安全操作系统非传统开发模式研究*)

石文昌　孙玉芳
(中国科学院软件研究所　北京100080)

Research on Nontraditional Development Model of Secure Operating Systems

SHI Wen-Chang　SUN Yu-Fang
(Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract The TCSEC-based traditional development model of secure operating systems is exhibiting more and more obviously inability in the rapidly changing world of computer applications. With an experiment of building a secure operating system in accordance with the philosophy of the international standard for computer security evaluation, i. e. the Common Criteria, research on nontraditional development model of secure operating systems is conducted in this paper.

Keywords Secure operating system, Development model, TCSEC, Common criteria

1 引言

美国国防部于1983年颁布并于1985年修定的 TCSEC 标准[1]是安全操作系统发展史上的第一个计算机安全评价标准，它对安全操作系统的开发具有根深蒂固的影响，基于 TCSEC 标准的开发模式成了传统开发模式的代名词。基于 TCSEC 标准的开发模式的根本特点是根据以 TCSEC 标准为中心的彩虹系列的报告来确定安全操作系统的安全需求。TCSEC 标准确定的安全需求是静态的，而计算机技术的发展速度是惊人的。资深的计算机安全专家、安全操作系统的开发先驱[2]与彩虹系列很多报告的编撰者[3]V. G. Gligor 指出，基于 TCSEC 标准的传统开发模式存在着严重的问题[4]。事实上，在当前具有大范围分布式异构特点的应用环境中，按照传统开发模式开发出来的安全操作系统难以有效地用于解决实际应用所面临的形态多样的安全威胁问题。为了建立健壮的信息安全基础软件平台，必须在传统开发模式的框架之外，为安全操作系统的开发创立新的开发模式。本文以安全操作系统 RS-Linux 的开发实践为基础，围绕信息安全国际标准(CC 标准)[5]的思想，探讨安全操作系统的非传统开发模式问题，旨在从开发模式方面为满足实际应用需要的安全操作系统的开发建立基础。

2 研究方法的设计

本文工作的指导思想是试图摆脱基于 TCSEC 标准的传统开发模式的思维定式的影响，研究开发满足实际应用需要的安全操作系统的思想方法问题。一个安全系统是否能够满足实际应用的需要，可以从两个方面进行衡量，一是检验该安全系统提供的安全功能是否能够处理应用环境中将面临的安全威胁，二是检验系统提供的安全功能的实现是否值得信赖。国际 CC 标准的思想充分考虑了安全产品的这两个重要侧面，并提供了比较灵活的处理方法。

为了围绕 CC 标准的思想探讨安全操作系统的非传统开发模式，本文设计了一个按照 CC 标准的思想框架开发安全操作系统的研究实验，该研究实验的目标是开发一个以主流操作系统 Linux 为基础的安全操作系统 RS-Linux。

CC 标准框架的一个显著特点是把一个计算机安全产品应该具有的安全特性与为确保这些安全特性的正确实现而采取的安全措施作为两个独立的内容进行分别对待。在研究实验中，本文主要以 LSPP[6]中确定的安全特性为基础锁定 RS-Linux 的安全特性，考虑到历史上评价过的计算机安全产品的安全确信度大部分都介于 EAL3与 EAL4之间这个实情[7]，RS-Linux 的安全确信度被定位在增强的 EAL3安全保障等级。

