安全操作系统中基于安全性损害分析的可信恢复*)

李松林 袁春阳 朱继峰 淮晓永 梁洪亮 贺也平
(中国科学院软件研究所 北京 100080)

摘 要 安全操作系统可能因为内部或外部的原因发生失效或中断,进而导致其安全性损害。本文首先描述了一个通用的安全模型,然后扩展此模型以描述安全操作系统中的安全性损害,并提出依据安全策略从安全审计日志中分析计算安全性损害的方法,最后给出了安全性损害相应的可信恢复算法。在消极的安全防御外,本文的研究主动保证安全操作系统的安全性,增强了安全操作系统的可靠性和可恢复性。

关键词 安全性损害,可信恢复,机密性,完整性,安全操作系统

Trusted Recovery Based on Security Compromises Analysis in Secure Operating System

LI Song-Lin YUAN Chun-Yang ZHU Ji-Feng HUAI Xiao-Yong LIANG Hong-Liang HE Ye-Ping
(Institute of Software, Chinese Academy of Science, Beijing 100080)

Abstract Failures or discontinuities in secure operating systems may be inevitable because of internal or external reasons, and then security compromises. First, a common security model is given out in this paper. Then it is extended to in clude security compromises. And it is suggested to detect security compromises by analyzing security audit log. At last, trusted recovery algorithms is argued how to guarantee security in secure operating systems. Aparting from passive defenses, works in this paper guarantee security forwardly, and promote reliability and recoverability of secure operating systems.

Keywords Security compromise, Trusted recovery, Confidentiality, Integrity, Secure operating system

1 引言

从20世纪60年代开始,安全操作系统开始引起了各研究机构的重视[1,2]。目前,对于操作系统安全性的研究主要集中在防范攻击和威胁。但是,鉴于操作系统面临着内外部环境的各种安全攻击和威胁,以及操作系统自身设计和实现的复杂性,各种各样的错误(error)和失效(failure)都可能发生在操作系统中[3,4]。防范机制和保护措施并不能完全解决系统的安全问题,这主要体现在防范机制可能被绕过和防范机制可能本身也会出错。即使经过形式化严格证明后的安全操作系统,也会因为某些无法预料的原因(如自身错误,外部攻击、磁盘故障等)而导致系统关键信息被破坏,安全特性和安全功能的缺失,无法保证能够提供应有的安全服务[5]。所以,需要一种机制和手段,在消极的安全防御外,主动保护安全操作系统的安全性,从而增强安全操作系统的可靠性和可恢复性。可信恢复是国家标准GB17859-1999[6]中评估最高安全级别——《访问验证保护级》(相当于TESEC[7]的B3级)——安全操作系统中的要求。可信恢复是安全操作系统安全性的一种主动保护方式。目前,现有的符合TCSEC中B3级以上安全操作系统(如XTS-300[8]等)中可信恢复仍然实现方式简单,保护模式单一,也没有对安全性损害进行形式化描述。

本文在对安全操作系统安全策略模型研究的基础上,通过扩展安全性模型描述了安全性损害,并提出了根据安全策略分析安全审计日志检测安全性损害的方法,最后给出了安全性损害的可信恢复算法,并举例分析了可信恢复如何保证安全操作系统的安全性。

2 安全性模型

本节描述了一个抽象的安全操作系统中安全模型的通用定义(来自文[9]和[10]),并描述了它们是如何在安全操作系统中通过访问控制机制实现的,然后扩展该模型描述了系统中的安全性损害。

