文章编号:1001-2486(2007)06-0054-05

基于用户意愿的文件访问控制策略*

何鸿君,罗 莉,曹四化,宁京宣,李 朋,董黎明
(国防科技大学 计算机学院,湖南 长沙 410073)

摘 要:访问控制是保护计算机上文件安全的重要技术手段。针对文件攻击,提出一种量化的评估方法,对主流访问控制策略进行了量化评估,指出主流访问控制策略的脆弱性在于赋予了程序访问用户能够访问的文件集合的权利。提出一种基于用户意愿的访问控制策略,其风险远远小于主流访问控制策略,能够防御未知文件攻击,证明了策略的安全性质,并讨论了其实现方案。

关键词:访问控制;用户意愿;文件攻击;恶意程序
中图分类号:TP309 **文献标识码**:A

A File Access Control Policy Based on User's Intention

HE Hong-jun, LUO Li, CAO Si-hua, NING Jing-xuan, LI Peng, DONG Li-ming
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Access control is an important technique to protect computer files. Aiming at malwares that attack files, the paper proposes a quantified estimation method, and points out that the fragibility of prevalent access control policies lies in authorizing programs to access files what the user can access. The paper novelly proposes an access control policy based-on user's intention, which is able to defend unknown file attacks, and has extraordinarily less risk than prevalent access control policies. Furthermore, the paper proves security properties of the policy presented, and its application is discussed.

Key words: access control; user's intention; file attack; malware

计算机系统面临的攻击中,窃取或破坏文件的文件攻击占据着主导地位。根据CERT/CC的统计,1999年由恶意程序带来的安全事故为9859起,而2003年已跃升到137 529起,且绝大多数的恶意程序以文件攻击为目的。因此,我们需要一种能够有效防御文件攻击的手段。

访问控制是保护计算机上文件安全的重要技术手段。访问控制机制可以限制用户、程序对信息资源的访问,阻止用户对信息的非授权访问[1-2]。以系统中处于不安全状态下的文件数量作为风险的量化评估标准,主流访问控制策略的风险值均为当前用户能够访问的文件数量。对于个人计算机,当前用户能够访问的文件通常是整个文件系统,即使对于服务器,当前用户能够访问的文件也是一个巨大的数量。因此,当系统遭受恶意程序攻击时,会造成巨大的损失。

本文提出一种量化的风险评估方法,对主流访问控制策略的风险进行量化评估,进而提出一种基于用户意愿的访问控制策略(Intension-based Access Control Policy,简称IBAC),重点讨论IBAC的风险、安全性质、现实意义和实现问题。

1 主流访问控制策略

当前主要的访问控制策略有自主访问控制(Discretionary Access Control,简称DAC)[3],强制访问控制(Mandatory Access Control,简称MAC)[1]和基于角色的访问控制(Role-based Access Control,简称RBAC[4-5])。

DAC允许对象的属主来制定针对该对象的保护策略,即拥有授予某种访问权利的主体(用户)能够

* 收稿日期:2007-06-13
基金项目:国家部委基金资助项目
作者简介:何鸿君(1968—),男,副教授,博士。

自己决定是否将访问控制权限的子集授予其他的主体或从其他主体那里收回他所授予的访问权限。通常 DAC 通过授权列表(或访问控制列表)来限定哪些主体针对哪些客体可以执行什么操作。DAC 的优点是用户有很大的灵活性,能适应现实世界的许多系统,使得它广泛成为各种操作系统和应用程序的访问控制策略选择。其主要问题是没有对信息流向控制提供保证,当用户自主地将其权限授予其他主体时,常常会导致安全级别高的客体信息流向安全级别较低的主体,从而破坏信息的机密性原则。