CC 标准框架下的 RS-Linux 的开发过程及涉及到的与安全性密切相关的行为可描述为图1的形式。

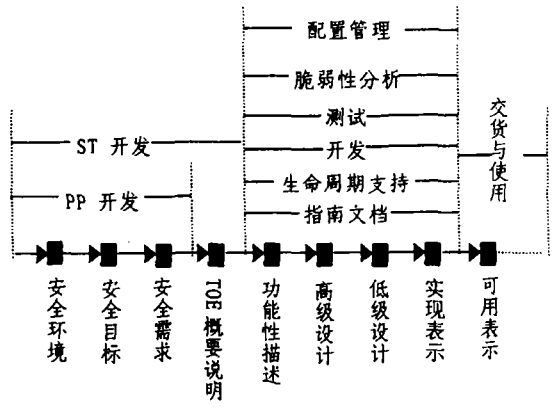

图1　CC 标准框架下的 RS-Linux 开发过程

CC 标准用“安全评价对象”(简记为 TOE)来定义一个打算被评价的计算机安全产品及其相应的管理员和用户指南文档。这里，RS-Linux 及其有关的文档就是一个 TOE，因为我们假定以后要按照 CC 标准对 RS-Linux 进行安全性评价。“保护轮廓定义书”(简记为 PP)和“安全对象定义书”(简记为 ST)是 CC 标准中定义的两种安全需求定义结构。一个 PP 针对一些确定的用户需要，为某一类 TOE 定义安全需求，这些

*)国家自然科学基金项目(60073022)、国家863高科技项目(863-306-ZD12-14-2)和中国科学院知识创新工程项目(KGCX1-09)资助。石文昌　博士，研究员，主要研究方向为系统软件与计算机安全。孙玉芳　研究员，博士生导师，主要研究方向为系统软件和中文信息处理。

安全需求与实现无关。一个 ST 为一个特定的 TOE 定义安全需求和概要说明，它是相应的 TOE 的评价基础。

在图1中，我们用一个由9个步骤构成的过程来描述 RS-Linux 的开发活动，这个过程可划分成3个阶段，即，ST 定义阶段，TOE 开发阶段和 TOE 交货和使用阶段，其中，ST 定义阶段中还包含一个 PP 定义子阶段。这个过程中的每一个步骤都产生一个相应的结果，第一个结果是“安全环境”，最后一个结果是系统的“可用表示”。

在不同的开发动机的驱动下，PP 和 ST 的开发对安全问题的解决可能会产生不同的影响[8]。不过，本文的讨论不区分这方面的差别，而且，我们假定开发过程中的各项活动都是按照实现良好愿望的方式开展的，即，都希望能推动计算机安全技术水平向前发展。

3 系统安全功能的确立

系统开发的第一个阶段(ST 设计阶段)的主要目的是确立 RS-Linux 的安全功能[5,9]，这些安全功能锁定 RS-Linux 将要提供的所有安全特性。

3.1 安全环境的确立

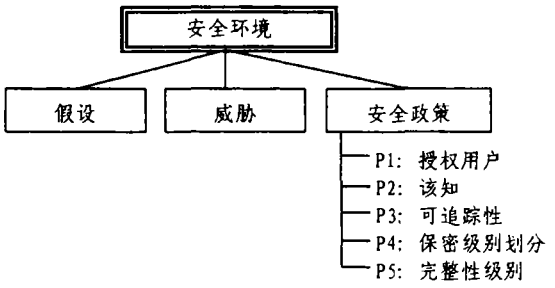

图2 RS-Linux 的安全环境

为达到这个阶段的目的，第一个步骤是建立系统的“安全环境”，它阐明 RS-Linux 将用在什么样的环境中。安全环境定义 RS-Linux 拟处理的安全问题的性质和范围，这包括安全假设、安全威胁和用户机构的安全政策等内容。就 RS-Linux 而言，安全问题中要考虑的主要是安全政策，没有特别的安全威胁需要描述。图2描绘 RS-Linux 的安全环境。

3.2 安全目标的产生

安全问题提出来之后，我们需要考虑 RS-Linux 将如何对所面临的安全问题进行处理。下一个步骤是确定 RS-Linux 应该达到的“安全目标”。安全目标应该是对安全问题的响应或解决方案的简明陈述。

确立安全目标时，我们必须证明，对于每一个安全政策和安全假设，至少必须有一个安全目标来对它们进行处理，对于每一个安全目标，至少必须处理一个安全政策或安全假设。图3给出 RS-Linux 的安全目标及安全政策与安全目标之间的映射。

安全目标

01: 授权
02: 自主访问控制
03: 强制访问控制
04: 数据完整性
05: 审计
06: 剩余信息
07: 管理
08: 实施

映射

P1	O1, O7, O8
P2	O2, O6, O7, O8
P3	O5, O7, O8
P4	O3, O6, O7, O8
P5	O4, O6, O7, O8

图3 安全目标及其到安全政策的映射

3.3 安全需求的定义

第三个步骤确立能够达到已建立的安全目标的“安全需求”，特别是安全功能需求(SFR；Security Functional Requirement)。在可能的情况下，应尽量应用 CC 标准中预定义的 SFR 组件来构筑安全产品的 SFR。

