第 33 卷 第 1 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.1 Computer Engineering January 2007
·安全技术· 文章编号：1000—3428(2007)01—0161—03 文献标识码：A 中图分类号：TP309

访问控制的验证测试方法研究

丁洪达[1,2]，曾庆凯[1,2]，包必显[1,2]
(1. 南京大学计算机软件新技术国家重点实验室，南京 210093；2. 南京大学计算机科学与技术系，南京 210093)

摘 要：对访问控制的评测是信息系统和产品安全评估中的一项重要内容。该文从安全标准中对访问控制的需求出发，研究了访问控制的自动测试方法，扩展了 GFAC 测试接口，并且使用该方法实现了在 Linux+RSBAC 的环境下对自主访问控制的自动测试。
关键词：自主访问控制；通用访问控制框架；自动测试；通用准则

Study of Test Approach on Access Control

DING Hongda[1,2], ZENG Qingkai[1,2], BAO Bixian[1,2]
(1. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093;
2. Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

【Abstract】 Test and validation of access control is a crucial part of the security evaluation of the system. A testing approach by extending GFAC is proposed, automatically to test the access control service according to requirements of security evaluation based on common criteria. The implementation of testing on Linux+RSBAC demonstrates the approach available.
【Key words】 Discretionary access control; Generalized framework for access control; Automation test; Common criteria(CC)

访问控制是计算机信息系统的重要功能。至今已经提出了多种访问控制模型，在TCSEC、通用准则(Common Criteria, CC)等信息安全标准中，都有对访问控制的需求描述。作为评估信息系统及其产品安全性的世界性通用准则，CC的用户数据保护类描述了安全系统中访问控制的需求[1]，大多数保护轮廓中都定义了具体的访问控制策略和功能的需求[2]。因此，访问控制的评估是信息系统和产品安全评估的重要内容。

评估一个系统的安全性需要要对系统进行测试，其目的在于检验是否满足规定的需求或弄清预期结果与实际结果的差别。通常的测试是基于执行的测试：测试案例直接在软件上运行，通过对运行中的软件输入输出情况观测和分析，判断软件是否满足需求或可能存在的问题。本文研究的访问控制测试属于此类。测试过程可以分为几个步骤：测试案例生成，测试执行，测试结果验证与测试评估。本文研究重点是测试过程的自动化，包括测试执行以及测试结果验证的自动化，即在测试执行以及测试结果验证的过程中不需要人来干预，程序能够自动完成这些工作。根据保护轮廓的需求描述来研究对访问控制的测试，主要是对系统功能的测试。因此，测试应该覆盖系统访问控制的各项功能。

本文根据 CC 中对访问控制需求的描述，研究了访问控制的自动测试方法，提出了基于测试的扩展 GFAC(Generalized Framework for Access Control)及一般的测试方法，并具体介绍了在 Linux+RSBAC(Rule Set Based Access Control)环境下测试方法的实现。

1 访问控制的测试方法

1.1 访问控制描述

一个计算机系统中的所有实体都可以被抽象为主体或客体，所有的操作可以被看作是主体对客体的访问过程，访问控制策略就是根据主客体的属性来确定主体对客体是否具有某种访问权限，其功能就是允许合法访问(具有访问权限的访问)，拒绝非法访问(不具有访问权限的访问)。

定义 1 逻辑访问 它是某个主体对某个客体的一次访问请求，可以用三元组(S，O，r)来表示，其中 S 表示主体，O 表示客体，r 表示相应的访问操作请求。

定义 2 决策结果 逻辑访问是否允许，即主体请求访问客体的操作是否允许，可以用二元组(R，ret)来表示，其中 R 表示逻辑访问，ret 表示是否允许访问。

定义 3 安全访问 如果逻辑访问的决策结果符合访问控制策略的访问规则，则称该逻辑访问是安全访问。

定义 4 安全状态 如果系统中的所有逻辑访问的决策结果都是安全访问，那么称这个系统处于安全状态。

定义 5 安全操作 如果系统处于安全状态，进行一次访问操作之后，系统仍然处于安全状态，则称该访问操作是安全操作。

1.2 访问控制的测试

访问控制验证测试的基本思想就是先测试系统是否处于安全状态，然后测试所有可能的访问操作是否是安全操作。

根据这一思路，测试的关键就在于测试系统当前状态以及系统状态转换的安全性，根据 1.1 节所给出的定义展开可以得到访问控制自动测试方法的一般步骤：

(1)分析具体实现的访问控制模型及接口(可以适当修改一下模型，增加必要的测试接口)，使测试程序能够从该接口中获取所有主体和客体的访问控制属性和任一逻辑访问的决策结果。

基金项目：国家自然科学基金资助项目(60473053)；国家“863”计划基金资助项目(2004AA147070)；江苏省自然科学基金资助项目(BK2005074)
作者简介：丁洪达(1981－)，男，硕士生，主研方向：信息安全；曾庆凯，教授；包必显，硕士生
收稿日期：2006-03-15 **E-mail：**d_hongda@163.com

