

文章编号:1001-9081(2006)12-2941-04

基于强制访问控制的文件安全监控系统的设计与实现

王 雷, 庄 毅, 潘龙平

(南京航空航天大学 信息科学与技术学院, 江苏 南京 210016)

(mad_ray1982@hotmail.com)

摘 要:重点分析了基于信息保密的 BLP(Bell-LaPadula)模型和基于信息完整性的 Biba 模型,基于这两个模型设计了兼顾系统保密性和完整性需求的强制访问控制模型,并结合 Windows 文件过滤驱动程序开发了一个基于该强制访问控制模型的文件安全监控系统,对其主要模块和关键技术进行了详细介绍。该文件安全监控系统可有效地维护文件系统的保密性和完整性,检测并阻断本地与网络的入侵。

关键词:强制访问控制;模型;文件过滤驱动;文件监控系统

中图分类号: TP309 **文献标识码:** A

Design and implementation of file watching system based on mandatory access control

WANG Lei, ZHUANG Yi, PAN Long-ping

(College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu 210016, China)

Abstract: Bell-Lapadula model based on information confidentiality and Biba model based on information integrity were introduced. Then a new Mandatory Access Control (MAC) model based on the two modules above was designed. By using Windows NT file filter driver, a file watching system which adopted this new MAC model was developed, and the components and key technologies of it were given in detail. This file watching system has the advantages of protecting information confidentiality and integrity, and resisting the attacks from both local and remote users.

Key words: Mandatory Access Control (MAC); model; NT file filter driver; file watching system

0 引言

随着计算机应用的日益普及,特别是网络和数据库技术的发展,计算机安全也越来越成为一个亟待解决的问题,其根本目标就是要保障计算机中信息的保密性、完整性和可用性。Windows 操作系统采取的是自主访问控制系统,灵活度高、粒度小,缺点是信息在移动过程中其访问权限关系会被改变,通过添加用户会使访问权限传递给另外一个用户,从而使不具备访问权限的用户也可以访问该目标资源。

强制访问控制(Mandatory Access Control, MAC)利用强制性的规定防止信息的不安全流动,可以非常有效地防止特洛伊木马的攻击。它在一些对操作系统安全要求很高的部门如多级军用系统中对信息安全的维护作用是非常大的。

本文重点研究强制访问控制,讨论了 BLP(Bell-LaPadula)模型和 Biba 模型,在此基础上提出了可同时维护 Windows 文件系统保密性和完整性的新强制访问控制模型。同时加强系统的入侵检测功能,有效地维护了操作系统的信息安全。

1 强制访问控制技术

强制访问控制技术的基本思想是:每个主体和客体都有既定的安全属性,主体对客体能否执行特定的操作取决于二

者安全属性之间的关系。通常所说的 MAC 主要是指 TCSEC(Trusted Computer Security Evaluation Criteria)中的 MAC,它主要用来描述美国军用计算机系统环境下的多级安全策略。在多级安全策略中,安全属性用二元组表示,记作(密级,类别集合),密级表示机密程度,类别集合表示部门或组织的集合。BLP 安全模型^[1,2]是最著名的安全策略模型。BLP 模型是根据军方的安全政策设计的,它要解决的本质问题是对具有密级划分的信息访问进行控制。

BLP 模型中密级是集合{绝密,机密,秘密,公开}中的任意一元素,此集合是全序的,即:绝密>机密>秘密>公开。类别集合是系统中非分层元素集合中的一个子集,这一集合的元素依赖于所考虑的环境和应用领域。BLP 模型中,安全属性的集合形成一个满足偏序关系的格,此偏序关系称为支配关系。BLP 模型对系统中的每个用户分配一个安全属性(又称保密等级),它反映了对用户不将保密信息泄露给无相应安全属性用户的置信度。用户激活的进程也将被授予此安全属性;BLP 模型为系统中的每个客体也分配一个安全属性,它反映了客体内信息的保密度,也反映了未经授权向不允许访问该信息的用户泄露这些信息所造成的潜在威胁。BLP 模型考虑以下几种访问模式:

- (1) 只读:读包含在客体中的信息;
- (2) 添加:向客体中添加信息且不读客体中的信息;