表1 用 CC 标准的组件定义的 RS-Linux 安全功能需求

安全审计	身份标识与鉴别
R01: 审计数据生成 (FAU_GEN.1) R02: 用户身份关联 (FAU_GEN.2) R03: 审计查看 (FAU_SAR.1) R04: 限制的审计查看 (FAU_SAR.2) R05: 可选择的审计查看 (FAU_SAR.3) R06: 选择性审计 (FAU_SEL.1) R07: 审计数据可用性保护 (FAU_STG.1) R08: 审计数据可能丢失的提防措施 (FAU_STG.3) R09: 防止审计数据丢失 (FAU_STG.4)	R22: 用户属性定义 (FIA_ATD.1) R23: 鉴别数据的强度 (FIA_SOS.1) R24: 鉴别 (FIA_UAU.1) R25: 保护的鉴别反馈 (FIA_UAU.7) R26: 身份标识 (FIA_UID.1) R27: 用户-主体绑定 (FIA_USB.1)
用户数据保护	安全管理
R10: 自主访问控制政策 (FDP_ACC.1) R11: 自主访问控制功能 (FDP_ACF.1) R12: 非标记用户数据导出 (FDP_ETC.1) R13: 标记用户数据导出 (FDP_ETC.2) R14: 强制访问控制政策 (FDP_IFC.1) R15: 强制访问控制功能 (FDP_IFF.2) R16: 数据完整性政策 (FDP_IFC.1) R17: 数据完整性功能 (FDP_IFF.2) R18: 非标记用户数据导入 (FDP_ITC.1) R19: 标记用户数据导入 (FDP_ITC.2) R20: 客体遗留信息保护 (FDP_RIP.2) R21: 主体遗留信息保护 (FDP_RIP.2+)	R28: 客体安全属性管理 (FMT_MSA.1) R29: 静态属性初始化 (FMT_MSA.3) R30: 审计踪迹管理 (FMT_MTD.1) R31: 审计事件管理 (FMT_MTD.1) R32: 用户属性管理 (FMT_MTD.1) R33: 鉴别数据管理 (FMT_MTD.1) R34: 用户属性撤销 (FMT_REV.1) R35: 客体属性撤销 (FMT_REV.1) R36: 安全管理角色 (FMT_SMR.1)
	TOE 安全功能保护
	R37: 抽象机器测试 (FPT_AMT.1) R38: 访问仲裁 (FPT_RVM.1) R39: 域隔离 (FPT_SEP.1) R40: 可靠时间印记 (FPT_STM.1)

这里要确立的SFR由两个方面的SFR组成，第一个方面是能够直接达到安全目标的那些SFR，另一个方面是被已选定的SFR依赖的那些SFR。例如，当SFR1依赖SFR2时，选定SFR1必须同时要选定SFR2，使得所确立的SFR既能实现安全目标又能满足SFR间的依赖关系。

我们必须证明，对于RS-Linux的每一个安全目标，至少必须有一个SFR用于实现它，对于每一个SFR，至少必须用于实现RS-Linux的一个安全目标。我们还必须证明，所有SFR的依赖关系是得到满足的，所有SFR之间是相互支持的，用于身份鉴别的口令机制的安全功能强度是符合安全目标要求的。

表1定义RS-Linux的安全功能需求，表2给出安全目标与安全需求之间的映射。

表2 RS-Linux安全目标与安全功能需求的映射

O1	R22,R23,R24,R25,R26,R33
O2	R10,R11,R22,R27,R28,R29,R35
O3	R12,R13,R14,R15,R18,R19,R27,R28,R29
O4	R12,R13,R16,R17,R18,R19,R27,R28,R29
O5	R01,R02,R03,R04,R05,R06,R07,R08,R09,R27,R30,R31,R40
O6	R20,R21
O7	R03,R05,R06,R08,R09,R30,R31,R32,R33,R34,R36
O8	R37,R38,R39

