlay 文件系统; 实用性

引用格式: 轩浩,刘金刚.基于 Space 系统的叠合式安全机制.计算机系统应用,2018,27(5):26–32. http://www.c-s-a.org.cn/1003-3254/6321.html

Overlay Security Mechanism Based on Space Operating System

XUAN Hao[1], LIU Jin-Gang[1,2]

[1](Information Engineering College, Capital Normal University, Beijing 100048, China)
[2](Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: In this study, we propose an Overlay security mechanism, which is secure and practical, based on Space Operating System (SpaceOS). Firstly, the system security domain is divided, and the requirement of security mechanism is defined. The design idea of security mechanism is put forward, and the security requirement is proved by formal method. Then, the feasibility of the technology is explained by Overlay file system, and the security mechanism is applied to SpaceOS. Finally, the performance test, security test, and practicability test are carried out. The test shows that the Overlay security mechanism is secure and practicable. It has little effect on the speed of the SpaceOS has practical value.

Key words: Space Operating System (SpaceOS); security mechanism; formal method; Overlay file system; practicability

Space 操作系统是计算机科学联合研究院研发的一款基于 Linux 内核, 拥有多项自主知识产权的新型操作系统. 它拥有新颖的安全模式、软硬件一体、操作简单、跨平台使用和升级简单等特点, 使其在各个领域都有很好的应用前景.

长期以来, 安全机制的研究已经取得了很多成果, 如 Xen VMM 安全保障机制[1], MILS 安全机制[2], Linux UID/GID 安全机制. 国内外的研究人员努力的通过一些安全机制的操作系统来保证信息和系统的安全, 如 Adept-50[3]、安全 Xenix[4]、SE-Linux[5]等, 但目前的发展证明这些系统在实际应用上都没有取得成功. 其主要原因是有以下几点: 1) 安全的操作系统一方面强调重要信息的分离, 另一方面确保这些重要信息平台的共享度, 用以减少系统成本, 然而这两个要求很难同时被满足; 2) 如需保证重要信息之间的分离, 参与系统开发的一些平台和工具同样需要保证其安全性, 一旦

① 收稿时间: 2017-08-21; 修改时间: 2017-09-06; 采用时间: 2017-09-14; csa 在线出版时间: 2018-03-12

© 中国科学院软件研究所 http://www.c-s-a.org.cn

提高系统的共享度就意味着代码量的大幅度增加, 最终很难确保系统安全; 3) 同一个系统平台上处理不同的重要数据, 如数据库, 网络等, 需要系统有很强的安全标记能力, 然而如考虑技术支持和经济成本, 这样的设计, 在现有的操作系统生态环境下很难得到实现.

针对 Space 系统安全实用的设计要求, 本文基于 Space 系统, 提出一种叠合式安全机制. 叠合式安全机制通过将安全系统划分为多个安全域, 根据安全策略控制它们的数据流向. 与传统安全系统不同, 叠合式安全机制不需要对重要信息进行安全标记, 保证操作系统正常使用的前提下, 最大程度地降低开发成本和系统应用复杂度. 文章第 1 节对安全模型的概念和安全机制的研究进行说明; 第 2 节提出安全机制的要求并进行形式化的证明; 第 3 节对该安全机制的可行性予以说明; 第 4 节, 介绍该安全机制在 Space 系统上的实现; 第 5 节对该安全机制进行全面的评价. 通过对具有该安全机制的 Space 系统和不具有该安全机制的 Space 系统进行性能测试, 安全性测试以及实用性测试, 得出对系统运行速度影响很小, 满足安全性, 有效性和实用性, 具有实用价值.

1 相关研究

1.1 安全模型的概念

安全模型阐述了一种需求, 即安全策略所表达的安全的简单、抽象和无歧义. 它为安全模型与其实现机制的之间提供了一种基本框架, 也是对一些安全方案需要用哪些机制来确定, 而机制的实现如何把相应的机制应用于系统中, 实现特定的安全方案, 最终达到系统安全的目的[6].