2.1 安全性模型概述

安全操作系统中安全模型主要由以下内容组成:主体(Subject)集 S,客体(Object)集 O,访问权限(right)集 R,访问矩阵 $A[s,o]\subseteq R$。请求(Request)集 $Q:=\{\langle s,o,r\rangle \mid s\in S, o\in O, r\in R\}$。$q:=\langle s,o,r\rangle, q\in Q$ 对应着一个由主体 s 使用访问权限 r(比如读操作、写操作等)作用在客体 o 的访问请求。一般而言,系统中的访问权限集合 $R:=\{read, write\}$。如果主体能够通过访问权限 r 改变客体,那么它就属于 *write* 类的,否则便属于 *read* 类的。系统状态 $v\in V$ 是系统中当前活跃的主体和客体,以及访问矩阵的集合,表示为 (S,O,A)。状态转变 $T: V\times Q\to V$ 表示从一个系统状态到另一个系统状态的转变。状态转变记为 $v\to v'$,也就是 $\langle(S,O,A),\{q\}\rangle\to\langle(S',O',A')\rangle, q\in Q$。在状态转变过程中,系统状态(S,O,

*)科技部攻关项目,编号2005BA113A02、中国科学院研究生创新资金"安全操作系统中可信恢复的研究"和发改委项目"安全系统软件产业化"资助。**李松林** 硕士研究生,主要研究方向为系统软件和信息安全。**袁春阳** 博士研究生,主要研究方向为系统软件和信息安全。**朱继峰** 博士,主要研究方向为系统软件与信息安全。**淮晓永** 副研究员,主要研究方向为智能计算与基础软件。**梁洪亮** 副研究员,主要研究方向为系统软件与信息安全。**贺也平** 研究员,主要研究方向为安全操作系统。

A)可能改变,也可能不改变。为了保持符号的紧凑,这里仅仅包含了系统当前状态集发生状态转变时发生了改变的资源集。

美国国防部 TCSEC[7]标准中定义安全性主要体现在机密性和完整性上。机密性用来隐藏信息,或者阻止对信息的未经授权的访问;完整性主要阻止对信息的未经授权的修改。它们被定义如下:

机密性 作用于客体子集 $O_C \subseteq O$ 上的机密性策略 P_C 将主体集分为两个子集 S_C 和 S'_C。在 S'_C 中的主体并不知道 O_C 中客体的存在或不能访问 O_C 中客体,也不可以通过访问矩阵 $A[s',o']$, $\forall s' \in S'_C$ 中的访问权限访问 O_C 中客体。P_C 则表述了主体 $s \in S_C$ 可以用访问权限 r 访问 O_C 中客体。

完整性 作用于客体子集 $O_I \subseteq O$ 上的机密性策略 P_I 将主体集分为两个子集 S_I 和 S'_I。在 S'_I 中的主体不被允许修改 O_I 中客体。P_I 则表述了主体 $s \in S_I$ 可以用修改 O_I 中客体。S_I 中任一主体对 O_I 中客体的改变都被 S_I 中其他主体所信赖。

系统的状态和行为符合安全策略的描述是系统安全的条件。定义系统中的安全策略集 $P:=\{\langle s,o,r\rangle \mid s \in S, o \in O, r \in R\}$,它描述了请求集 Q 中的为安全策略所授权的访问请求集 Q_A。子集 $Q'_A:=Q-Q_A$ 则描述了未经授权的访问请求。安全策略在安全系统中通过访问控制机制实施。访问控制机制中途截取访问请求并根据安全策略进行验证,拒绝违背策略的访问请求[9]。在下面,通过对安全操作系统中系统状态转变的描述说明安全策略是如何在安全操作系统中实施的。

一个安全的系统状态转变为符合安全策略 P 的条件状态转变。值得注意的是:安全的系统状态转变可能并没有改变系统状态。

定义 1(T1) 一个系统状态转变 $\langle(S,O,A),\{q\}\rangle \rightarrow \langle(S',O',A')\rangle$ 被认为是安全的,只要请求 $q \in P$。

空状态转变描述了根据安全策略 P,主体因为没有访问权限,访问请求被访问控制机制拒绝的情况。空状态转变没有改变系统状态。

定义 2(T2) 一个'空'系统状态转变 $\langle(S,O,A),\{q\}\rangle \rightarrow \langle(S,O,A)\rangle$ 中系统状态保持不变,因为 $q \notin P$。