3.4 安全功能的描述

这个阶段的最后一个步骤要形成TOE“概要说明”，概要说明定义出满足所有SFR要求的RS-Linux的所有安全功能。安全功能可以刻画为RS-Linux要满足的SFR的精确化表述，与SFR相比，安全功能对RS-Linux的安全特性给予更详细的描述，使得RS-Linux的安全特性更容易被理解。

我们必须证明，对于每一个SFR，至少必须映射到一个安全功能，对于每一个安全功能，至少必须映射到一个SFR，并且，安全功能之间必须是相互支持的。

表3 RS-Linux安全功能简要描述

F01:身份标识与鉴别	F08:安全管理
F02:基于权限位的访问控制	F09:遗留信息处理
F03:基于访问控制表的访问控制	F10:硬件平台测试
F04:基于保密性标记的访问控制	F11:访问监控
F05:基于完整性标记的访问控制	F12:安全域隔离
F06:安全审计	F13:机器时钟操纵
F07:特权用户角色	

表4 RS-Linux安全功能与安全需求的映射

F01	R22,R23,R24,R25,R26,R33
F02	R10,R11,R22,R27,R28,R29,R35
F03	R10,R11,R22,R27,R28,R29,R35
F04	R12,R13,R14,R15,R18,R19,R27,R28,R29
F05	R12,R13,R16,R17,R18,R19,R27,R28,R29
F06	R01,R02,R03,R04,R05,R06,R07,R08,R09,R27,R30,R31
F07	R36
F08	R32,R34,R36
F09	R20,R21
F10	R37
F11	R38
F12	R39
F13	R40

表3给出RS-Linux的安全功能的简要描述，表4给出安全功能与安全需求之间的映射。

4 安全保障措施的实施

在第一个阶段里，我们也需要制定确保RS-Linux的安全性的安全保障措施，但这些保障措施是在第二个阶段(开发阶段)和第三个阶段(交货和使用阶段)实施的。

在RS-Linux的开发活动中，我们需要考虑7类安全保障措施，即，开发类(ADV)、生命周期支持类(ALC)、指南文档类(AGD)、测试类(ATE)、配置管理类(ACM)、脆弱性评估类(AVA)以及交货和使用类(ADO)，如图1所示。

需要注意的是，CC标准只预定义了可供使用的安全保障需求组件和可供选用的安全保障需求，但它没有也不可能规定具体的安全保障措施。制定安全保障措施是安全产品开发人员的职责。同一个安全保障等级的不同安全产品的开发可以采取不同的安全保障措施。

4.1 安全产品的表示

在第二和第三个阶段，CC标准区分一个安全产品的5种表现形式，它们分别是功能性描述、高级设计、低级设计、实现表示和可用表示。

在CC标准中，一个TOE中实施安全性的部分简记为TSF(TOE Security Functions)。功能性描述是在较高层次上对TSF中用户看得见的接口和行为的描述，图4是RS-Linux功能性描述的一个图示。高级设计把TSF划分为子系统进行描述。低级设计从模块的角度对TSF的内部工作机理进行描述。RS-Linux的实现表示就是系统的源码。RS-Linux的可用表示就是在计算机上运行的RS-Linux操作系统，这样的系统可以运行在开发人员的工作场所，也可以运行在用户的工作场所。

4.2 安全产品的开发

ADV类需要开展的工作涉及非形式化的功能性描述(ADV_FSP.1)、安全性实施的高级设计(ADV_HLD.2)、非形式化的一致性证明(ADV_RCR.1)以及非形式化的安全政策模型化(ADV_SPM.1)等方面的安全保证措施。

ADV_FSP.1措施为产品提供非形式化的功能性描述。ADV_HLD.2措施提供产品的高级设计，把产品的安全性实施子系统与产品的其它部分区别开来。安全保障等级EAL3不要求提供产品的低级设计。