收稿日期:2006-06-08;修订日期:2006-09-04

基金项目:航空基金资助项目(04C52009);国防工业基础基金资助项目(Q172005A001)

作者简介:王雷(1982-),男,江苏徐州人,硕士研究生,主要研究方向:计算机网络安全;庄毅(1956-),女,江苏南京人,教授,硕士,主要研究方向:网络安全、分布式计算;潘龙平(1981-),男,江苏连云港人,硕士研究生,主要研究方向:计算机网络安全。

(3) 执行: 执行一个客体(程序);

(4) 读写: 向客体中写信息且允许读客体中信息。

BLP 模型中主体对客体的访问必须满足以下两个规则:

(1) 简单安全规则: 仅当主体保密级不低于客体保密级且主体的类别集合包含客体的类别集合时, 才允许该主体读该客体;

(2) *—规则: 仅当主体保密级不高于客体保密级且客体的类别集合包含主体的类别集合时, 才允许该主体写该客体。

BLP 模型的不足主要表现在两个方面: 应用领域比较窄, 使用不灵活, 一般只用于军方等具有明显等级观念的行业或领域; 完整性方面控制不够, 它重点强调信息向高安全级的方向流动, 对高安全级信息的完整性保护强调不够。

Biba 模型^[1]主要针对信息完整性的保护方面。与 BLP 模型类似, Biba 模型用完整性等级取代了 BLP 模型中的保密等级, 而访问控制的限制正好与 BLP 模型相反:

(1) 简单完整规则: 仅当主体的完整级大于等于客体的完整级且主体的类别集合包含客体的类别集合时, 才允许该主体写该客体;

(2) 完整性制约规则(*—规则): 仅当主体的完整级不高于客体完整级且客体的类别集合包含主体的类别集合时, 才允许该主体读该客体。

本文在 BLP 和 Biba 模型研究的基础上, 结合二者的优点, 设计了可同时维护 Windows 文件系统保密性和完整性的强制访问控制模型。

设每个主体拥有两个安全标签: 保密性标签和完整性标签。主体保密性标签定义为三元组 (H_s, H_c, Ch) , H 为保密性等级, C 为信息范畴。其中 $H_s \subseteq H$, 为主体的最高保密性等级; $H_c \subseteq H$, 为当前主体的保密性等级; $Ch \subseteq C$, 为主体的保密性信息范畴。主体的完整性标签定义为二元组 (I, Ci) , I 为完整性等级, $Ci \subseteq C$, 为完整性信息范畴。

每个客体也有两个安全标签: 保密性标签和完整性标签。不同的是客体的保密性标签定义为二元组 (H_o, Ch) , $H_o \in H$, 为客体的保密性等级; $Ch \subseteq C$, 为客体的保密性信息范畴。客体的完整性标签与主体完整性标签的定义相同^[3,4]。该模型有如下三条安全特性:

(1) 主体 s 可以对客体 o 进行读操作必须满足: $(H_s(s) \geq H_o(o) \wedge Ch(o) \subseteq Ch(s)) \wedge (I(s) \leq I(o) \wedge Ci(s) \subseteq Ci(o))$;

(2) 由于添加操作不需要获取文件原有的内容, 因此主体 s 可以对客体 o 进行添加操作应满足: $(H_c(s) \leq H_o(o) \wedge Ch(s) \subseteq Ch(o)) \wedge ((I(s) \geq I(o)) \wedge (Ci(o) \subseteq Ci(s)))$;

(3) 主体 s 可以对客体 o 进行写操作必须满足: $(H_c(s) = H_o(o) \wedge Ch(s) = Ch(o)) \wedge (I(s) \geq I(o) \wedge Ci(o) \subseteq Ci(s))$ 。

2 监控系统体系结构

文件监控系统由两部分组成: 1) 安全监控模块, 通过编写文件过滤驱动程序实施对文件系统的监视, 并将监视信息及时反馈给用户控制模块; 2) 控制模块, 实现对监控模块的配置、控制, 设置监控条件, 进行入侵追踪, 实现与其他系统集成的接口。