有一种系统状态转变总是被允许的,那就是发生在不带有访问请求的系统状态和后续系统状态之间的状态转变。这个状态转变不考虑后续的系统状态带有何种请求,永远是允许的,所以也称为无条件状态转变。无条件状态转变的典型例子就是系统在初始状态时用户输入第一个请求时所触发的系统状态转变。

定义 3(T3) 无条件系统状态转变是 $\langle(S,O,A)\rangle \rightarrow \langle(S,O,A),\{q\}\rangle, q \in Q$。

对安全策略认可的状态转变的形式化描述提供了一个校验系统安全性的方法:比较观察到的系统状态转变和安全策略所允许的状态转变,如果观察到的系统状态转变始终与 T1、T2 和 T3 的描述相符,那么这个系统就是安全的。

2.2 安全性损害

在前面,我们给出了一个系统中安全性模型的定义,但是它并不能表述当系统面临攻击、发生失效或中断,产生安全性损害时的系统状态转变。为此,我们扩展了上面的安全性模型,以说明系统中发生安全性损害时的状态转变,并根据系统状态转变给出了机密性损害和完整性损害的定义。

在 2.1 节中定义的安全操作系统中的安全的状态转变 T1、T2 和 T3 基于安全策略总是被正确实施的假设基础上,即所有的状态转变行为总是符合安全策略的。这样的状态转变可以用图 1 表示。

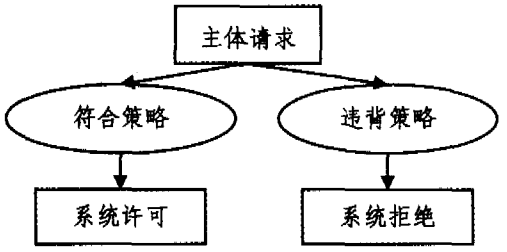

图 1 理论上系统中的状态转变

而在实际的安全操作系统中,黑客攻击、管理员的不当行为、软/硬件故障等都可能造成系统状态转变与安全策略不一致,导致安全性损害。我们在图 2 中描述了实际系统中系统状态转变的情况。其中,虚线描述了安全性损害时与安全策略不一致的系统状态转变。第一种是符合安全策略的请求却被系统拒绝,导致拒绝服务,表现为系统拒绝合法主体访问合法客体。然而,拒绝服务并不会导致信息泄漏或信息损害,所以又被称为良性安全性损害。第二种是违背安全策略的请求却被系统所执行,导致信息泄露或信息被非法的主体修改。由此产生的损害是不可挽回的,故又称为恶性安全性损害。

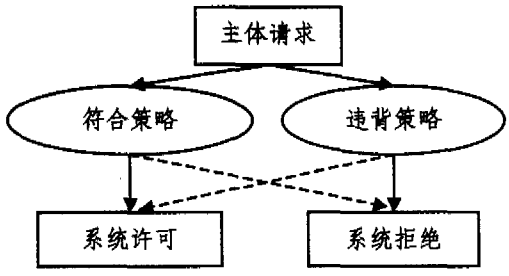

图 2 实际系统中的状态转变

定义 4(T4) 状态转变 $\langle(S,O,A),\{q\}\rangle \rightarrow \langle(S,O,A)\rangle$ 中 $q \in P$ 被拒绝,这个状态转变是不安全的却是良性的。主体发出的请求符合安全策略,系统却拒绝执行,导致拒绝服务,系统状态没有变化。状态转变过程中没有导致信息泄露或信息被非法主体修改。

定义 5(T5) 状态转变 $\langle(S,O,A),\{q\}\rangle \rightarrow \langle(S',O',A')\rangle, q \in Q$,是不安全且是恶性的,如果 $q \notin P$。此状态转变过程导致信息泄露或信息被非法主体修改,并且产生的损害不可挽回。

不安全的状态转变造成安全性损害。根据 T4 和 T5 的描述,系统中的机密性损害可以分为以下两种类型:

