ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.6, June 2007, pp.1501−1509
DOI: 10.1360/jos181501

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

对两个改进的 BLP 模型的分析*

何建波[1,2,4+]，卿斯汉[2,3,5]，王 超[2,3,4]

[1](中国科学院 软件研究所 基础软件国家工程研究中心,北京 100080)

[2](中国科学院 软件研究所 信息安全工程技术研究中心,北京 100080)

[3](北京中科安胜信息技术有限公司,北京 100086)

[4](中国科学院 研究生院,北京 100049)

[5](北京大学 软件与微电子学院,北京 102600)

Analysis of Two Improved BLP Models

HE Jian-Bo[1,2,4+], QING Si-Han[2,3,5], WANG Chao[2,3,4]

[1](National Engineering Research Center for Fundamental Software, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[2](Engineering and Research Center for Information Security Technology, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[3](Beijing ZhongkeAnsheng Corporation of Information Technology, Beijing 100086, China)

[4](Graduate School, The Chinese Academy of Sciences, Beijing 100049, China)

[5](School of Software and Microelectronics, Peking University, Beijing 102600, China)

+ Corresponding author: Phn: +86-10-62624525 ext 6006, E-mail: jianbo03@iscas.cn

He JB, Qing SH, Wang C. Analysis of two improved BLP models. *Journal of Software*, 2007,18(6):1501−1509.
http://www.jos.org.cn/1000-9825/18/1501.htm

Abstract: The security and flexibility are two goals that various improved BLP models attempt to achieve. How to enhance the flexibility of BLP model is a challenging problem that security researchers try to solve. However, the implementation of an insecure “security model” in the system will result in an insecure system. In this paper, two improved BLP models, for short DBLP (dynamic BLP) and SLCF (security label common framework), are analyzed. Although the designers of the two models claimed that their proposals can adjust the security level of the untrusted subject dynamically and accordingly improve the flexibility of the classical BLP model, the analytic results show that the two improved models are not secure at all. Under the rules of the two improved models a Trojan horse can “legally” read the high-level information and then write them to low-level objects, which violate the principle of multi-level security (MLS). This effort provides a theoretical foundation for avoiding the choice of insecure MLS model.

Key words: security; flexibility; BLP model; Trojan horse; information flow

* Supported by the National Natural Science Foundation of China under Grant No.60573042 (国家自然科学基金); the National Basic Research Program of China under Grant No.G1999035802 (国家重点基础研究发展计划(973)); the Beijing Natural Science Foundation of China under Grant No.4052016 (北京市自然科学基金)

Received 2005-12-16; Accepted 2006-05-18

摘 要: 安全性和灵活性是各种改进的 BLP 模型追求的目标.如何在保持安全性的前提下增加 BLP 模型的灵活性,一直是安全操作系统研究人员研究的重点.安全模型是系统设计的基础,如果在系统中实现了不安全的"安全模型",其后果是严重的.结合多级安全(MLS)的核心思想,通过实例列举的方式深入分析了两个改进的 BLP 模型——DBLP(dynamic BLP)和 SLCF(security label common framework).尽管这两个模型都提出了在系统运行时动态地调整主体安全级的规则,但是分析表明,它们还是不安全的.在这两个模型的规则控制下,特洛伊木马可以通过显式地读和写操作将高安全等级的信息泄漏给低安全等级的主体,从而违反了多级安全(MLS)策略.研究结果为人们避免选用不安全的模型提供了有意义的理论支持.

关键词: 安全性;灵活性;BLP 模型;特洛伊木马;信息流
中图法分类号: TP309 文献标识码: A

Bell-LaPadula 模型[1,2](简称 BLP 模型)是实现多级安全(multilevel security,简称 MLS)机密性策略的经典模型.MLS 的目的是防止高密级的信息泄漏给低密级的主体.为了实现 MLS 策略,BLP 模型定义了两个安全属性,即简单安全性(ss-属性)和限制性安全性(*-属性).ss-属性模拟了现实世界的情况,即主体不能读敏感级*比自己高的客体中的信息.*-属性的引入则主要是为了解决特洛伊木马问题[3].*-属性要求主体只能写敏感级高于自己的客体.BLP 模型的两条规则合在一起,保证了系统通过控制主体的读和写操作不会引起信息从高向低的流动.

尽管 BLP 模型能够很好地防止信息的非授权泄漏,保护信息的机密性,但是它仍存在如下两个不足:

1) 模型缺乏对可信主体访问权限的限制;

2) 模型的安全依赖于平稳性规则(principle of tranquility),即主客体敏感级在整个生命周期中不可变.