1.2 安全机制的发展

目前基于安全策略的安全模型, Bell 和 LaPadula 提出了最早的安全模型[7], 即 BLP 模型. BLP 模型主要解决了对客体安全性的控制, 核心是管控信息流向中出现的问题, 目的是确保低安全域中的信息顺利地流向高安全域. 其基本想法是, 首先每个主体和客体会得到相应的安全级别, 当主体对象提出对客体进行管控的时候, 该机制通过判断其安全级别能不能达到授权读写的要求. 例如, 通过对主客体的安全级别进行对比, 如果主体高, 给予其读权限. 反之, 给予写权限. 但缺少完整性控制是 BLP 模型不足之处, 主要是很难高效地控制隐蔽通道. 与 BLP 模型不同的是, Biba 模型[8]解决了完整性的问题, 但安全性控制的缺失是其重要的问题. 长久以来, 这两种模型很难相互弥补. 后来, 提出了基于重量级的安全策略的安全模型 Clark-Wilson 模型[9], CW 模型可以在理论上较好地解决了 BLP 模型和 Biba 模型出现的问题, 但在系统中很难实现. Jim Alves-Foss 等研究人员提出了多独立安全级别 MILS 这一概念. MILS 基于 Rushby 隔离[10,11]的思想, 被认为一种高保障的安全体系, 它使用多个可被独立分析的管理单元来构建一套可靠的安全解决方案. RBAC 模型[12]提出了角色的思想, 通过降低授权管理提高安全策略的灵活性. PMI 模型[13]提出了属性证书和授权机构的思想, 其主要优势是独立的授权管理和较低的开发维护成本. PKI 可信度模型[14], 通过提出不可信因子和密钥安全期的思想, 提高证书依赖者对证书持有者的信任评价的精确度. 文献[15]是对 PKI 模型以及其推理方法的一种改进. 文献[16]提出了一种基于虚拟化技术的安全机制, 它通过强调安全隔离避免不同安全信息级别之间的干扰. 这些研究成果, 为叠合式安全机制的形式化分析提供了研究思路和方法.

2 叠合式安全机制

2.1 安全机制要求

安全机制根据信息的重要性将信息划分为安全域, 分别为低安全域和高安全域, 信息流动方向是单向的, 即从低安全域流向高安全域, 同时低安全域的主体对象拥有读高安全域的客体对象的权限, 低安全域的客体对象无法写入高安全域主体, 同级别的域主体对象拥有读写同级别的客体对象的权限, 最终实现信息单向地从低域向高域流动, 防止了高安全域对象信息的泄露.

根据以上的基本思想, 本文对安全机制要求进行定义. 用 S 表示安全系统中全部域的集合, P 表示安全系统的全部物理空间集合, R 表示具有偏序关系$<$的安全标记集, $r1 \prec r2$表示 $r2$ 具有比 $r1$ 更高的安全级. 用函数 $Level: S \to R$ 表示 P 域的安全等级, 函数 $Record: S \to P$ 表示 P 域是可写的物理空间, 函数 $Observe: S \to P$ 表示 P 域是可读的物理空间.

定义 1. 对于$\forall y,z \in P$, $Level(y) \prec Level(z)$, 若:

$$Record(y) \cap Observe(z) \neq \emptyset \quad (1)$$

$$Record(z) \cap Observe(y) \equiv \emptyset \quad (2)$$

© 中国科学院软件研究所 http://www.c-s-a.org.cn

那么该安全策略符合安全机制的安全要求.

对于两个不同安全等级的安全域, 如果低安全 级别域的写物理空间和高安全级别域的读物理空间交集不为空集, 那 么低安全域的信息可以单向地写入高安全域; 如果低安全 级别域的读物理空间和高安全级别域的写物理空间交集为空集, 那 么低安全域拥有高安全域内容的读权限, 而信息无法写入高安全域, 同时也没有权限去读取高安全域所写入的内容, 确保信息不从高安全级别域泄露至低安全级别域, 这符合本文提出的叠合式安全策略的基本要求.