强制访问控制策略是指系统给主体和客体分配了不同的安全属性,用户不能改变自身或任何客体的安全属性,只有系统管理员可确定用户与用户组的访问权限。系统通过比较客体和主体的安全属性来决定主体是否可访问客体。MAC 提供了比 DAC 更强的安全保证,总能保证信息流是从低安全级别的实体流向高安全级别的实体,保证了信息的机密性。然而,这种强制性的后果是导致了其可用性较差,使得系统的授权管理等问题比较棘手。同时,对写访问的特殊要求使得合理的高安全级别主体向低安全级别的客体的写访问受到了限制。

RBAC 的研究比较热[6-10],其基本思想是在用户和访问权限之间引入角色的概念,将用户和角色联系起来,通过对角色的授权来控制用户对系统资源的访问。RBAC 与现实世界的许多系统相类似,能应用到现实中的许多系统中,保证了其良好的安全性质,系统授权管理等问题比较容易解决。然而,RBAC 无法在执行过程中对执行主体的权限进行精确的控制,主体执行过程中获得的权限仍是本次会话所激活角色的所有权限,而不是完成任务所需要的最小权限。

2 访问控制策略的评估

2.1 风险的量化评估

不同访问控制策略的安全性质不一样,采用它们的系统面临的风险不同。如何定量地度量这种风险大小呢? 这里提出一种量化评估方法,用处于风险状态的文件数量来表示策略的安全风险值。

定义 1 计算机系统采用了某种访问控制策略,设在某一时刻系统中面临窃取或者破坏危险的文件的数量为 R,则称 R 为该访问控制策略的风险。

一个程序是否存在可以利用的安全漏洞,或者是否存在故意设计的恶意,在目前的程序设计理论发展水平下,是没有办法做出肯定结论的。因此,如果一个程序在运行,该程序能够自主访问的文件就可能被窃取或者破坏。所谓自主访问,是指程序执行过程中访问什么文件、以什么模式访问文件,不需要用户的干预。这种认识可表述为以下的合理假设。

风险假设 活动程序能够自主访问的文件面临窃取或者破坏危险。

根据定义 1 和风险假设,访问控制策略的风险应该是动态变化的,即随着活动程序的数量变化以及程序能够自主访问的文件的数量变化在不断变化。这与人们的常识相符合,因为工作所运用的资源越多,系统面临的风险也就越大。

2.2 主流访问控制策略的风险分析

设系统当前登录的用户为 u,F_u 为用户 u 可以访问的文件集合,$|F_u|$ 为 F_u 中文件的数量,R_{DAC}、R_{MAC}、R_{RBAC} 分别为 DAC、MAC、RBAC 的风险值。

定理 1 $R_{DAC} = |F_u|$。

证明 因为 DAC 认为程序的行为代表用户的行为,所以活动程序能够自主访问的文件集合为 F_u。根据风险假设,有 F_u 面临风险,所以 $R_{DAC} = |F_u|$。证毕!

DAC、MAC、RBAC 均认为用户登录后,程序的行为就代表了用户的行为。因此,可以得到与定理 1 相同的结论。

定理 2 $R_{MAC} = |F_u|$, $R_{RBAC} = |F_u|$。

定理 1、2 很好地解释了为什么现在的计算机系统对文件攻击无能为力:访问控制策略赋予了程序访问用户能够访问的文件集合的权利,一旦恶意程序或者包含恶意代码的应用程序被运行,用户的所有文件就处在它的攻击范围之内,可造成巨大的损失。如果 u 是特权用户,F_u 就是整个文件系统,攻击的

危害将是毁灭性的。

3 IBAC

3.1 定义

IBAC的基本思想是:以用户(人)的意愿为本,程序只有获得用户的许可才准许访问相应的文件;如果程序的文件访问行为违背了用户的意愿,则拒绝执行。用户的意愿是由用户需要进行的工作明确决定的。例如用户正在运用Word处理文件 $d1$、$d2$,则用户的意愿就是只允许Word访问 $d1$、$d2$,如果Word试图访问 $d1$、$d2$ 以外的文件,则违背了用户意愿。用户授予程序某种访问权,程序可以长期保有这种访问权,也可以只是临时性的获得。对应的,将用户意愿区分为静态意愿和动态意愿。