良性机密性损害 主体 $s \in S_C$ 根据策略 P_C 被允许访问客体 $o \in O_C$,但是被系统所拒绝。原本可以自由交互的信息现在变得不可用。然而,它并没有扩散机密信息。

恶性机密性损害 主体 $s \in S'_C$ 根据策略 P_C 不被允许访问客体 $o \in O_C$,现在突然可以访问 o 并可以进一步扩散这个机密信息,并允许其他 S'_C 中的未授权主体获取这个信息。

相应地,系统中的完整性损害也可以分为两种类型:

良性完整性损害 主体 $s \in S_I$ 根据策略 P_I 被允许写或者修改客体 $o \in O_I$,但是被系统所拒绝。信息可能变得陈旧,同时对 S_I 中的其他主体不再有用。

恶性完整性损害 主体 $s \in S'_I$ 根据策略 P_I 不被允许修

改客体$o\in O_I$，现在突然可以修改它。S_I 中的其他主体现在不得不信任 S'_I 中的主体。已经损害的信息可以在系统中进一步扩散。

3 利用审计日志检测安全性损害

安全审计是安全操作系统的一个基本安全功能[6]。安全审计通过记录和检查系统安全相关事件，来检测由于外部渗透和内部误用所引起的安全性损害行为，实现安全监视功能。通过对当前系统状态安全性的判定，安全审计给用户提供系统安全策略实施有效性的判定。我们这里通过分析审计日志来检测系统中的安全性损害。日志安全性监视的具体做法是首先参照安全策略的规则表达式方式生成各种安全监视目标和日志项目集，然后根据日志项目集来选取审计日志数据，最后依据安全监视目标和安全策略对日志数据进行合法性分析，即判别系统状态转变行为是否符合安全策略。

(1)确定安全监视目标

$CreateAuditTarget: P\Rightarrow AUDITTARGETs$

根据安全策略 P 来生成安全监视目标。这里的安全策略 P 主要指安全操作系统中实施的机密性策略(如 Bell-LaPadula[10])和完整性策略(如 Biba[11])，即 $P\subseteq(P_C\cup P_I)$。相应的安全监视目标到安全策略上的映射为：

$MAP_{AUDITTARGET\text{-}P}: AUDITTARGET\Rightarrow P$

这里的 P 是指安全策略集，所有的安全监视目标都由其中的元素决定，每个安全监视目标都反映了系统对部分(或全部)状态转变的满足情况。

(2)选择日志项目集

$SelectLogItems: AUDITTARGETs\times p\Rightarrow LOGITEM$

该功能主要用以从审计日志中选择特定日志内容。其方法依据安全监视目标集以及相关安全策略项 $p\in P$ 来选择日志项。审计日志数据主要依照系统状态以及状态转变，加上日志项集形成。

$SysLOG: V\times LOGITEMs\Rightarrow LOGDATAs$

在日志项和系统状态之间存在一种的映射关系：

$MAP_{LOGITEM\text{-}STATE}: LOGITEMs\Rightarrow V$

(3)安全监视

$AuditMointor: AUDITTARGETs\times LOGDATAs\times p\Rightarrow RESULTs$

这里 $p\in P$，$RESULT:=\{\text{true},\text{false}\}$，true 和 false 是布尔值，true 表示安全，false 则为不安全。安全监视功能根据安全监视目标并参照相关安全策略和与策略相关的系统状态所对应的日志项，从该系统状态相关部分所规定的项目日志数据，再根据上述信息判定系统是否发生了安全性损害。

4 恶性安全性损害的可信恢复

在第 2、3 节中讨论了安全性损害的发生以及如何借助审计日志来检测安全性损害。良性安全性损害主要表现为拒绝服务，但是它并不会破坏安全操作系统的安全特性和安全功能。恶性安全性损害造成了信息泄漏或信息被破坏，破坏了安全操作系统的安全特性和安全功能。本段主要讨论可信恢复如何在发生恶性安全性损害后保证安全操作系统的安全性。