定义 1 和传统的操作系统安全模型存在区别, 定义 1 支持低安全域保留独立的数据信息. 如图 1 所示, 假设 G 和 B 为两个任意的域, B 的安全级别高于 G, $G=\{Gc,Gp\}$, G 是一个划分. 其中 Gc 属于 B 的可读区域, 而 Gp 属于 B 的不可读区域, 即 G 的私有区域, B 为 G 的可写区域, 由于 B 安全级别高于 G, 此时 G 既可以通过 Gc 向高安全域的 B 通报数据, 实现了低安全域数据流向高安全域, 也可以通过 Gp 保护自身数据的安全. 由于 G 的安全级别低于 B, G 的信息无法写入 B, 避免了 G 的数据流向 B, 实现了低安全域数据无法写入高安全域. 这是符合安全系统的安全要求的. 当 $G_C=G$ 时, G 没有私有区域, 此时 B 可以读 G 中的所有的信息.

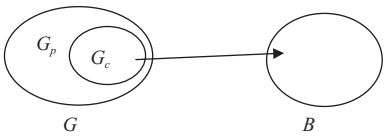

图 1 叠合式安全策略图

2.2 叠合式安全机制

本章节主要提出了叠合式安全机制的设计思想, 用来保证不同的安全域之间的信息流动符合 2.1 节定义 1 的安全要求.

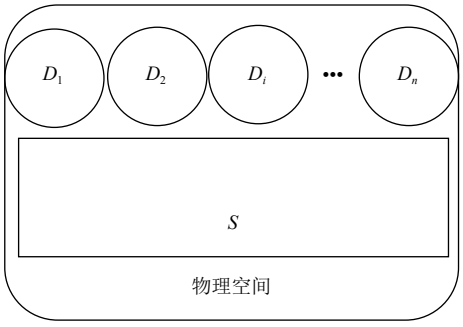

图 2 叠合式安全机制物理空间划分

首先把系统划分为两个类型. 信息域, 每一个信息域对应着不同级别的安全信息; 系统域, 除了信息域之外就是系统域, 有且仅有一个系统域. 根据上述的描述, 需要把物理空间分割为 N+1 个部分: 信息域的物理空间是 D_i 并分配于不同的信息域, 系统域物理空间是 S. 系统域对不同的信息域, 赋予不同的读写权限, 系统域对信息域仅赋予读权限. 系统域和信息域在物理空间上没有任何交集, 相互独立.

基于图 2 对物理空间进行划分, 提出如下规则.

$D=\{D_1,D_2,D_i,\cdots,D_n\}$, $D_i(1\le i\le n)$表示信息域, D 表示信息域物理空间的所有集合. S 表示系统域物理空间. 根据以上定义和物理空间的划分制定如下规则:

规则 1. 对于$\forall i$, $1\le i\le n$, 则:

$$\bigcap_{i=1}^{n} D_i\cap S\equiv\emptyset \tag{3}$$

规则 2. 对于$\forall i$, $1\le i\le n$, 则:

$$D_i\subset Record(D_i) \tag{4}$$

$$D_i\subset Observe(D_i) \tag{5}$$

规则 3. 对于$\forall m,n,1\le m<n\le k$, 若 $Level(D_m)<Level(D_n)$, 则:

$$D_m\subset Observe(D_n)\wedge D_m\not\subset Record(D_n) \tag{6}$$

规则 4. 对于$\forall m,n,1\le m<n\le k$, 若$Level(D_m)<Level(D_n)$, 则:

$$D_n\not\subset Observe(D_m)\wedge D_n\not\subset Record(D_m) \tag{7}$$

规则 1 表示各个安全域存在于不同的物理空间, 各个域之间没有任何交集, 即信息域和系统域在物理空间上没有交集, 信息域集合内的每一个域在物理空间上是相互独立的.