Bell[4]指出,用*-属性限制所有主体的系统是不实用的.对于一些违反 BLP 模型*-属性安全规则,为了保障系统的有效运行,不得不存在的操作如高敏感级主体向低敏感级主体发送非密级消息,系统通过定义可绕过*-安全属性规则检查的可信主体(trusted subject)**,如信息降级进程 *downgrader* 来完成这类操作.早期的高安全等级操作系统如 Multics[5],ASOS[6,7]和 SCOMP[8]等,均采用这种方式来实现 BLP 模型.然而,在这些 BLP 模型的实现中,可信主体是完全可信的,其行为完全在 BLP 模型的控制范围之外.尽管这样简化了模型在系统中的实现,但是,让一个可信主体完全可信地绕过强制访问控制机制检查的做法有悖于最小特权原则.为了对可信主体的行为进行限制,Bell 在文献[9]中把 BLP 模型推广到网络包交换数据的保护时,首次提出了用敏感级范围来限制可信主体读写范围的思想.随后,Schell 等人在开发 GEMSOS[10]时提出了多级主体的概念,TMach[11]及其后继项目 Fluke[12]中引入了可信主体敏感级范围的概念,IBM 开发的智能卡项目[13]和著名的 SELinux[14]项目在实现 MLS 策略时,也沿用了这一设计思想.本质上,这些改进的 BLP 模型都是相同的,都是为可信主体设定了一个读写范围,从而使可信主体只是部分可信的.这样,即使可信主体被恶意代码或病毒所感染,可信主体造成的危害也是有限的.为了具体地刻画部分可信主体,这些系统为主体定义了两个敏感级函数 a_min 和 v_max.从 a_min 到 v_max 之间的范围被称为可信主体的可信范围.在这些系统中,a_min 和 v_max 不等的主体被定义为可信主体,而非可信主体则是系统中 a_min 和 v_max 相等的主体.

尽管 GEMSOS,TMach 和 Fluke 等系统通过限制读和写安全级范围来限制可信主体的权限,在一定程度上克服了 BLP 模型的第 1 个不足.但是,这些改进的 BLP 模型只是限制了可信主体的可信范围,对于系统中大量存在的非可信主体(通常是普通用户进程),仍然严格满足 BLP 模型的规则.而且在安全操作系统开发的历史上,无论是严格实施 BLP 模型的系统,如 Multics,ASOS,SCOMP 等,还是通过限制可信主体来改进 BLP 模型的系统,如 GEMSOS,TMach,Fluke 和 SELinux 等,都把平稳性原则作为系统实现机密性保护的基本准则.平稳性原则要

* 在有关 BLP 模型的文献中,敏感级、安全级和安全标记通常具有相同的含义,本文延续了这一约定,在文中后面的部分将不区分这 3 个术语.

** 这里的可信与非可信是相对于进程是否允许超越 BLP 模型的*-属性来定义的.在文献[5]的 BLP 模型的形式化描述中,BLP 模型在进行访问仲裁时,先判断进程是否可信.如果是可信进程,则绕过 BLP 模型*-属性的限制,否则,根据安全级判断进程的访问权限.此外,用户可信和进程可信是两种不同的概念.代表可信用户执行操作的进程不一定是可信的,有可能隐藏特洛伊木马[3].

求在系统运行过程中,主体和客体的安全级始终保持不变.在 BLP 模型中严格执行平稳性原则主要有两个原因:1) 如果一个客体的安全级发生改变,那么已经被许可的访问(在 BLP 模型中即是在 b 中的访问元组)将不再满足安全状态定义的条件.如果把这些已经许可的访问都移走,则有可能产生隐通道[3];2) 正如 McLean 在文献[15]中所指出的,如果在系统运行过程中,由用户在满足 BLP 模型的 ss-属性和*-属性的基础上,自动地调节系统主体和客体的安全级,那么系统是不安全的,他列举了 System *Z* 的例子来说明这一问题.尽管严格执行平稳性原则可以保证模型的安全性,但是这样往往造成对非可信主体限制过强,限制了 BLP 模型的实用性.如果通过一定的强制规则在保持模型安全性的前提下能够实现对主体安全级的动态调节,则这种改进的 BLP 模型可以更好地满足实际应用的需要.

因此,如何在保持 BLP 模型安全性的前提下动态地调整非可信主体的安全级(也称敏感标签),从而增加 BLP 模型的灵活性,就成为安全操作系统研究人员研究的目标.Ott 在其硕士论文[16]中提出了一种改进的 BLP 模型,通过引入 Auto-write,Auto-read 和 Auto-read-write 规则,使得在某些前提条件下可以动态地调整系统中非可信进程的当前安全级.目前,Ott 已在 RSBAC 系统的 MAC 模块中实现了该模型[17].文献[18]提出了类似于文献[16]的改进 BLP 模型,简称为 ABLP(adaptive BLP)模型.ABLP 模型实施的中心思想是,把访问控制判断空间分成外层空间和内层空间,外层空间执行 BLP 模型静态敏感标记的判断,当在外层判断空间发现不满足常规判断条件的要求时,试图进入内层判断空间作进一步的判断.ABLP 模型允许一个主体在其生命周期内有条件地动态调整当前敏感标签.本质上,ABLP 与 Ott 的模型规则是一样的,都是根据主体进程最大读打开文件敏感级和最小写打开文件的敏感级历史来调整主体的敏感级标记.文献[19–21]也分别提出了两个改进的 BLP 模型(文献[19]将 SLCF(security label common framework)称为框架,当其用于实现机密性保护时,可被看作一种改进的 BLP 模型),分别简称为 SLCF 和 DBLP(dynamic BLP).SLCF 通过记录主体的访问历史来动态地调整主体的当前安全级.DBLP 模型则用主体的安全级范围取代了主体的当前安全级,并指出,不使用当前安全级的概念,也可以实现 Ott 的动态安全级调整规则.