4.1 恶性机密性损害的可信恢复

恶性机密性损害$\langle(S,O,A),\{q_D\}\rangle\rightarrow\langle(S',O',A')\rangle$，$q_D\in Q$并且$q_D\notin P_C$造成了信息泄露，泄漏的信息不可挽回。故恶性机密性损害的可信恢复策略是阻止主体扩散信息，降低机密性损害造成的损失。针对恶性机密性损害行为 $q_D:=\langle s_D,o_D,r\rangle$，且 $q_D\notin P_C$ 的可信恢复算法如下：

(1)为了防止主体 s_D 扩散 o_D 的信息，如果在 q_D 随后的请求为$\langle s_D,o,write\rangle$，$o\in O_C$，则该请求也被拒绝。同时 s_D 后续的请求 $q:=\langle s_D,o_D,r\rangle$，$q\in Q$，都将被拒绝。

(2)为了防止主体 s_D 扩散 o_D 访问权限给其他主体，针对系统中主体的机密性策略的安全测试是必须的，即对访问请求$\langle s,o_D,r\rangle\in Q$，如果主体 $\forall s\in S'$，那么 s 作用于 o_D 上请求都将被否决，且$\langle s,o_D,r\rangle\notin P_C$。

下面举例分析说明可信恢复算法对系统机密性的保证。假定系统状态中当前主体集 $S:=\{John,Tom\}$，客体集 $O:=\{passwd,file\}$，访问权限 $R:=\{read,write\}$，安全策略 $P_C:=\{\langle John,file,write\rangle,\langle Tom,file,read\rangle\}$。已经执行的请求 $H:=\{\langle John,passwd,read\rangle,\langle(John,Tom),passwd,write\rangle\}$，后续的请求 $Q:=\{\langle John,file,write\rangle,\langle John,passwd,read\rangle\}$。审计日志中检查到请求$\langle John,passwd,read\rangle$被执行，但是$\langle John,passwd,read\rangle\notin P_C$，故此执行导致中 $passwd$ 中信息被泄露给 $John$，导致机密性损害。请求$\langle(John,Tom),passwd,write\rangle$表示 $John$ 扩散 $passwd$ 的访问权限 $write$ 给 Tom。按照可信恢复过程(1)，后续的请求$\langle John,passwd,read\rangle\notin P_C$被拒绝，阻止信息泄露的再次发生；$\langle John,file,write\rangle\in P_C$被系统拒绝，$John$ 没有能够扩散 $passwd$ 中的信息。经过可信恢复过程(2)，系统检测到$\langle Tom,passwd,write\rangle$，$Tom\in S'$，则阻止 Tom 对 $passwd$ 的 $write$ 请求，且$\langle Tom,passwd,write\rangle\notin P_C$。可见，可信恢复算法阻止了信息和访问权限的扩散，保证了系统的机密性。

4.2 恶性完整性损害的可信恢复

恶性完整性损害$\langle(S,O,A),\{q\}\rangle\rightarrow\langle(S',O',A')\rangle$，$q:=\langle s_D,o_D,r\rangle$，$q\in Q$并且$q\notin P_I$发生时，系统所采纳的恢复策略是使用客体的备份替换被损害的客体，即 $o'_D\Rightarrow o_D$。不过这可能造成另一个问题，即当一个客体的状态被恢复到先前状态时，因为信息的陈旧，造成系统内部客体的不一致，进而可能造成系统的不一致，产生安全性损害。例如系统状态中客体集 $O:=\{libz.so,libz.so.1.2.2\}$，且 $libz.so=libz.so.1.2.2$，即 $libz.so$ 是 $libz.so.1.2.2$ 的软链接。当系统发生了恶性完整性损害，$libz.so.1.2.2$ 被破坏。使用备份 $libz.so.1$ 替换被破坏 $libz.so.1.2.2$，此时访问 $libz.so$ 会出现找不到文件的情况。在安全操作系统中，首先观察客体 $b\in O$ 的值，然后调用访问权限操作 r，再观察 b 的值，如果能够经过以上的过程推断出全部(或部分)客体 $a\in O$ 的值，则称客体 a 依赖于 b[12]，记作 $a\infty b$。