规则 2 表示安全域的可写的物理空间对应着该安全域所存在的物理空间中, 安全域的可读的物理空间对应着该安全域相对应的物理空间.

规则 3 表示如果两个安全域的安全级别不同, 则认为高安全域可以对低的安全域物理空间进行读操作, 不可以对低安全域物理空间进行写操作.

规则 4 表示如果存在两个安全域处于不同的安全级别, 则认为低安全域不可以对高安全域的物理空间进行读操作, 也不可以进行对高安全域的物理空间进行写操作.

下面将对上述的叠合式安全策略进行证明.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

证明:

根据定义 1 对于$\forall y,z\in P$, $Level(y)\prec Level(z)$, 若:

$$Record(y)\cap Observe(z)\neq\emptyset \tag{8}$$

$$Record(z)\cap Observe(y)\equiv\emptyset \tag{9}$$

那么该安全策略符合的安全的要求.

本文所提出的叠合式安全策略, 则需要证明, 对于$\forall n,1\le n\le k$, 若$Level(D_n)\prec Level(S)$, 则:

$$Record(D_n)\cap Observe(S)\neq\emptyset \tag{10}$$

$$Record(S)\cap Observe(D_n)\equiv\emptyset \tag{11}$$

那么叠合式安全策略符合安全要求.

根据规则 2, 可得出:

$$D_n\subset Record(D_n) \tag{12}$$

$$S\subset Record(S) \tag{13}$$

∵对于$\forall n,1\le n\le k$, 当$Level(D_n)\prec Level(S)$, 则:

根据规则 3, 规则 4, 可以得出:

$$\because D_n\subset Record(D_n)\wedge D_n\subset Observe(S) \tag{14}$$

$$\therefore D_n\subset Observe(S) \tag{15}$$

$$\therefore D_n\subset Record(D_n)\wedge D_n\subset Observe(S) \tag{16}$$

$$\therefore Record(D_n)\cap Observe(S)\neq\emptyset \tag{17}$$

根据规则 2, 可得出:

$$S\subset Record(S) \tag{18}$$

∵对于$\forall n,1\le n\le k$, 当$Level(D_n)\prec Level(S)$, 则:

根据规则 3, 规则 4, 可以得出:

$$\because S\not\subset Observe(S)\not\subset Observe(D_n) \tag{19}$$

$$\therefore S\subset \mathrm{Record}(S)\wedge S\not\subset Observe(D_n) \tag{20}$$

$$\therefore \mathrm{Record}(S)\cap Observe(D_n)\equiv\emptyset \tag{21}$$

∴根据式 (17) 和 (21) 的结论, 定义 1 成立.

根据以上的证明, 本文所提出的叠合式安全策略符合安全要求.

3 叠合式安全机制可行性说明

根据第 2.2 节的论述和叠合式安全机制的结构特点, 基于 Space 操作系统的环境下, 可以采用 Overlay 文件系统[17]技术实现该安全机制. Overlay 文件系统是目前使用广泛的叠加式文件系统, 它通过把多个目录文件进行联合, 允许可读写和只读的目录并存, 该文件系统拥有上下层合并, 同名覆盖, 写时复制技术 (COW)[18]等特点. Overlay 文件系统将系统域的底层文件系统和信息域的底层文件系统进行合并. 用户对系统域的文件的修改是在信息域的文件中进行的, 系统域只进行虚拟写的过程, 最终对系统域的修改的数据保存于可写的信息域中. 文件的修改的数据最终挂在到同一个虚拟文件系统下.

Overlay 文件系统和其它层次文件系统 (如 Union FS, AUFS 等) 不同之处是 Overlay 文件系统只有两层: 一个是上层文件系统和一个下层文件系统. 上层拥有读写权限, 下层被赋予只读权限. 当系统需要修改一个文件的时候, 使用写时复制技术 (COW) 将下层的文件复制到可写的上层进行修改. 本文所介绍的实现的叠合式安全机制, 下层为只读的镜像文件即为系统域, 上层是可读写的用户数据层, 即为数据域.