2.1 安全监控模块

2.1.1 文件操作监控器

监控器是文件监控系统的核心, 拦截用户进程的操作并进行检查, 这部分功能主要通过文件过滤驱动来实现^[5,6]。Windows 所有的 I/O 请求均由 I/O 管理器进行管理, 执行过程如下:

(1) 操作系统的 I/O 管理器从非分页内存分配一个 IRP, 响应一个 I/O 请求。基于客户指定的 I/O 函数, I/O 管理器将该 IRP 传递给合适的文件过滤驱动程序的 Dispatch 例程;

(2) Dispatch 例程检查请求的参数是否有效, 如果有效, 驱动程序根据请求的内容进行相应的操作。否则设置错误状态信息直接返回;

(3) 操作完成时, 将数据(如果有)和状态信息存放到 IRP 中并返回给 I/O 管理器;

(4) I/O 管理器对返回的 IRP 进行适当的处理后将状态和数据(如果有)返回给用户。

文件过滤驱动程序接收到打开、创建、读写、关闭文件等请求, 这些请求通常由用户进程发起并通过 I/O 子系统管理器分发到文件系统。图 1 说明了本地磁盘处理用户线程文件操作请求的流程。

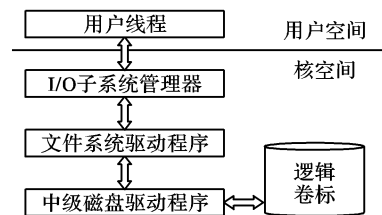


图1 本地磁盘处理用户线程文件操作请求流程

WDM 支持分层模型, 所以驱动程序的功能可在多个层次上实现。文件系统监视驱动程序是一个过滤驱动程序, 将文件操作请求截获后进行安全审查。过滤驱动程序的工作过程是:

(1) 挂接到目标设备对象, 截获所有分发到目标设备对象的 IRP 请求;

(2) 根据实际需要建立合适的 IRP, 分发到目标设备对象, 驱动程序可以创建一个新的 IRP 结构或者根据主 IRP 创建一个关联 IRP;

(3) 设定完成例程。当被挂接的驱动程序完成 IRP 时将会调用该完成例程。

要实现文件系统过滤驱动程序, 就要生成挂接到文件系统的设备对象。例如要对 E 盘进行监控, 则要获得 E 分区的驱动设备对象, 然后驱动程序生成一个无名的设备对象, 挂接到 E 分区的驱动设备对象。挂接成功后, 所有对该分区的文件操作(IRP)都将先经过驱动程序创建的设备对象的处理, 进行安全过滤审计, 决定是拒绝还是继续分发 IRP。

本监控系统将文件的权限分为三种: 完全控制、只读和禁止访问。文件操作的拦截在下列系统处理例程中执行: IRP_MJ_CREATE、IRP_MJ_CLOSE、IRP_MJ_CLEANUP、IRP_MJ_READ、IRP_MJ_WRITE 等, 以及各个 FastIO 例程。用户线程执行的文件操作最终将会调用相关例程, 因此要实现安全过滤审计就需要对相关的例程进行拦截, 验证其合法性。对于需要拦截的操作, 过滤程序调用相关的 Dispatch 函数进行处理, 例如:

```
pDriverObject -> MajorFunction [ IRP_MJ_CREATE ] = FW_CreateDispatch; //创建例程
```

```
pDriverObject -> MajorFunction[ IRP_MJ_DEVICE_CONTROL ] =
    FW_DeviceControlDispatch;           //控制例程
pDriverObject -> MajorFunction [ IRP_MJ_READ ] = FW_
    ReadDispatch;                       //读例程
pDriverObject -> MajorFunction [ IRP_MJ_WRITE ] = FW_
    WriteDispatch;                      //写例程
```

此外,过滤程序调用相关的 FastIo 例程,例如:

```
pFastIoDispatch -> FastIoRead = FW_FastIoRead;
pFastIoDispatch -> FastIoWrite = FW_FastIoWrite;
pFastIoDispatch -> FastIoQueryBasicInfo =
    FW_FastIoQueryBasicInfo;
pFastIoDispatch -> FastIoQueryStandardInfo =
    FW_FastIoQueryStandardInfo;
```