定义2 如果用户授权程序 p 以模式 m 访问文件 f,其有效时间是永久性的,那么,用户意愿(p, f, m)称为静态意愿。

例如,用户希望系统启动后自动运行某日程安排程序MySchedule,那么其意愿"授权操作系统只读访问文件MySchedule"就是一种静态意愿。除非用户后来改变了这种授权,否则,每次系统启动时都会自动运行MySchedule。

定义3 如果用户授权程序 p 以模式 m 访问文件 f,其有效时间是直到文件关闭,那么,用户意愿(p, f, m)称为动态意愿。

用户与程序的交互过程中,其发布的文件访问授权通常是动态意愿。例如,用户操作程序Word编辑文档doc,用户希望的是当他授权Word打开doc时,Word才能访问doc,关闭doc后,Word不能访问doc。

定义4 IBAC表示为(UA, P, I, IS),其中,UA 是当前登录系统的用户,P 是程序集合,I 是用户意愿的集合,IS 是用户的静态意愿集合,它是 I 的初始值。将目录当作文件看待。$\forall p \in P$,IBAC遵循以下规则:

(1)如果 UA 授权 p 以模式 m 打开了文件 f,则 $I = I \cup \{(p, f, m)\}$;

(2)如果 p 关闭了文件 f,并且 $(p, f, m) \notin IS$,则 $I = I - \{(p, f, m)\}$;

(3)对于 p 发出的任意文件访问请求 $op(p, f, m)$,如果 $(p, f, m) \in I$,允许执行;否则,拒绝之。

3.2 安全性质

根据IBAC的规则(3),可以直接得到以下性质。

性质1 如果程序 P 的文件访问操作违背用户意愿,那么,P 被当场捕获。

该性质非常重要,它表明IBAC是一种主动防御技术,能够防御未知文件攻击。而采用传统的反病毒软件技术,通常是病毒出现后才能研制对应的查杀工具。

性质2 如果文件 f 被破坏,则 $\exists (p, f, write) \in I$。

证明 根据题设,有 $\exists p$ 发出的文件访问请求 $op(p, f, write)$ 被执行。

根据IBAC的规则(3),有 $(p, f, write) \in I$。证毕!

该性质很重要,它表明基于IBAC的计算机系统,恶意程序能够破坏的文件局限于用户意愿内的文件集合。传统计算机系统中,一旦恶意程序发作,它能破坏的文件集合往往是整个文件系统。对于文件窃取活动,有类似的重要结论。

性质3 如果文件 f 被窃取,则 $\exists (p, f, read) \in I$。

本性质还有另外一层重要意义。绝大多数的泄密事件都是在用户毫不知情的情况下,恶意程序偷偷地将信息窃取的,是一种被动方式的泄密。而主动泄密,是指用户明明知道文件 f 是涉密的,并且访问 f 可能造成泄密,却仍然操作程序访问 f。根据本安全性质,对于涉密信息,被动泄密将不会发生。

根据IBAC的定义,可直接得到以下性质。

性质4 如果用户不授权程序写文件 f,则对 f 的破坏性攻击是无效的。

计算机系统中,程序安装后是不可修改的,还有一些重要数据文件通常是不需要修改的。因此,按照IBAC,只要用户在使用计算机的过程中不授权程序"写"这些文件,这些文件就不会遭受破坏。这不

但为用户的重要信息提供了可靠的保护,而且还为整个计算机系统安全提供了有力的支撑。

IBAC不涉及文件的密级、权属,能够与DAC、MAC、RBAC等主流访问控制模型直接结合:在DAC(MAC、RBAC)的基础上,再实现IBAC。这种结合可以带来双重的安全性。

定理3 在DAC的基础上实现IBAC,系统能够同时得到IBAC与DAC的安全性质。

证明 根据题设,DAC的实现与IBAC无关,所以系统可获得DAC的安全性质。根据题设,IBAC在DAC的基础上实现,有$\forall(p, f, m)\in I$,DAC允许用户访问f,即I的可能值受到DAC的限定。

又对于IBAC来说,系统没有其他变化。

所以,系统可获得IBAC的安全性质。证毕!