例子表明，在恶性完整性损害中，仅执行恢复 $o'_D\rightarrow o_D$ 是不够的，在恢复 o_D 的同时还需要将那些依赖于 o_D 的客体状态也恢复到前一个状态。恶性完整性损害$\langle s_D,o_D,r\rangle$的可信恢复过程的算法为：

(1)$o'_D\Rightarrow o_D$，即将被损害的 o'_D 恢复到先前的 o_D；

(2)定义回滚集 $ROLLBACK:=\{\}$

对客体 $o\in O$，

a)如果 $o\infty o_D$，$ROLLBACK:=ROLLBACK\cup\{o\}$；

b)如果 $o\infty o_r$，$o_r\in ROLLBACK$，$ROLLBACK:=ROLLBACK\cup\{o\}$。

(3)对所有 $o\in ROLLBACK$，也将 o 恢复到先前保存的状态。

(下转第 273 页)

人的思考方式、自主行为等)。基于 Agent 的行为建模仍存在一些困难[18]:如人脑思维模型的复杂性导致无法保证系统建模中抽象、表达、推理及学习等过程的正确性等。

小结 随着虚拟环境系统越来越复杂、要求越来越高,对环境中物体进行合适的行为建模也越来越成为人们关注的焦点。行为建模经历了基于过程的行为建模、基于对象的行为建模到基于 Agent 的行为建模,由于 Agent 具有描述人的信念、愿望、意图等精神状态的能力,所以基于 Agent 的行为建模会成为未来智能化、真实化虚拟环境的主要建模方法。

目前,基于 Agent 的行为建模技术虽然还处于研究与探讨阶段,但仍已被认为具有较为广阔的研究前景。主要在以下两个方面需要进一步的研究:一、系统应用方面,即如何综合运用现有的基于 Agent 的行为建模技术构建更加智能、快捷的虚拟环境;二、模型构建方面,即如何构建更加智能、更能反应人各种真实行为的 Agent 模型。

参 考 文 献

1 Stuart R. The Design of Virtual Environments. McGraw-Hili, New York, NY, 1996

2 Greenhalgh C. Large Scale Collaborative Virtual Environments: [PhD Thesis]. The University of Nottinghan, 1997

3 潘志庚,姜晓红,张明敏,石教英. 分布式虚拟环境综述. 软件学报, 2000, 11(4): 461~467

4 Benford S,et al. Collaborative Virtual Environments. Communications of The ACM, 2001, 44(7): 79~85

5 郑援,李思昆. 虚拟实体对象的行为建模方法研究. 国防科技大学学报,1998, 20(1): 78~82

6 Cermer J,Kearney J,Ko H. Simulation and scenario support for virtual environments. Computer & Graphics, 1996, 20(2): 199~206

7 Reynolds, Flocks C W, Herds, Schools. A Distributed Behavioral Model. Computer Graphics, 1987, 21(4): 25~34

8 Tu Xiaoyuan, Terzopoulos D. Artificial fishes: physics, locomotion, perception, behavior. In:Proceedings of SIGGRAPH 94, 1994

9 Hagsand O. Interactive Multiuser VEs in the DIVE System, IEEE Multimedia Magazine, 1996, 3(1): 30~39

10 Kallmann M, Thalmann D. Modeling Objects for Interaction Tasks. In:Proc. Eurographics Workshop on Animation and Simulation, 1998. 73~86

11 Kallmann M,Thalmann D. A Behavioral Interface to Simulate Agent-Object Interactions in Real Time. In: Proc. IEEE International Conference on Computer Animation, 1999. 138~146

12 Jorissen P,Lamotte W. A Framwork Supporting General Object Interactions for Dynamic Virtual Words. In: Proc. International Symposium on. Smart Graphics, 2004. 154~158