2.1.2 强制访问控制模型

要实现强制访问控制必须设计和实现强制访问控制模型,该模型主要功能是接收来自用户控制模块的管理请求,添加、删除、更新主客体的保密性和完整性标签,实现强制访问控制的保密性和完整性两大策略:

(1) 保密性模块。该模块为每个主体和客体定义一个保密性标签,系统中分别以结构体定义主体和客体的保密性标签:

```
/* 主体保密性标签定义 */
typedef struct
{
    SECURITY_LEVEL top_level;           //主体最高保密性等级
    SECURITY_LEVEL cur_level;           //主体当前保密性等级
    SECURITY_CATEGORY category;         //主体保密性类别集合
} SUB_SEC_LABEL, * P_SUB_SEC_LABEL;
/* 客体保密性标签定义 */
typedef struct
{
    SECURITY_LEVEL sec_level;           //客体保密性等级
    SECURITY_CATEGORY category;         //客体保密性类别集合
} OBJ_SEC_LABEL, * P_OBJ_SEC_LABEL;
```

(2) 完整性模块。该模块为每个主体和客体定义完整性标签,系统中主体和客体共用相同的完整性标签:

```
/* 完整性标签定义 */
typedef struct
{
    INTERGRITY_LEVEL int_level;         //完整性等级
    INTERGRITY_CATEGORY category;       //完整性类别集合
} INT_LABEL, * P_INT_LABEL;
```

(3) 主客体安全标签。系统将保密性标签和完整性标签结合,形成主客体的安全标签:

```
/* 主体安全标签 */
typedef struct
{
    SUB_SEC_LABEL sec_label;           //主体保密性标签
    INT_LABEL int_label;               //主体完整性标签
} SUB_MAC_LABEL, * P_SUB_MAC_LABEL;
/* 客体安全标签 */
typedef struct
{
    OBJ_SEC_LABEL sec_label;           //客体保密性标签
    INT_LABEL int_label;               //客体完整性标签
} OBJ_MAC_LABEL, * P_OBJ_MAC_LABEL;
```

(4) 标签比较函数。系统为主客体的保密性标签和完整性标签分别定义比较函数:

```
cmp_sec_label( P_SUB_SEC_LABEL psub_sec_label, P_OBJ_SEC_
    LABEL pobj_sec_label, boolean isread)
cmp_int_label( P_SUB_INT_LABEL psub_int_label, P_OBJ_INT_
    LABEL pobj_int_label, boolean isread)
```

关于保密性标签的比较,有如下定义:

(1) 若两个保密性标签的密级相等并且类别集合也相

等,则称这两个保密性标签相等;

(2) 若以下条件成立则称保密性标签 A 大于保密性标签 B: A 的密级大于 B 的密级并且两个类别集合相等;两密级相等并且 A 的类别集合是 B 的类别集合的真超集; A 的密级大于 B 的密级并且 A 的类别集合是 B 的类别集合的真超集;

(3) 若两个保密性标签不相等并且哪个保密性标签都不大于另一个保密性标签,则称这两个保密性标签不可比较。

关于完整性标签的比较,有如下定义:

(1) 若两个完整性标签的完整级相等并且类别集合也相等,则称这两个完整性标签相等;

(2) 若以下条件成立,则称完整性标签 A 大于完整性标签 B: A 的完整级大于 B 的完整级并且两个类别集合相等;两完整级相等并且 A 的类别集合是 B 的类别集合的真超集; A 的完整级大于 B 的完整级并且 A 的类别集合是 B 的类别集合的真超集;

(3) 若两个完整性标签不相等并且哪个完整性标签都不大于另一个完整性标签,则称这两个完整性标签不可比较。

比较结果结合新模型定义的规则,可以实现操作系统的强制访问控制,并且兼顾了系统的保密性与完整性。

2.1.3 安全标签的维护

安全标签的维护通过两个数据结构实现,主体(用户进程)安全标签采取的是哈希表结构,而客体(文件)安全标签则采取树结构,这主要是出于优化查询速度以及方便存取的目的。通过对数据结构的操作实现对主客体安全标签的创建、插入、删除、更新等,例如:

```
get_obj_mac_label(( char * ) &fileName) //获取客体安全标签
insert_obj_mac_label(( char * ) &filename, &objlabel)
//插入客体安全标签
get_sub_mac_label(( char * ) &username, ( char * ) &procName,
    &sublabel) //获取主体安全标签
insert_sub_mac_label(( char * ) &username, ( char * ) &procName,
    &sublabel) //插入主体安全标签
```