图 3 介绍下层文件 B 复制到上层过程. 用户修改下层中的一个 B 文件时, 用户的修改请求会发给叠合文件系统, 首先判断上层中是否存在 B 文件, 如果没有该文件, 则在上层中创建 B 文件; 从下层中复制对应目录的文件, 则下层中 B 文件复制到对应上层的文件目录中, 这时用户即可在上层中修改 B 文件, 待用户对 B 文件操作完成之后, 数据将保留在上层, 而下层中的 B 文件并没有发生任何变化.

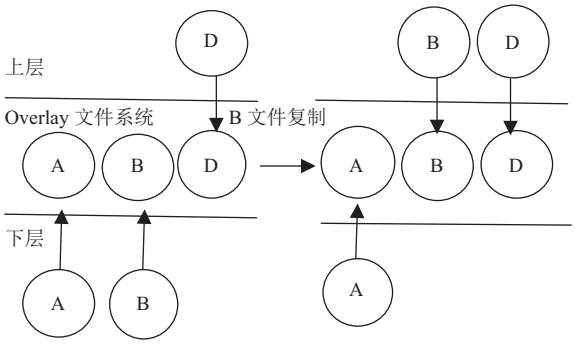

图 3 上层的 B 文件初始化图

在 Overlay 文件系统中, 即有两个 B 文件, 一个来自于下层中的未修改的 B 文件, 另一个来自于上层中可以修改的 B 文件. Overlay 文件系统中呈现的是下层文件系统和上层文件系统的并集, 而上层文件系统中的文件拥有较高的优先级. 由于用户操作的是上层中的一个已经修改了的文件, 而下层中操作系统文件并没有被修改, 符合本文所提出的安全机制的设计要求.

实际的开发中, 也可以使用其他的叠加式文件系统实现叠合式安全机制, 如 AUFS, BTRFS, ZFS 等, 这

© 中国科学院软件研究所 http://www.c-s-a.org.cn

些叠加式文件系统大部分已经使用在一些商业的 Linux 系统. 最终选择使用 Overlay 文件系统的原因不仅是因为其高效稳定轻量级, 更主要是 Overlay 文件系统只有两层的叠加, 这决定了它的读效率高于其它的叠加式文件系统, 更容易满足实用性的需求.

综上所述, 本文所提出的叠合式安全机制在技术上是可行的.

4 实际应用

根据第三节提出的可行性说明, 可以将叠合式安全机制运用于 Space 操作系统上, 图 4 为 SpaceOS 的实现框架, 其中 Space_User 域对应图 2 中的信息域中的 $D_i(i=1)$, Space_System 域对应图 2 中的系统域 S.

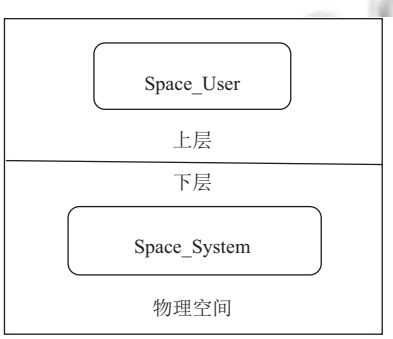

图 4 Space 系统实现框架图

根据图 1 为定义的安全策略, Gc 作为 Space_System 域位于下层, 该域是只读的, 在物理上不可被改写. Space_System 域和 Space_User 域物理空间没有交集, 相互独立, 符合规则 1 要求. Space_User 域位于上层且文件系统 B 本身作为读写层空间上独立于 Space_System 域文件系统, 这样, 对 Gc 中的所有变化均可以在 Space_User 域文件系统 B 中反应出来, 而对 B 的任何操作 G 都无法感知, 这符合规则 2 的要求.