尽管能够有效地调整进程的当前敏感标记将极大地增加系统的灵活性,但是,这往往会使模型的安全性受到极大的威胁,因为敏感标记的动态调整有可能导致机密性策略的失控.任何一种改进的 BLP 模型都应该把安全性作为首要考虑要素,其次才是灵活性.本文结合多级安全的核心思想,深入分析了 DBLP 和 SLCF 的安全性.分析结果表明,尽管 SLCF 和 DBLP 模型从主体标记动态调整方面改进了 BLP 模型,增加了模型的灵活性,但这两个模型都是不安全的.被 BLP 模型所禁止的信息流***在他们的模型中是“合法的”.在这两个模型的限制性条件下,特洛伊木马可以“合法”地通过显式地读写操作将高安全等级的信息泄漏给低安全等级的主体.

为了分析 DBLP 模型和 SLCF 框架,我们在第 1 节首先深入分析 MLS 策略的核心思想,简要介绍 BLP 模型实现 MLS 的方式,并引了多级客体当前安全级的概念.在第 1 节讨论的基础上,第 2 节和第 3 节采用列举实例的方式,分别分析 DBLP 模型和 SLCF 框架的安全性,分析结果将表明这两个改进的模型都是不安全的.最后是对分析结果的总结.

1 多级安全策略(MLS)和 BLP 模型

20 世纪 60 年代,MLS 思想最初起源于美国国防部对保护计算机中的机密性信息的迫切要求,因此,MLS 策略又称为军事安全策略[22].MLS 策略的目的是防止高密级信息泄漏给低密级的主体(进程).为了形式化地描述 MLS 策略,假设系统中客体 o 的敏感级为 $level(o)$,主体的敏感级为 $level(s)$,则一个系统状态 $state_t$ 满足 MLS 当且仅当对于任意的非可信主体 s,若$(s,o_1,r)\in b_t$,$(s,o_2,a)\in b_t$,则有 $level(o_1)\leq level(o_2)$,即在一个满足 MLS 策略的系统中,非可信主体 s 对客体的读或写访问不会引起信息从高安全级向低安全级的流动.这实际上等同于某种形式的信息流模型[3].这里的 b_t 表示状态 $state_t$ 中的当前访问集,$(s,o,x)\in b_t$ 表示 b_t 中允许主体 s 以方式 x 访问客体 o,

*** BLP 模型主要是禁止主体通过读取和修改(写)客体的内容引起的非法信息流动,其目的是防止高安全级客体中的信息通过显式地读和写操作流向低安全级客体.正如 Gasser 在文献[3]中所指出的,隐蔽信道产生的信息流不在 BLP 模型的控制内.

这里的记号与 BLP 模型[5]中的符号是一致的.

BLP 模型通过两条安全公理(安全属性)实现了 MLS 策略:简单安全性(ss-属性)和限制性属性(*-属性).前者保证了主体只能读安全级不高于自己的客体,即如果$(s,o,r)\in b_t$,则 $level(o)\leq level(s)$;后者则限制了主体只能对安全级不低于自己的客体执行盲写(append)操作,即如果$(s,o,a)\in b_t$,则 $level(s)\leq level(o)$.因此,若在某一安全状态 $state_t$ 下,有$(s,o_1,r)\in b_t,(s,o_2,a)\in b_t$,则有 $level(o_1)\leq level(s),level(s)\leq level(o_2)$,由安全级支配关系"≤"的传递性,有 $level(o_1)\leq level(o_2)$,因此,BLP 模型实现了 MLS 策略.

对于 BLP 模型,由于每个客体只有一个敏感级,因此,通常认为客体的敏感级就是客体中所含信息的敏感级.但是对于一个具有敏感级范围(表示为 $ran(o)$)的多级客体而言,客体允许包含敏感级在客体敏感级范围 $ran(o)$内的信息.因此对于多级客体,我们定义一个敏感标记 $level_c$,$level_c(o)$表示多级客体 o 当前所含信息的敏感级,且有 $level_c(o)\in ran(o)$.相应地,MLS 策略可以形式化地描述为:对于任意的非可信主体 s,若$(s,o_1,r)\in b_t$,$(s,o_2,a)\in b_t$,则 $level_c(o_1)\leq level_c(o_2)$.我们将基于 MLS 的形式化表述来分析 DBLP 模型和 SLCF 框架.