本系统核心是在文件过滤驱动程序中的 FW_DeviceControlDispatch 控制例程中进行安全标签的相关操作,通过在该例程中传入 IoCtrlCode 决定具体的操作方式。如表 1 所示。

表 1 IoCtrlCode 及其说明

IoCtrlCode	说明
IOCTL_FW_AddMACLabel	添加安全标签
IOCTL_FW_RemoveMACLabel	移除安全标签
IOCTL_FW_STARTFILTER	开始过滤
IOCTL_FW_STOPFILTER	停止过滤
IOCTL_FW_VERSION	获取当前驱动版本

在实验中,分别针对特定的主体(例如用户名“Administrator”,进程名“notepad.exe”)和客体(例如文件名“E:\FileWathcher\test.txt”)加上了相应标签,例如:

```
SUB_MAC_LABEL sublabel; //主体安全标签
sublabel.sec_label.top_level = SEC_LEVEL_UNCLASSIFIED;
//最高保密性等级
sublabel.sec_label.cur_level = sublabel.sec_label.top_level;
//当前保密性等级
sublabel.sec_label.category = 0x1010; //保密性类别集合
sublabel.int_label.int_level = INT_LEVEL_IMPORTANT;
//完整性等级
sublabel.int_label.category = 0x1010; //完整性类别集合
```

```

OBJ_MAC_LABEL objlabel;           //客体安全标签
objlabel.sec_label.sec_level = SEC_LEVEL_SECRET;

//保密性等级
objlabel.sec_label.category = 0x1010; //保密性类别集合
objlabel.int_label.int_level = INT_LEVEL_IMPORTANT;

//完整性等级
objlabel.int_label.category = 0x1010; //完整性类别集合

```

在进行了安全标签的比较后,系统决定主体能否访问或修改客体,监控系统运行结果如图2所示。

时间	文件名	进程名	进程端口	用户名	进程...	攻击主机IP	源
2006/04/24 15:19:11	E:\FileWatcher\test.txt	notepad.exe	未打开	Administrator	打开	本机	16
2006/04/24 15:20:24	E:\FileWatcher\test.txt	notepad.exe	未打开	Administrator	修改	本机	16

图2 文件安全监控系统运行界面

由图2可知监控系统阻断的进程信息,包括进程名、用户名以及端口等;实验结果表明“E:\FileWathcher\test.txt”的内容无法读取和更改,原因在于监控系统发现了非法的进程(打开)操作,并成功阻断了该进程。

2.1.4 驱动程序与控制程序的通信

本文件安全监控系统采取 DeviceIoControl 异步调用的方法实现驱动程序与控制程序的通信。驱动程序提供了一组控制接口,实现对驱动程序的配置等功能,这些接口主要由 DeviceIoControl 实现,应用程序通过调用 Win32 DeviceIoControl 函数,向驱动程序发出控制命令。同时驱动程序捕获到一些异常操作,通过接口及时发送给控制程序进行分析。

2.2 用户控制模块

控制模块主要实现对文件系统过滤驱动程序的图形化视图控制、监视消息的显示、攻击主机 IP 地址追踪、Socket 接口等功能,具体结构如图3所示:

(1) Driver Message Controller 负责和驱动程序通信,启动一个线程监听驱动程序发送过来的消息;

(2) 标签维护模块实现了主客体安全标签的添加、删除、清空等操作,监视消息显示模块主要用于显示文件访问日志;

(3) Sniffer 模块对流经本机网卡的所有 IP 包进行截获,保存一定数量的最新 IP 包,记录接收到 IP 包的时间戳,为实现对攻击主机 IP 地址的追踪准备数据;