对于要求提供的TSF的所有各种表示形式，ADV_RCR.1措施分析所有相邻的两种表示之间的一致性。具体地说就是，安全功能与功能性描述之间、功能性描述与高级设计之间、高级设计与源码之间，等的一致性。这些分析必须证明，相对抽象的一种表示中蕴涵的所有相关安全特性，必须在相对具体的对应表示中全部正确地得到细化，比如，功能性描述中蕴涵的所有安全特性，必须在高级设计中全部正确地得到细化。

ADV_SPM.1措施为所有可用模型表示的安全政策提供非形式化的安全政策模型，并证明系统的功能性描述与安全模型间的一致性。RS-Linux的设计提供基于保密性标记和基于完整性标记的强制访问控制支持，基于保密性标记的强制访问控制根据BLP模型[10]设计，基于完整性标记的强制访问控制根据Biba低水标模型[11]设计。

4.3 开发控制

ALC类处理开发环境的安全性，需要开展的工作涉及安

全措施的描述(ALC_DVS.1)等方面的安全保证措施。ALC_DVS.1措施描述并实施在开发环境中应采取的安全措施，开发环境中的安全措施保护RS-Linux的设计与实现的完整性，其目的是消除或降低开发环境中可能存在的对安全性的威胁。

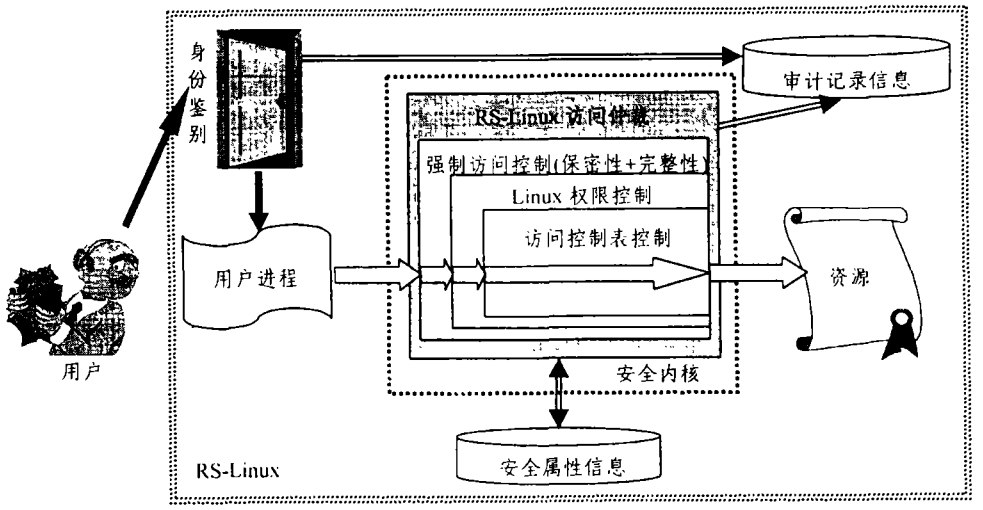

图4 安全操作系统RS-Linux的用户视图

ACM类涵盖安全产品的版本控制，需要开展的工作涉及授权控制(ACM_CAP.3)和TOE配置管理覆盖范围(ACM_SCP.1)等方面的安全保障措施。

ACM_CAP.3措施确保RS-Linux的所有配置组件都得到妥善的管理，对配置组件的任何修改都是在经过授权的情况下进行的。ACM_SCP.1措施确保最低限度也要对以下配置组件进行追踪管理：RS-Linux源码、设计文档、测试文档、用户文档、管理员文档以及配置管理文档。

4.4 产品安全性检查

ATE类需要开展的工作涉及覆盖范围分析(ATE_COV.2)、高级设计测试(ATE_DPT.1)、功能性测试(ATE_FUN.1)以及独立抽样测试(ATE_IND.2)等方面的安全保障措施。

ATE_COV.2措施证明所指定的测试能覆盖功能性描述中给出的所有安全功能。ATE_DPT.1措施确保所指定的测试能证明RS-Linux的工作行为与高级设计中确定的情况一致。ATE_FUN.1措施确保所指定的测试能证明所有的安全功能都按指定的方式进行工作。ATE_IND.2措施确保第三方能对开发人员所做的测试进行抽样重复测试，并得到开发人员所描述的测试结果。