还可以证明IBAC与MAC(或RBAC)相结合,能够同时得到IBAC与MAC(或RBAC)的安全性质。

3.3 风险分析

设系统当前登录的用户为u,I_u为用户u授权可以访问的文件集合,$|I_u|$为I_u中文件的数量,R_{IBAC}为IBAC的风险值。根据性质2、性质3,直接得到以下结论。

定理4 $R_{IBAC}=|I_u|$。

计算机运行过程中,F_u、I_u随时间变化。绝大多数情况下,用户拥有成千上万的文件,用户同时处理的文件数量往往是2、3个,$|F_u|\gg|I_u|$。因此,IBAC的风险远小于主流访问控制策略的风险。

3.4 恶意程序案例分析

实际的恶意攻击案例中,文件攻击占据的比例是多少?这能够说明IBAC的实用价值。2004年5月,我们对Trend Lab提供的2004年1月9日至2004年4月22日公布的一共151种恶意程序进行了分析。其中Worm病毒高达143例,Trojan Horse仅1例,其他病毒共7例。2006年1月,我们对Trend Lab提供的2005年12月5日至2006年1月8日公布的99种恶意程序资料进行了分析。其中Worm病毒达35例,Trojan Horse 29例,其他病毒共35例。分析发现,它们在执行过程中都包含了对文件系统的非授权访问操作,特别是大多都具有安装环节,在安装环节中修改注册表以保证在系统启动时能够自动运行。因此,理论上IBAC完全可以实时捕获这些恶意程序,避免损失。

4 IBAC的实现

4.1 系统结构

IBAC在Windows XP操作系统下的实现结构见图1,由静态授权模块、动态授权模块、文件访问监控模块、事件过滤模块和授权对话框等构成。静态授权模块完成用户静态意愿的授予、取消以及静态意愿信息的保存。动态授权模块完成用户动态意愿的授予,并将动态意愿信息传送给文件访问监控模块。事件过滤模块对静态授权模块、动态授权模块、授权对话框所处理的事件进行分析,过滤掉不是用户真实操作所产生的输入事件,确保授权是用户发出的。授权对话框是一个简单的对话框,询问用户是否授权程序访问某个文件,授予的权限属于动态意愿。文件访问监控模块监视所有的文件访问请求,如果是用户授权过的,则允许执行;否则,弹出授权对话框,询问用户是否授权程序p访问f。用户做出判断的原则很简单:如果用户需要处理文件f,就授权;否则,拒绝授权。

4.2 动态授权模块

静态意愿具有长期稳定的特点,用户使用计算机过程中,很少需要进行新的授权或者取消原有的授权,对用户使用计算机几乎没有影响。动态授权是用户在使用计算机过程中临时授予程序的短暂性权限,文件关闭、程序退出或者关机后权限不再存在。这就意味着,每当程序需要新访问一个文件,用户就要进行授权,授权的实现方式对用户使用计算机有很大的影响。

为减少对用户正常工作的影响,采取工作区方式实现动态授权。其基本思想是:如果设立一个工作区,则程序可以自主访问工作区中的文件。系统中可以有多个工作区同时存在。工作区的类型可以根据需要自行定义:(1)安全工作区是一个目录(不包括子目录),程序可以自主读写该目录下的文件(必须