(4) Socket Controller 实现了控制程序的 Socket 接口。

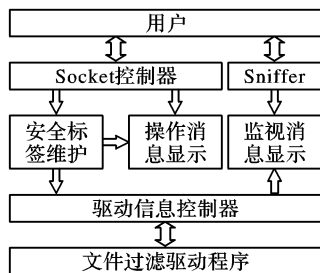


图3 用户控制程序结构

3 结语

在 Windows 原有的安全基础上,通过设置文件过滤驱动程序和加载强制访问控制模型,实现了基于强制访问控制的文件安全监控系统,并且对原有的强制访问控制模型进行改进,使其兼顾保密性和完整性,有效地保护了主机资源。当网络入侵突破防火墙的控制进入操作系统时,文件监控系统能够设置主客体的安全级别,对非法用户进程的请求进行阻断,通过比较主客体的安全标签来有效地保护系统数据,追踪攻击进程的相关信息并生成文件访问日志,能够很好地监视和抵御来自本地和网络上的攻击。文件安全监控系统加固了

Windows 文件内核的安全性,可大幅度提高网络安全等级和系统安全。将网络安全技术应用由单一的依靠防火墙、入侵检测等应用层网络防护转入内外结合的、更加安全的立体化防护体系,使网络安全“标本兼治”成为可能。

参考文献:

- [1] SAMARATI P, DE CAPITANI DI VIMERCATI S. Access Control Policies, Models and Mechanisms[A]. Foundations of Security Analysis and Design: Tutorial Lectures[C]. London, UK: Springer-Verlag, 2003, Vol 2171: 137 - 196.
- [2] BELL DE, LAPADULA L. Secure Computer Systems: Mathematical Foundations and Model[R]. Technical Report M 74 - 244, Mitre Corp, Bedford, MA, 1973.
- [3] 庄毅, 潘龙平, 刘坤. 分布式资源安全监控系统模型的研究[J]. 南京航空航天大学学报, 2006, 38(1): 90 - 94.
- [4] 潘龙平, 庄毅, 吴学成. 基于强制访问控制的安全 Linux 系统设计与实现[J]. 计算机工程与应用, 2006, 42(5): 142 - 145.
- [5] NAGAR R. Windows NT File System Internals[M]. O'REILLY, 1997.
- [6] BAKER A, LOZANO J. Windows2000 设备驱动程序设计指南[M]. 施诺, 等译. 北京: 机械工业出版社, 2001.
- [7] 刘海峰, 卿斯汉, 刘文清. 安全操作系统审计的设计与实现[J]. 计算机研究与发展, 2001, 38(10): 1262 - 1268.
- [8] (美) Internet Security Systems 公司. Windows 2000 安全技术参考[M]. 北京: 机械工业出版社, 2001.
- [9] SOLOMON DA, RUSSINOVICH ME. Inside Microsoft Windows 2000[M]. Microsoft Press, 2000.
- [10] WU SL, SHETH A, MILLER JA, et al. Authorization and Access Control of Application Data in Workflow Systems[J]. Intelligent Information Systems, 2002, 18(1): 71 - 94.

中国计算机学会

全国第二届语义 Web 与本体论学术研讨会 (SWON2007) 征文通知

一、征文范围(包括但不限于)

语义 Web 语言与工具; 语义 Web 知识表示; 语义 Web 知识管理; 语义 Web 推理; 语义 Web 服务; 语义 Web 安全; 语义 Web 挖掘; 语义信息标注; 语义检索和查询; 本体学习与元数据生成; 本体存储与管理; 语义集成和映射。

二、来稿要求

本次会议只接受 E-mail 投稿, 且中英文稿均可(Word 或 PDF 格式), 一般不超过 6000 字, 必须附中英文摘要、关键词、资助基金与主要参考文献, 注明作者及主要联系人姓名、工作单位、详细通信地址(包括 Email 地址)与作者简介。

投稿邮箱: 华中科技大学计算机学院

李瑞轩, 文坤梅

(swon2007@hust.edu.cn)

会务情况: 中国人民大学

杨楠, 姜芳芳(wisa2007@gmail.com)

大会网站: <http://www.ruc.edu.cn/wisa2007/>

<http://www.neu.edu.cn/wisa2007>

征文截止日期: 2007 年 4 月 1 日

录用通知发出日期: 2007 年 4 月 20 日

正式论文提交日期: 2007 年 5 月 10 日

会议召开日期: 2007 年 9 月 14 ~ 16 日