AVA类需要开展的工作涉及指南审查(AVA_MSU.1)、TOE安全性强度评价(AVA_SOF.1)以及开发人员对脆弱性的分析(AVA_VLA.1)等方面的安全保证措施。

AVA_MSU.1措施确保文档中不存在误导性的、不合理的或冲突性的指导信息。AVA_SOF.1措施确保由口令机制实现的安全功能能够达到指定的强度级别。AVA_VLA.1措施确保开发人员对RS-Linux的所有可发行媒介都进行检查，排查用户有可能违反安全政策的任何明显途径，并证明，没有任何明显的脆弱性会在RS-Linux的使用环境中被利用。

4.5 应用指导

AGD类需要开展的工作涉及管理员指南(AGD_ADM.1)和用户指南(AGD_USR.1)等方面的安全保障措施。AGD_ADM.1措施为RS-Linux的安全性管理提供管理员指南。AGD_USR.1措施提供用户指南，用户指南描述用户可用的RS-Linux的安全功能的正确使用方法。

ADO类需要开展的工作涉及交货程序(ADO_DEL.1)以及安装、生成和启动程序(ADO_IGS.1)等方面的安全保障措施。ADO_DEL.1措施提供发行RS-Linux的交货程序，以便按照该交货程序能把RS-Linux安全地送到用户手中。ADO_IGS.1措施提供安全地安装、生成和启动RS-Linux的操作程序。

5 系统开发过程的分析

以图1中描述的三阶段开发活动模式为讨论的基础，本文的第3节指出，第一个阶段首先对安全问题进行描述，然后试图通过逐步渐进的方式定义一组能够有效地解决相应安全问题的安全功能。CC标准对ST的评价将证实第一个阶段定义的安全功能能有效地解决相应的安全问题。因而，如果一个ST通过了CC标准的评价[12]，那么，这个ST中定义的安全功能能够处理该ST中描述的安全问题，这一点是可信的。

表5 在RS-Linux开发中实施的保障措施

保障措施	步骤5	步骤6	步骤7	步骤8	步骤9
表示间的一致性	√	√		√	
安全政策模型的建立	√				
开发环境的安全性	√	√		√	
配置管理组件的覆盖面	√	√		√	
配置组件修改的授权	√	√		√	
测试：功能性描述的覆盖面	√				
测试：按高级设计工作		√		√	
测试：功能的执行	√			√	
测试：第三方抽样	√	√		√	
指南检查	√	√		√	
口令强度评价	√	√		√	
发行媒介的脆弱性检查	√	√		√	
管理员指南	√	√		√	
用户指南	√	√		√	
发行过程					√
使用准备					√

图例：√表示保障措施与相应步骤中的工作或工作结果有关

本文的第4节指出，为了在一定程度上确保一个安全产品能够实现指定的安全功能，安全保障措施在第二和第三个阶段付诸实施。CC 标准对一个安全产品的评价是以一个已通过评价的 ST 为参考依据的。一个安全产品通过了 CC 标准评价的证明可以证实，所有相应的安全保障措施都在第二和第三个阶段有效地得到了实施。由此可以断定，如果一个安全产品通过了 CC 标准的评价，那么，这个安全产品就在一定程度上正确地实现了相应 ST 中定义的安全功能，进而，这个安全产品就能够在一定程度上解决相应 ST 中描述的安全问题。

这里，我们说，一个安全产品能够在一定程度上解决某些安全问题，这个程度就是安全确信度的重要衡量指标，它是由在开发活动中实施的安全保障措施决定的。我们必须对安全保障措施加以了解，才能认识一个安全产品的安全性在哪些方面以及在多大程度上是值得相信的。就 RS-Linux 的研究实验而言，保障措施的概貌可由表5描述，它是对前面各小节的研究结果的一个概括。

CC 标准的思想内涵鼓励我们通过安全工程的应用去提高安全产品的安全特性。安全工程不能孤立于软件工程，但它是有别于软件工程的一个专业学科，它集中解决软件工程中涉及不到的安全性问题。