2 DBLP 模型的分析

2.1 DBLP模型简介

文献[20,21]提出了一个改进的 BLP 模型(DBLP).在 DBLP 模型中,对主体敏感级采用了与文献[11]类似的处理方式,主体不再具有当前敏感级,而是代之以最小可写敏感级 a_min_s和最大可读敏感级 v_max_s.这样, DBLP 模型中主体就有 3 个敏感级标签:f_s(主体最大敏感级标签),a_min_s 和 v_max_s.同时,客体具有两个敏感级标签 L_min_o和 L_max_o.DBLP 模型定义了单级客体和多级客体,但是 DBLP 模型的多级客体的概念与 Fluke 微内核[12]中实现的改进 BLP 模型中定义的多级客体概念不同.在 Fluke 微内核中,多级客体是指系统中 $L\_min_o\neq L\_max_o$ 的客体,多级客体只能由可信主体访问,在系统实现中对应为可信主体的私有数据或者可信进程间共享变量;而在 DBLP 模型中,即使客体的 $L\_min_o\neq L\_max_o$,如果主体对客体的访问是以客体安全级范围内的一个安全级决定,那么该客体仍是单级客体.因此,在 DBLP 模型中,客体的单级特性和多级特性与对该客体的操作密切相关.文献[20]的定义 1 给出了 DBLP 模型单级客体和多级客体的定义.

定义 **1**[20,21]. 如果一个客体 o 满足 $L\_min_o=L\_max_o$,或者 $L\_min_o\neq L\_max_o$,但是主体每次对客体 o 的访问仅由该客体的安全标签范围内的一个安全级确定,这样的客体 o 被称为具有单级特性,此时的客体简称为单级客体,记为 $single(o)$;如果一个客体 o 满足 $L\_min_o\neq L\_max_o$,且主体每次对客体 o 的访问由该客体的整个安全标签范围确定,此时,安全标签范围不仅表明客体所包含数据的最小安全级及最大安全级,而且还表明对客体的某些完整性限制,这样的客体 o 被称为具有多级特性,简称为多级客体,记为 $multi(o)$.一个主体 s 称为可信主体,如果它的安全级无论从何种状态出发,在策略执行的过程中都满足如下条件:

$f_s^*(s)=f_s(s)$,

$v\_max_s^*(s)=v\_max_s(s)$,

$a\_min_s^*(s)=a\_min_s(s)$.

文献[20]指出,单级客体是可信主体和非可信主体共同访问的对象,而多级客体仅是可信主体访问的对象.非可信主体可以访问 $L_min_o(o)\neq L_max_o(o)$的客体,只要非可信主体对该客体的访问仅由该客体安全级范围的一个安全级确定即可.然而,我们认为,定义 1 中主体对单级客体的访问的定义是模糊不清的.定义 1 没有指出非可信主体在访问 $L_min_o(o)\neq L_max_o(o)$的单级客体时,如何确定定义中所提到的客体安全级范围内的那个安全级.而且在随后定义的模型限制性规则 CT_1 中,访问监控器(reference monitor)在仲裁非可信主体对 $L_min_o(o)\neq L_max_o(o)$的单级客体访问时,也没有把定义 1 中提到的客体安全级范围内的那个安全级纳入裁决过程中,而是以单级客体的最大和最小安全级来决定主体对客体的访问权.后面的分析表明,正是这一不足,造成了 DBLP 模型安全性的失控.DBLP 模型的描述详见文献[20,21],本文不再复述.

2.2 对DBLP模型的分析

2.2.1 DBLP 模型 CT_1 规则的化简

为了分析 DBLP 模型的安全性,我们首先对 DBLP 模型中复杂的限制性规则 CT_1 进行化简.由于我们只考虑非可信主体的行为,因此,下面的化简规则仅相对于 DBLP 模型中与非可信主体有关的不变量和限制性条件.

首先,经过分析发现,可以对 DBLP 模型复杂的限制性规则 CT_1 进行化简,有关的 CT_1 详细表述可参阅文献[20].

对于 CT_1 的规则(1)[20],只要$(s,r)\in M(o)$,$single(o)$,且 $L\_min_o(o)\le v\_max_s(s)$,主体 s 对 o 的读访问请求就被许可.相应地,s 的最小写敏感级 $a_min_s^*(s)$调整为 $L_min_o(o)$和 $a_min_s(s)$中的最大值.同理,CT_1 的规则(2)[20]也可以进行相应的调整.我们把经过简化后的限制性条件定义为 CT_1',其规则如下:

(1′) 若$(s,r)\in M(o)$,且 $single(o)$,只要 $L\_min_o(o)\le v\_max_s(s)$,就有:

$b^*=b\cup(s,o,r)$,

$a\_min_s^*(s)=\max\{L\_min_o(o),a\_min_s(s)\}$,

主体和客体的其他敏感级标签均保持不变.

(2′) 若$(s,a)\in M(o)$,且 $single(o)$,只要 $a\_min_s(s)\le L\_max_o(o)$,就有:

$b^*=b\cup(s,o,a)$,

$v\_max_s^*(s)=\min\{L\_max_o(o),v\_max_s(s)\}$,

主体和客体的其他敏感级标签均保持不变.