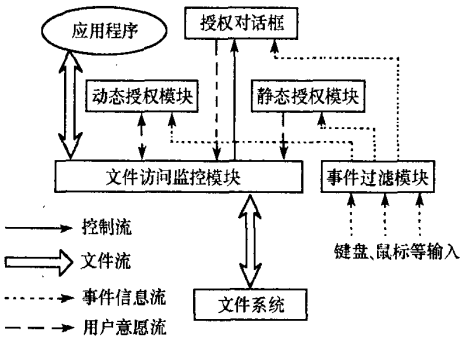

图1 实现结构

Fig.1 Implementation architecture

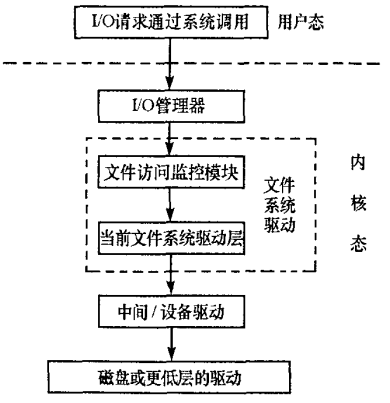

图2 文件访问监控模块的位置

Fig.2 Position of the file access monitor

遵循文件的访问模式,不能写只读文件或者读只写文件)。(2)普通工作区是一个目录(不包括子目录),程序可以自主读写该目录下的文件(必须遵循文件的访问模式,不能写只读文件或者读只写文件),或者创建/删除文件。(3)危险工作区是一个目录(包括子目录),程序可以自主读写该目录(包括子目录)下的文件(必须遵循文件的访问模式,不能写只读文件或者读只写文件),或者创建/删除文件或目录。

动态授权模块的界面与资源管理器类似,可以浏览文件。其设计有以下特点:(1)不同颜色突出了不同工作区的性质。(2)设定/取消工作区操作简洁。鼠标双击展开后的目录图标,设定目录为普通工作区;再次双击,取消工作区。其他类型的工作区,通过下拉菜单/弹出菜单进行设定/取消。(3)提供安装选项,仅仅支持普通工作区,用户培训工作简单。

工作区方式是IBAC的一种近似实现,具体表现在两个方面:(1)扩大了允许程序访问的文件集合。从工作实际看,用户很可能只需要让程序访问工作区的几个文件,而不是所有文件。(2)授权的有效时间变长了,在工作区取消前程序能够访问其中的文件。而动态意愿认为文件关闭后就不可再访问,除非再次授权。尽管如此,系统的风险被限定在工作区范围,远小于主流访问控制策略的风险,并且具有较好的可用性。

4.3 访问监控模块

文件访问监控模块在内核态下实现,是文件系统驱动的一部分,位于I/O管理器之下、当前文件系统驱动层之上,是文件系统驱动层次中的最上层。当前文件系统驱动层是指系统中已经存在的文件系统驱动组件,参见图2。对于每一个文件访问系统调用,I/O管理器都会构造一个相应的I/O请求包,文件系统驱动中则有相应的处理过程。通过截获I/O请求包,文件访问监控模块能够监视所有程序的文件访问请求。

5 下一步工作及总结

第4节给出的IBAC实现方案还不完善,仍有以下问题需要进一步研究:(1)研究可用性更好、严格遵循IBAC的动态授权方法。动态授权模块给出的授权方案并没有严格遵循IBAC,本质上扩大了需要授权的范围,这就意味着更多的文件面临风险。(2)给出的实现方案适合于保护用户文件,而对系统文件、应用程序文件的保护并不合适。(3)直观上看,IBAC比较适合于个人计算机,能否应用到服务器上呢?这也是我们非常感兴趣的。

本文的主要贡献在于:针对文件攻击,提出了一种量化的评估方法;对主流访问控制策略进行了量化评估,指出其脆弱性在于赋予了程序访问用户能够访问的文件集合的权利;提出了一种基于用户意愿的访问控制策略,该策略的风险远远小于主流访问控制策略,能够防御未知文件攻击;给出了IBAC的一种近似实现方案,具有较好的可用性。

(下转第80页)