由于 Space_System 域安全级别高于 Space_User 域, 所以 Space_System 域中存放一个完整的操作系统, Space_User 域存放着对 Space_System 域内的操作系统的修改 (如安装 Space_User 存放着重新配置系统的信息等) 的一个副本和用户的一些文件等并具有不同的文件系统类型 (如 Space_System 域物理空间是 FAT32 格式, Space_User 域的格式是 Ext4), Space_System 域可以虚拟写入 Space_User 域, 实际上的操作是在 Space_User 域完成的, 这符合规则 3 要求. Space_User 域可以看到 Space_System 域的修改变化, 但不可写入 Space_System 域, 反之 Space_System 域的变化不会被 Space_User 域察觉, 这符合规则 4 的要求. Overlay 文件系统会合并 Space_System 域和 Space_User 域的文件系统树, 最终看到的是叠加后文件系统, 用户不会察觉到这样的变化, 符合叠合式安全模型的要求.

事实上, 这种方式的应用是广泛的, 如同管理员和客户关系一样, 管理员拥有高于客户的优先级, 管理员可以查看其他客户的修改变化, 但是客户无法查看管理员的一些操作. 这样在不同个体对象之间一方面保证独立, 同时确保优先级较高的个体可以查看优先级较低的个体, 而优先级较低的个体察觉不到高级别个体的变化, 从而实现了信息由一个或者多个低安全域单向地流入同一个高安全域.

5 安全机制评价

为了对叠合式安全机制进行全面的评价, 对安装了叠合式安全机制的 Space 系统进行了性能测试, 安全性测试和实用性测试.

5.1 实用性测试

通过对比安装该安全机制的 Space 系统和未安装该安全机制的 Space 系统的进行实用性对比. 安装了该安全机制的 Space 系统不需要做额外的配置, 同样可以运行 Space 系统上的应用软件和服务, 如 SpaceWS, QQ, Microsoft Word 2003 和 Photoshop 等大型软件. 在安装了安全机制的 Space 系统上, 如一些大型软件和服务崩溃, 用户可以删除信息域相关的错误配置文件, 该软件或者服务同样可以正常使用. 而未安装了机制的 Space 系统, 如删除相关的软件和服务的配置文件, 则软件和服务崩溃, 最终导致软件和服务无法正常使用, 甚至会影响系统的整体性能.

该测试分别删除安装了该安全机制和未安装安全机制的 Space 系统下的/usr/bin/spacews/windows/lib 文件. 之后, 再运行两个系统的 Photoshop 软件, 发现未安装叠合式安全机制的 Space 系统的 Photoshop 软件已经无法正常使用. 而安装了该安全的 Space 系统的 Photoshop 软件依然可以正常运行.

测试表明, 相比于未安装该安全机制的 Space 系统, 该安全机制可以为用户的使用提供更加稳定的系统保障, 可以满足系统的实用性.

5.2 安全性测试

安全性测试的主要目的是测试该安全机制是否能够抵御一般性的网络攻击. 黑客利用服务器的漏洞获

© 中国科学院软件研究所 http://www.c-s-a.org.cn

取超级权限的用户的 shell 程序. 为了让简化测试流程, 通过在安装了该安全机制的和未安装安全机制的 Space 系统上运行 netcat 程序为攻击者提供一个拥有根权限的 shell 进程. 这和黑客利用系统程序漏洞获得 shell 的方法是一样的. 为了测试系统的安全性, 主要通过以下两个实验.

第一个实验是修改网络服务器上的相关的文件(如网页文件), 通过破坏用户的文件进行攻击. 当安全机制起作用时, 通过 netcat 获得的远程的 shell 无法修改服务器目录下的文件, 同时也不能在该目录下创建其它文件.

第二个实验是文件的盗取实验. 通过破坏系统的安全性进行攻击. 主要通过 mutt 程序获取/etc/shadow 文件进行攻击. 在未安装该安全机制的 Space 系统上运行 muut-a/etc/shadow test1@cnu.edu.cn 命令可以把/etc/shadow 文件发送到 test1@cnu.edu.cn 邮箱中, 但是在安装了该安装该安全机制后, 邮件中未接收该文件, 主要是因为系统域的安全级别高于信息域, 没有权限读取/etc/shadow 的文件. 实验表明该安全机制能够保证 Space 系统的数据的安全, 可以满足安全性.