(3′) 若$(s,w)\in M(o)$,且 $single(o)$,则

状态变迁既满足$(s,r)\in M(o)$,$single(o)$的情况,又满足$(s,a)\in M(o)$,$single(o)$的情况.

2.2.2 DBLP 模型安全性的分析

作为一个改进的机密性模型,必须在保证模型安全性的基础上再考虑增加灵活性.但是分析表明,DBLP 模型是不安全的,非可信主体(特洛伊木马)可以在不违反 DBLP 模型规则的条件下,通过读、写操作,造成信息从高级别客体流向低级别主体.我们通过实例来说明 DBLP 模型是不安全的.

由于 DBLP 模型没有提供表达 $L_min_o(o)=L_max_o(o)$的客体当前所含的信息安全级的方式,我们用前面定义的安全级函数 $level_c$ 来表示这类客体 o 中当前信息的安全级.根据 DBLP 模型的定义,对于客体 o,如果 $L\_min_o(o)\ne L\_max_o(o)$,则 $level_c(o)\in ran(o)$;如果 $L\_min_o(o)=L\_max_o(o)$,则 $L\_min_o(o)=L\_max_o(o)=level_c(o)$.

例 1:假设系统设置了 4 个安全级,由高到低分别为 *top secret*,*secret*,*confidential* 和 *unclassified*.系统有一个非可信进程(特洛伊木马)*process*1,初始时,$v\_max_s(process1)=top\ secret$,$a\_min_s(process1)=confidential$.系统中有两个单级客体 *file*1 和 *file*2,其中 $L\_min_o(file1)=confidential$,$L\_max_o(file1)=top\ secret$,$L\_min_o(file2)=L\_max_o(file2)=confidential$.文件 *file*1 当前包含密级 $level_c(file1)$为 *secret* 的信息.由 DBLP 模型的定义,$secret\in ran(file1)$,因此,*file*1 包含安全级为 *secret* 的信息是允许的.假设系统当前处于安全状态,即满足 DBLP 模型的不变量条件.下面我们分析,当特洛伊木马 *process*1 首先向系统中实现 DBLP 模型的访问监控器提出读 *file*1 的访问请求,再提出盲写 *file*2 的访问请求后,系统中信息的流动情况:

1) *process*1 提出对 *file*1 的读访问请求,且满足$(process1,file1)\in M(file1)$,$single(file1)$.因为 $L\_min_o(file1)=a\_min_s(process1)=confidential$,根据 DBLP 模型的 CT_1' 规则(1′),*process*1 对 *file*1 的读访问请求被许可,因此,进程 *process*1 就读取了 *file*1 中安全级为 *secret* 的信息,且 *process*1 的敏感级 a_min_s 调整为

$$a\_min_s^*(process1)=\max\{L\_min_o(file1),a\_min_s(process1)\}=confidential.$$

2) 然后,*process*1 提出对文件 *file*2 的盲写访问请求,且满足$(process1,file2)\in M(file2)$,$single(file2)$.$L\_min_o(file2)=a\_min_s(process1)=confidential$.根据 DBLP 模型的 CT_1' 规则(2′),*process*1 对 *file*2 的写访问请求被许可,因此,*process*1 把在 1)中读取的密级为 *secret* 的信息写入密级为 *confidential* 的文件 *file*2 中,这样,密级为 *confidential* 的 *process*2 就可以通过读文件 *file*2 获取密级为 *secret* 的信息,从而造成了信息由高向低的流动.

图 1 显示了在 DBLP 模型规则限制下,特洛伊木马引起的非法信息泄漏.图中实线箭头表示实际的读和盲写操作引起的信息流,虚线箭头表示特洛伊木马的读/盲写操作实际产生的违反 MLS 策略的信息流.

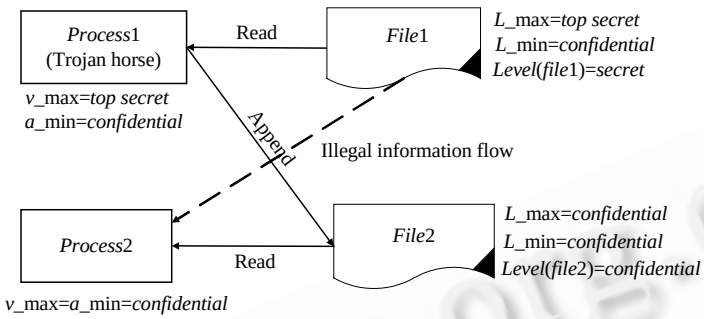

Fig.1 The illegal information flow caused by Trojan horse under the DBLP rules

图 1 在 DBLP 规则下,特洛伊木马引起的非法信息流

2.2.3 DBLP 安全失控原因分析

造成 DBLP 模型机密性安全失控的主要原因是,DBLP 模型允许非可信主体访问 $L\_min_o \neq L\_max_o$ 的客体.但是,在模型的限制性规则中,访问监控器仲裁非可信主体对这类客体的读访问操作时依据的是主体的敏感级 v_max_s 和客体的敏感级 L_min_o,而不是根据客体所含信息的敏感级 $level_c$ 来判断的.由于这类客体中所含信息的敏感级不低于 L_min_o,从而造成非可信主体通过读操作"隐式地"降低了信息的敏感级,并在 DBLP 模型的规则约束下,造成信息从高往低"合法地"流动.