表2 PE_EMRA算法与SRA算法比较数据

Tab.2 The result comparison of PE_EMRA and SRA

比较项目	SRA算法	PE_EMRA算法
调用资源优化分配算法的次数	200次	47次
拒绝用户请求的次数	44次	44次
实际调整资源配置的次数	156次	3次

综上所述,与SRA算法相比较,PE_EMRA算法在减少系统控制开销的同时,服务质量性能没有显著变化。基于预先判断策略的PE_EMRA算法优于SRA算法。

5 结束语

针对不同丢失率要求的业务类提出PE_EMRA模型与算法,为各应用类合理分配带宽和缓冲,提高整个资源的利用率。与相关算法比较,PE_EMRA算法在提供强有力的服务质量保证的同时又最大化资源利用率,并且具有配置灵活、实现代价小等特点。在今后的研究中,我们将进一步研究服务价格、丢失率限制和资源分配之间的关系,并考虑在网络层具体实现PE_EMRA资源分配算法。

参考文献:

[1] Odlyzko A. The Economics of the Internet: Utility, Utilization, Pricing, and Quality of Service [R]. DIMACS Technical Report 99-08, 1999.

[2] Zhang L, Deering S, Estrin D, et al. RSVP: A New Resource Reservation Protocol[R]. IEEE Network, Sept. 1993.

[3] Odlyzko A. Paris Metro Pricing: The Minimalist Differentiated Services Solution [EB/OL]. Available at http:www.research.att.com/~amo, 1999.

[4] Varian H R, Mackie-mason J K. Pricing the Internet [C]//Public Access to the Internet, 1995:1-19.

[5] Martin M. Declarative Scheduling for Optimally Graceful QoS Degradation[R]. Tohoku University, Tech. Rep.: CSE_TR 260-95, 1995.

[6] Kelly F P. Charging and Rate Control for Elastic Traffic [J]. European Transactions on Telecommunications, 1997,8(1):33-37.

[7] Semret N. Market Mechanisms for Network Resource Sharing [D/OL]. PhD Thesis, Columbia University, http://comet.columbia.edu/~nemo/work.html, 1999.

[8] Sairamesh J, Ferguson D F, Yemini Y. An Approach to Pricing, Optimal Allocation, and Quality of Service Provisioning in High-speed Packet Networks [C]//Proc. IEEE Infocom'2000, Boston, MA, Apr. 2000.

[9] [美]哈尔·瓦里安. 微观经济学:高级教程(第三版)[M]. 周洪,等译.北京:经济科学出版社,1997.

(上接第58页)

参考文献:

[1] Sandhu R S, Samarati P. Access Control: Principles and Practice[J]. IEEE Communications Magazine, 1994,32.

[2] Hu V C, Ferraiolo D F, Kuhn D R. Assessment of Access Control Systems[R]. Interagency Report 7316, Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930, September 2006.

[3] Pfleeger C P, Security in Computing. Second Edition[M]. Upper Saddle River: Prentice Hall, Inc., 1997.

[4] Sandhu R S, Coynek E J. Role-based Access Control Models[J]. IEEE Computer, 1996,29(2).

[5] Ferraiolo D F. Proposed NIST Standard for Role-based Access Control[J]. ACM Transactions on Information and System Security, 2001,4(3).

[6] Ferraiolo D, Kuhn R. Role-based Access Control[C]//Proceedings of 15th National Computer Security Conference, 1992.

[7] Barkley J. Implementing Role Based Access Control Using Object Technology[R]. First ACM Workshop on Role-based Access Control, 1995.

[8] Barkley J. Comparing Simple Role Based Access Control Models and Access Control Lists[R]. Second ACM Workshop on Role-based Access Control, August 11, 1997.

[9] Barkley J F. Supporting Relationships in Access Control Using Role Based Access Control[R]. Fourth ACM Workshop on Role-based Access Control, July 29, 1999.

[10] Gallaher M P, O'Connor A C, Kropp B. The Economic Impact of Role-based Access Control[R]. RTI Project Number 07007.012, March 2002.