13 Wang Weihua,Lin Qingping. Behavior Based Interaction Management Mechanism for an Internet CVE System. In:Proc. IEEE International Conference on Multimedia and Expo, 2001. 820~823

14 Lin Qingping,Wang Weihua, et al. A Novel Method for Supporting Collaborative Interaction Management in Web-based CVE. In: Proc. Conference on Virtual Reality Software and Technology, 2003

15 Mulgund SS, Harper KA, Zacharias GL. SAMPLE: Situation Awareness Model For Pilot-In-The-Loop Evaluation. In: Proc. 9th Computer Generated Forces and Behavior Representation, 2000

16 梁晓辉,胡晓雁. 一种基于 Agent 的构造性虚拟实体的体系结构. 系统仿真学报, 2001, 13(11): 138~141

17 党岗,刘华峰,等. 多 Agent 虚拟环境中行为建模及其动画表现. 系统仿真学报, 2001, 13(11): 142~145

18 张航义. 基于 Agent 的 CGF 行为建模技术研究. 计算机仿真, 2003, 20(8): 79~81

(上接第 266 页)

在上面的例子中,*libz. so*∞*libz. so*. 1. 2. 2,所以在用备份 *libz. so*. 1 替换 *libz. so*. 1. 2. 2 后,也应该同时用备份替换 *libz. so*。可信恢复算法不仅保证了系统内部单个客体的完整性,而且还保证了系统内部客体的一致性。

结束语 可信恢复主动保护安全操作系统的安全性。文章在对安全操作系统安全模型研究的基础上,首先描述了一个通用的安全模型,然后扩展该模型描述了安全操作系统中安全性损害。随后,描述了如何根据安全策略借助审计日志的安全监视功能来检测安全操作系统中发生的种种安全性损害。最后,讨论了可信恢复如何修复恶性机密性损害和恶性完整性损害,保证安全操作系统的安全性。可信恢复主动保证安全操作系统的安全性,增强了安全操作系统的可靠性和可恢复性。本文的工作为今后开发《访问验证保护级》安全操作系统和对可信恢复的进一步研究提供了理论模型和方法指导。

参 考 文 献

1 石文昌,孙玉芳. 安全操作系统研究的发展(上)[J]. 计算机科学,2002,29(6)

2 石文昌,孙玉芳. 安全操作系统研究的发展(下)[J]. 计算机科学,2002,29(7)

3 Sullivan M, Chillarege R. Software Defects and their Impact on System Availability - A Study of Field Failures in Operating System [J]. 21th Int. Symp. On Fault-Tolerant Computing, June 1991. 2~9

4 Randell B. Operating systems: The problems of performance and reliability [J]. In: Proc. IFIP Congress, North-Holland, 1971. 281~290

5 National Computer Security Center. A Guide to Understanding Trusted Recovery in Trusted Systems [R]. NCSC-TG-022, Library No. 5-236,061, Version 1, December 1991

6 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[S], GB17859-1999

7 Dept. of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria [S]. DOD 5200. 28-STD, GPO 1986-623-963, 643 0, Dec. 1985

8 National Computer Security Center. Final Evaluation report Wang Government Services, Incorporated XTS-300 STOP 5. 2. E [R]: [Report NO. CSC-EPL-92/003. E]. August 2000

9 Gasser M. Building a Secure Computer System [M]. Van Nostrand Reinhold. ISBN 0-442-23022-2

10 Bell DE, LaPadula LJ. Secure Computer Systems: Mathematical Foundations [R]: [Technical Report M74-224]. The MITRE Corp., Bedford, Mass., May 1973

11 Biba K J. Integrity considerations for secure computer systems [R]. USAF Electronic System Division, Hanscom Air Force Base: [Tech. Rep., ESD-TR-76-372]. 1977

12 Naval Research Laboratory. Handbook for the Computer Security Certification of Trusted Systems. NRL Technical Memorandum 5540:062A, 12 Feb. 1996. 14