另外,需要指出的是,文献[20]中的定理 5.1 对 DBLP 模型的安全性进行了证明.但是,定理 5.1 只证明了非可信主体在读写 $L\_min_o=L\_max_o$ 的单级客体时不会发生信息从高到低的流动,但是没有证明非可信主体在访问 $L\_min_o \neq L\_max_o$ 的单级客体时不会出现违反 MLS 信息流的情况.因此,定理 5.1 的证明是不完备的.

3 SLCF 安全标记公共框架中改进 BLP 模型的分析

文献[19]针对 BLP 模型不灵活的特点,对 BLP 模型进行了改进,提出了一个可动态调节主体安全标记(即主体敏感级)的改进的 BLP 模型,文献[19]将其称为安全标记公共框架(SLCF 框架).尽管 SLCF 克服了原 BLP 模型主体标记固定不变的不足,给出了在系统运行过程中动态调整主体标记的规则,但是,与 DBLP 模型一样,改进的模型无法限制非可信主体读取高密级信息,然后写入低密级客体中这一违反 MLS 策略的信息流.因此,SLCF 安全标记框架也是不安全的.我们首先简要描述 SLCF 框架,然后再对其安全性进行分析.为了便于分析,本节的术语和符号将尽量与文献[19]保持一致,其中标记与前面提到的安全级和敏感级具有相同的意义.

3.1 SLCF框架简介

SLCF 在 BLP 模型的基础上,做了如下改进:

1) 除主体的两个标记函数 f_c 和 f_s 以外,还为主体增加了 4 个标记函数(f_{il},f_{ih},f_{ol},f_{oh}),它们分别表示在一个进程的生命周期内流入信息的最低标记、流入信息的最高标记、流出信息的最低标记、流出信息的最高标记.这 4 个函数主要用于记录主体的访问历史;
2) 为客体增加了一个标记函数 f_d,f_d 把客体标记 f_o 映射为易于理解的信息(一般是一组字符串),用于客体信息的输出(如显示或发布信息).

SLCF 实现多级安全策略的方法如下:首先,主体初始化时标记的初始值为 $f_{il}=f_{ih}=LOW$(系统的最小标记值),$f_{ol}=f_{oh}=HIGH$(系统的最大标记值).接下来用与 C 语言相似的语法来描述在多级保密性安全策略下,信息流动时所需的安全判定条件以及主体标记的变化规则.为了方便描述,假设信息源的标记值表示为 f_o,$T(o,s)$=true,表示允许信息从 o 流入 s,$T(o,s)$=false 表示禁止信息从 o 流入 s.

SLCF 访问控制规则:

(1) 在信息从客体 o 流入主体 s 时(如 s 读取 o),判定条件及主体标记的变化规则(规则 1)如下:

IF ($f_c \geq f_o$) THEN $T(o,s)$=true;

ELSE IF ($f_s \geq f_o$) && ($f_{ol} \geq f_o$) THEN {f_c=Max(f_c,f_o),f_{ih}=Max(f_{ih},f_o),$T(o,s)$=true};

ELSE $T(o,s)$=false.

(2) 在信息从主体 s 流入客体 o 时(如 s 写入 o),判定条件及主体标记的变化规则(规则 2)如下:

IF ($f_o \geq f_c$) THEN $T(s,o)$=true;

ELSE IF ($f_o \geq f_{ih}$) THEN {f_c=Min(f_c,f_o),f_{ol}=Min(f_{ol},f_o),$T(s,o)$=true};

ELSE T(s,o)=false.

(3) 当信息在主体 s 和客体 o 之间双向流动时(如 s 既读 o 又写 o),判定条件及主体标记的变化规则(规则 3)如下:

IF (f_c==f_o) THEN {$T(o,s)$=true,$T(s,o)$=true};

ELSE IF ($f_s \geq f_o$) && ($f_{ol} \geq f_o$) && ($f_o \geq f_{ih}$)

THEN {f_c=f_o,f_{ih}=Max(f_{ih},f_o),f_{ol}=Min(f_{ol},f_o),$T(o,s)$=true,$T(s,o)$=true};

ELSE {$T(o,s)$=false,$T(s,o)$=false}.

3.2 SLCF框架安全性的分析

SLCF 模型存在如下不足:

首先,模型变量的定义存在冗余.文献[19]指出,模型增加了 4 个安全级函数:f_{il},f_{ih},f_{ol},f_{oh},但是从模型规则来看,f_{il} 和 f_{oh} 没有在模型中使用.实际上,流入 s 的信息的最小敏感级和流出 s 的信息的最高敏感级对 s 进行读和写操作的判断并没有影响.主体 s 是否能读客体 o(引起信息流入 s)不是由 $f_{il}(s)$决定的,而是由 f_c 和 f_{ol} 决定的.同理,s 是否能写客体 o(引起信息流入 s)也不是由 f_{oh} 决定的,而是由 f_c 和 f_{ih} 决定的.