安全工程是一门发展中的学科，虽然至今还没有一个公认的精确定义，但它的目标是明确的，以下是这门学科的主要目标[13]。

(1)安全风险：掌握系统的应用单位可能面临的安全风险；

(2)安全需要：根据已掌握的安全风险，确立与之相应的安全需要；

(3)安全指导：把安全需要转换为安全指导，指导工程项目中其它工程学科的活动，指导对系统配置和操作的描述；

(4)安全信心：为人们建立起对安全机制之正确性和有效性的信任感，即，建立起系统的安全确信度；

(5)可接受风险：确定系统中或系统操作中由不可避免的安全脆弱性可能带来的影响是可以容忍的；

(6)安全信赖：通过在工程项目中涉及到的各个工程学科和专业方面的共同努力，使人们相信系统的安全性是可以信赖的。

结束语 要摆脱已经难以适应新的应用发展需要的传统开发模式的制约，首先必须探讨出一个能够顺应技术发展需要的切实可行的新的开发模式，遵循国际 CC 标准思想的开发模式是一个值得尝试的选择。在我们的工作之前，国际上尚未有可供借鉴的遵循 CC 标准思想的安全操作系统开发范例，我们在这方面自主地进行了探讨并取得了阶段性成果，在新的开发思想的指导下开发出了安全操作系统 RS-Linux，本文的工作可以为安全操作系统非传统开发模式的研究提供非常有意义的借鉴。

本文相关的研究成果已通过了中国科学院组织的技术鉴定，由包括五位院士在内的著名专家组成的鉴定委员会对成果的评价认为，成果整体上与国际先进水平同步。本文讨论的 RS-Linux 系统也通过了公安部计算机信息系统安全产品检验中心的检验，检验结论认为系统实现了中国国家标准 GB17859-1999[14]第三级的要求。

参 考 文 献

1 DoD 5200.28-STD, Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. National Computer Security Center, USA, Dec. 1985

2 Gligor V D, Burch E L, Chandersekaran C S, et al. On the Design and the Implementation of Secure Xenix Workstations. In: Proc. of the 1986 IEEE Symposium on Security and Privacy, Apr. 1986. 102～117

3 Gligor V D. A Guide to Understanding Covert Channel Analysis of Trusted Systems. National Computer Security Center, NCSC-TG-030, Nov. 1993

4 Gligor V D. 20 Years of Operating Systems Security. In: Proc. of the 1999 IEEE Symposium on Security and Privacy, Oakland, California, May 1999. 108～110

5 Joint Technical Committee 1. Evaluation Criteria for IT Security. ISO/IEC 15408: 1999 (E), The International Organization for Standardization and the International Electrotechnical Commission, 1999

6 Information Systems Security Organization. Labeled Security Protection Profile, Version 1.b. National Security Agency, USA, Oct. 1999

7 Smith R E. Trends in Government Endorsed Security Product Evaluations. In: 23rd National Information Systems Security Conf. Baltimore, Maryland, USA, Oct. 2000

8 Olthoff K G. Thoughts and Questions on Common Criteria Evaluations. In: 23rd National Information Systems Security Conf. Baltimore, Maryland, USA, Oct. 2000

9 Joint Technical Committee 1. Guide for Production of PPs and STs, Version 0.6. ISO/IEC JTC 1/SC 27/WG 3 N452, The International Organization for Standardization and the International Electrotechnical Commission, July 1998

10 Bell D E, LaPadula L J. Secure Computer System: Unified Exposition and MULTICS Interpretation. MTR-2997 Rev. 1, The MITRE Corporation, USA, Mar. 1976

11 Landwehr C E. Formal Models for Computer Security. ACM Computing Surveys, 1981, 13(3): 247～278

12 Common Criteria Interpretation Management Board. Common Evaluation Methodology for Information Technology Security, Part 2: Evaluation Methodology, Version 1.0. CEM-99/045, Common Criteria Project Sponsoring Organizations, Aug. 1999

13 SSE-CMM Project. Systems Security Engineering Capability Maturity Model (SSE-CMM) Model Description Document, version 2.0. National Security Agency, Office of the Secretary of Defense, and Communications Security Establishment (Canada), Apr. 1999

14 GB 17859-1999, 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. 中国国家质量技术监督局, 1999年9月13日发布, 2001年1月1日实施