5.3 性能测试

由于 Space 系统增加了叠合式安全机制, 同时也增加了操作系统运行的复杂性. 这种复杂性若影响系统的运行速度, 会导致这种安全机制不具实用价值, 同样影响系统的有效性.

对系统运行速度采用的测试方法为: 在具有叠合式安全机制和不具有叠合式安全机制的 Space 操作系统上运行同一个文件 IOzone 工具. 在高并发高吞吐量的环境下, 同时读写多个大小不一的文件, 来测试不同的安全机制操作系统的运行速度, 并比较其性能差异. 测试机器的配置 2.20 GHZ, i5-5200U 处理器, 4 G 内存以及 24 G 固态硬盘.

在高并发高吞吐量的环境下同时进行如下测试: (1) 测试一个已存在的大小为 1 M 小文件和 256 M 的大文件的文件读速度. (2) 测试一个最近读过的大小为 1 M 小文件和 256 M 的大文件的文件读速度. (3) 测试向一个新文件写入一个大小 1 M 的小文件和 256 M 大文件的性能. (4) 测试向一个已存在的文件写入一个大小为 1 M 的小文件和大小为 256 M 大文件的性能.

图 5 是叠合式安全机制的读性能对比, 图 6 是叠合式安全机制写性能对比. 横坐标是记录大小为递增的 4 K 到 16 M 的文件块, 单位为 Kbyte; 纵坐标为传输速率, 单位为 Kbyte/s, 虚线为安装安全机制之前的数据, 实线为安装之后的数据.

从图 5, 图 6 结果可以看出, 具有叠合式安全机制 Space 操作系统的读写性能与不具有叠合式安全机制的性能相差不大. 本文为 Space 操作系统研究、开发的叠合式安全机制能够承担高吞吐量的并发读写操作, 并保证了 Space 系统的有效性和实用性.