其次,与 DBLP 模型一样,SLCF 也是不安全的.与分析 DBLP 模型的方法类似,我们举出反例加以说明.

例 2:与例 1 类似,假设系统机密性敏感级由高到低分为 4 级:*top secret*,*secret*,*confidential* 和 *unclassified*.如图 2 所示,进程 *process*1 是一个特洛伊木马,其当前敏感级是 *top secret*,文件 *file*1 和 *file*2 是系统中两个客体,*file*1 的敏感级为 *top secret*,*file*2 的敏感级为 *confidential*.下面我们来看特洛伊木马 s 是如何读取 *file*1 中的信息,并写入 *file*2 中的.根据 SLCF,进程 *process*1 创建时其相应的标记函数是 f_c(*process*1)=*top secret*,f_{ih}(*process*1)=*unclassified*, f_{ol}(*process*1)=*top secret*,*file*1 和 *file*2 的标记为 f_o(*file*1)=*secret*,f_o(*file*2)=*confidential*.

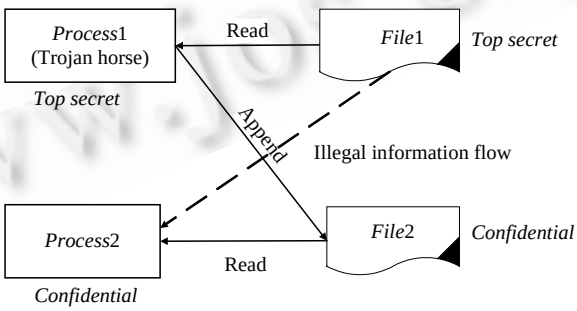

Fig.2 'illegal' information flow caused by Trojan horse under SLCF

图 2 SLCF 框架下特洛伊木马引起的信息泄漏

当 *process*1 提出对 *file*1 的读访问请求时,由于 f_c(*process*1)=f_o(*file*1),根据规则(1),读请求被许可,且 *process*1 的各安全标记不发生改变.

随后,*process*1 提出对文件 *file*2 的写访问请求.根据规则(2),f_o(*file*2)(=*confidential*)不大于 f_c(*process*1)(=*top secret*),因此,判断 f_o(*file*2)是否比 f_{ih}(*process*1)大,经比较发现,f_o(*file*2)≥f_{ih}(*process*1),所以写访问请求被许可,同时

调整 f_c(*process*1)和 f_{ol}(*process*1).这样,在访问控制规则许可下,进程 *process*1 就把刚才读取的敏感级为 *top secret* 的信息写入了敏感级为 *confidential* 的文件 *file*2 中,从而特洛伊木马 *process*1 就完成了信息从高敏感级向低敏感级的传递.敏感级为 *confidential* 的进程 *process*2 通过读 *file*2 就获取了 *file*1 中敏感级为 *top secret* 的信息.显然,在 MLS 策略下,这是不允许的.

3.3 SLCF安全失控原因分析

SLCF 框架不满足机密性策略的主要原因是,当主体对一个敏感级不大于其当前敏感级的客体提出读访问请求时,比较的是主体的当前敏感级 f_c,且访问被允许后,主体的 $f_{ih}(s)$没有做出相应的调整,从而没有记录读访问历史.当 s 提出对安全级不大于 s 当前安全级的客体进行写访问时,访问监控器却根据 $f_{ih}(s)$进行判断和裁决.显然,由于此时 f_{ih} 为系统最低敏感级 *LOW*,因此,s 对任何客体的写访问请求总是被许可.这样,s 就能够读取高安全级的信息,并把信息写入低安全级的客体中.正确的模型应该是,每次读操作都要考虑 $f_{ih}(s)$的变化,只要主体 s 当前读取的客体的安全级高于 $f_{ih}(s)$,$f_{ih}(s)$就要调整为客体的安全级.同样,在执行写操作时,要考虑 $f_{ol}(s)$的变化,只要当前写的客体的安全级低于 $f_{ol}(s)$,就要对 $f_{ol}(s)$进行相应的调整,将其调整为客体的敏感级.这样,改进的模型实际上与 Ott 的模型就是一致的,但却可以保证模型的正确性.

4 总 结

本文对两个改进的 BLP 模型(DBLP 和 SLCF)的安全性进行了详细分析.分析结果表明,这两个模型都是不安全的.在这两个模型的限制性规则下,特洛伊木马通过显式地读/写操作,可以“合法地”将高密级的信息泄漏给低密级的主体,因此违反了 MLS 策略.本文的分析工作表明:1) 在对 BLP 模型改进的过程中,应当把安全性放在首要考虑的位置,其次才能考虑灵活性;2) 安全性的证明对于各种改进的 BLP 模型是必不可少的.

由于 MLS 及其实现模型即 BLP 模型在安全操作系统实现机密性保护方面的重要作用,研究如何在保持安全性的前提下增加其灵活性的改进 BLP 模型,仍具有非常重要的意义.但是,任何改进的 BLP 模型都必须把安全性放在优先考虑的位置,其次才能考虑改进其不足.如果一个改进的 BLP 模型失去了机密性保护的作用,则是有悖于 BLP 模型的初衷的.

致谢 感谢贺也平研究员和张宏博士给本文提出了宝贵意见,感谢审稿人提出的修改意见.

References:

[1] Bell DE, La Padula LJ. Secure computer systems: Mathematical foundations. ESD-TR-73-278, I (AD) 770 768, Electronic Systems Division, Air Force System Command, Hanscom AFB. Bedford, 1973.

[2] Bell DE, La Padula LJ. Secure computer systems: A mathematical model. ESD-TR-73-278, II (AD) 771 543, Electronic Systems Division, Air Force System Command, Hanscom AFB. Bedford, 1973.

[3] Gasser M. Building a Secure Computer System. New York: Van Nostrand Reinhold Company, 1988.

[4] Bell DE. Secure computer system: A retrospective. In: Proc. of the 1983 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1983. 161−162.

[5] Bell DE, La Padula LJ. Secure computer system: Unified exposition and multics interpretation. Mitre Report, MTR-2997 Rev. 1. 1976.

[6] Waldhart NA. The army secure operating system. In: Proc. of the 1990 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1990. 50−60.

[7] Di Vito BL, Palmquist PH, Anderson ER, Johnston ML. Specification and verification of the ASOS kernel. In: Proc. of the 1990 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1990. 61−75.

[8] Terry VB. Analysis of a kernel verification. In: Proc. of the 1984 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1984. 125−133.

[9] Bell DE. Security policy modeling for the next-generation packet switch. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE Computer Society Press, 1988. 212−216.

[10] Schell RR, Tao TF, Heckman M. Designing the GEMSOS security kernel for security and performance. In: Proc. of the 8th National Computer Security Conf. 1985. 108−119.

[11] Mayer FL. An interpretation of refined Bell-La Padula model for the TMach kernel. In: Proc. of the 4th Aerospace Computer Security Application Conf. IEEE Computer Society Press, 1988. 368−378.

[12] Secure Computing Corporation. Assurance in the fluke microkernel: Formal top-level specification. Technical Report, CDRL A004, Secure Computing Corporation, 1999.

[13] Karger PA, Austel VR, Toll DC. A new mandatory security policy combining secrecy and integrity. IBM Research Report, RC 21717, 2000.

[14] Loscocco P, Smalley S. Integrating flexible support for security policies into the linux operating system. Technical Report, NAI Labs, 2001.

[15] McLean J. Reasoning about security models. In: Proc. of the 1987 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1987. 123−133.

[16] Ott A. Regel-Basierte Zugriffskontrolle nach dem generalized framework for access controlansatz am beispiel linux. Diplomarbeit Universitat Hamburg, 1997.

[17] 2006. http://www.rsbac.org/documentation/rsbac_handbook/security_models/mac

[18] Shi WC. Research on and enforcement of methods of secure operating systems development [Ph.D. Thesis]. Beijing: Institute of Software, the Chinese Academy of Sciences, 2001 (in Chinese with English abstract).

[19] Liang HL, Sun YF, Zhao QS, Zhang XF, Sun B. Design and implementation of a security label common framework. Journal of Software, 2003,14(3):547−552 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/547.htm

[20] Ji QG, Qing SH, He YP. An improved dynamically modified confidentiality policies model. Journal of Software, 2004,15(10): 1547−1557 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1547.htm

[21] Ji QG. Study on formalization design for high-level secure operating system [Ph.D. Thesis]. Beijing: Institute of Software, the Chinese Academy of Sciences, 2004 (in Chinese with English abstract).

[22] Bishop M. Computer Security: Art and Science. New York: Addison-Wesley, 2002.

附中文参考文献:

[18] 石文昌.安全操作系统开发方法的研究与实施[博士学位论文].北京:中国科学院软件研究所,2001.

[19] 梁洪亮,孙玉芳,赵庆松,张相锋,孙波.一个安全标记公共框架的设计与实现.软件学报,2003,14(3):547−552. http://www.jos.org.cn/1000-9825/14/547.htm

[20] 季庆光,卿斯汉,贺也平.一个改进的可动态调节的机密性策略模型.软件学报,2004,15(10):1547−1557. http://www.jos.org.cn/1000-9825/15/1547.htm

[21] 季庆光.高安全等级操作系统形式设计的研究[博士学位论文].北京:中国科学院软件研究所,2004.

何建波(1978－),男,湖南新田人,博士生,主要研究领域为信息系统安全技术.

王超(1978－),男,博士生,CCF 学生会员,主要研究领域为大型网络信息安全与恶意代码.

卿斯汉(1939－),男,研究员,博士生导师,CCF 高级会员,主要研究领域为信息系统安全理论和技术.