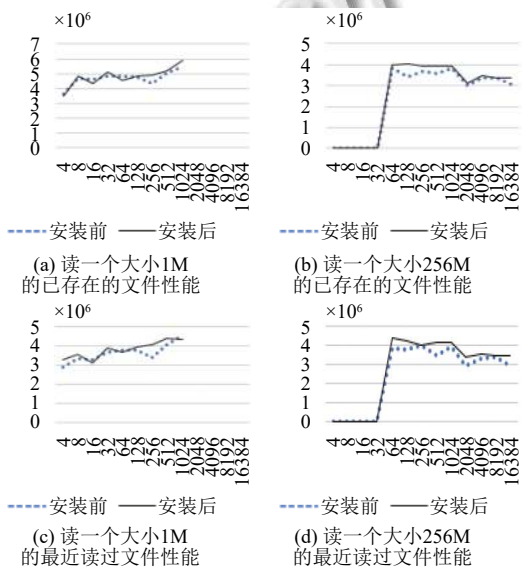

图 5 1 M 文件和 256 M 文件读性能对比图

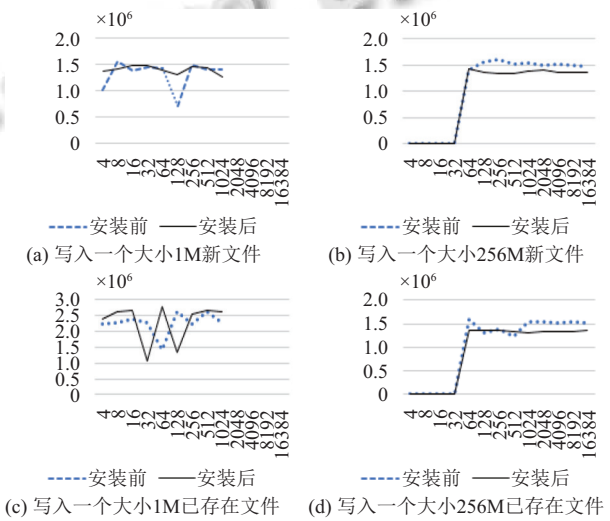

图 6 1 M 文件和 256 M 文件写性能对比图

6 结束语

本文设计实现一种叠合式安全机制, 形式化说明

© 中国科学院软件研究所 http://www.c-s-a.org.cn

证明该机制可以达到安全的目的. 通过对系统的测试可以看出, 该安全机制具有足够的安全性, 很好的实用性, 其具有实用价值. 相比于其它设计方案, 叠合式安全机制无论是在维护性和可扩展性方面都有着很大的优势.

参考文献

1 马喆, 禹熹, 袁傲, 等. Xen 安全机制探析. 信息网络安全, 2011,(11):31–35.[doi:10.3969/j.issn.1671-1122.2011.11.009]

2 Alves-Foss J, Oman PW, Taylor C, *et al*. The MILS architecture for high-assurance embedded systems. International Journal of Embedded Systems, 2006, 2(3-4): 239–247.

3 Weissman C. Security controls in the ADEPT-50 time-sharing system. Proceedings of the 1969 Fall Joint Computer Conference. Las Vegas, NV, USA. 1969. 119–133.

4 Gligor VD, Chandersekaran CS, Chapman RS, *et al*. Design and implementation of secure xenix. IEEE Transactions on Software Engineering, 1987, SE-13(2): 208–221. [doi: 10.1109/TSE.1987.232893]

5 Archer M, Leonard E, Pradella M. Analyzing security-enhanced Linux policy specifications. Proceedings of the IEEE 4th International Workshop on Policies for Distributed Systems and Networks. Lake Como, Italy. 2003. 158–169.

6 卿斯汉, 刘文清, 刘海峰. 操作系统安全导论. 北京: 科学出版社, 2003.

7 Bell DE, LaPadula LJ. Secure computer systems: Mathematical foundations. Bedford, MA: Mitre Corp, 1973.

8 周洲仪, 贺也平, 梁洪亮. 基于 Biba 和 Clark-Wilson 策略的混合强制完整性模型. 软件学报, 2010, 21(1): 98–106.

9 Clark DD, Wilson DR. A comparison of commercial and military computer security policies. Proceedings of 1987 IEEE Symposium on Security and Privacy. Oakland, CA, USA. 1987. 184.

10 Rushby JM. Design and verification of secure systems. Proceedings of the 8th ACM Symposium on Operating Systems Principles. Pacific Grove, CA, USA. 1981. 12–21.

11 Rushby JM. Proof of separability a verification technique for a class of security kernels. Proceedings of the 5th International Symposium on Programming. Turin, Italy. 1982. 352–367.

12 Bertino E. RBAC models-concepts and trends. Computers & Security, 2003, 22(6): 511–514.

13 梁冰, 廖湘柏. 基于 PKI、PMI 的数字图书馆安全管理. 计算机系统应用, 2009, 18(3): 130–132, 45.

14 Li W, Ping L. Trust model to enhance security and interoperability of cloud environment. IEEE International Conference on Cloud Computing. Springer. Berlin, Heidelberg. 2009. 69–79.

15 张明德, 郑雪峰, 吕述望. 改进的 PKI 可信度模型. 小型微型计算机系统, 2012, 33(2): 370–375.

16 吉晨, 石勇, 戴明, 等. 基于轻量级虚拟化环境的可信多级安全容器机制. 计算机应用研究, 2017, 34(6): 1770–1773.

17 Hipp BA, Wong C, Yeh YY. Method and system for an overlay filesystem: US, 7197516. [2007-03-27].

18 Hitz D, Malcolm M, Lau J, *et al*. Copy on write file system consistency and block usage: US, 6892211. [2005-05-10].

© 中国科学院软件研究所 http://www.c-s-a.org.cn