(2)遍历整个系统的主体以及客体，得出系统中所有可能的逻辑访问，对每个逻辑访问进行如下测试操作：

1)获取逻辑访问的主体和客体的相应的访问控制属性。

2)获取逻辑访问的决策结果。

3)根据相应的访问控制模型策略的描述得出访问控制规则，并判断访问控制属性和决策结果是否符合规则。

(3)根据访问控制策略的描述，分析所有可能的改变访问控制信息的操作，对涉及到被修改属性的主体或客体的逻辑访问判断是否是安全访问。

(4)根据访问控制的需求描述，实现对访问控制一些附加功能的测试。

2 基于 GFAC 的访问控制测试扩展

2.1 通用访问控制框架(GFAC)

GFAC是为了在单个系统上实现多个访问控制策略而提出的[5]。如图 1 所示，GFAC把访问控制分成了两大部分：访问控制实施部分(access-control enforcement facility，AEF)和访问控制决策部分(access-control decision facility，ADF)。其访问控制过程如下：在主体访问客体前，AEF先发送请求给ADF，ADF依据主客体的安全属性(访问控制信息ACI)，结合访问控制规则(ACR)，判断访问是否许可。若不许可，AEF中断主体对客体的访问，否则主体可以继续访问客体，访问过程中有时还需要更改一些访问控制信息。

GFAC分离了访问控制实施部分和访问控制决策部分，使访问控制实施和具体的决策无关，可以方便地加入新的访问控制策略而无需修改AEF。目前，已有GFAC的国际标　准[6]：RSBAC就是一个典型的GFAC下的多安全策略支持　系统。

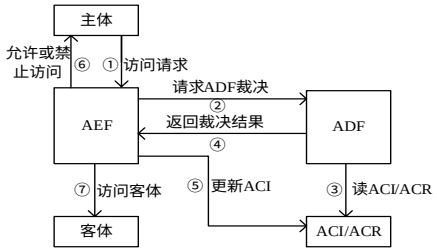

图 1 GFAC 结构

2.2 基于 GFAC 的测试扩展

GFAC 中访问控制策略的核心在 ADF，ADF 决定了主体是否有访问客体的权限，因此，测试一个系统是否正确使用了某种访问控制策略的关键也在 ADF。对 GFAC 进行必要的扩展，提供一些测试的接口以便测试程序对此框架进行测试。如图 2 所示，主要对原来的 GFAC 增加了两类接口：属性和权限。对任一主体对客体的访问，可以通过该接口获得主客体的属性，以及主体是否有对改客体访问的权限，根据这些信息，再结合访问控制策略的需求，就可以实现测试程序对该访问控制框架的测试了。

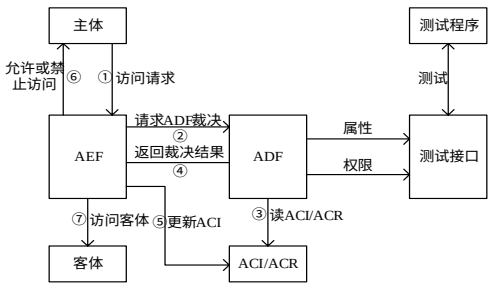

图 2 基于测试的扩展 GFAC

3 访问控制测试的实现

本测试方案实现在 fedora core 3 系统下安装 RSBAC1.2.4 并升级内核至 2.6.10 的环境下对 ACL 模块的测试。

3.1 RSBAC 简介

RSBAC是一个基于Linux的安全框架[7]，它为用户提供了更好的访问控制模型，允许灵活的选择和组合独立的访问控制模块，保证具有更好的可移植性和可扩展性等。RSBAC是基于GFAC开发的，其整个结构如图 3 所示[7]，其中只有AEF和ACI的部分是与操作系统相关的。

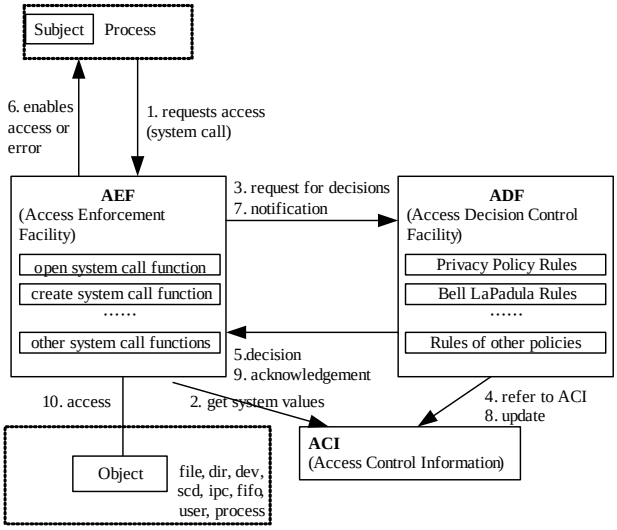

图 3 RSBAC 结构示意图

RSBAC中主体对应用户，而把客体分成多种类型[7]，如FILE、DIR、DEV等。主体对客体的请求划分得比较细，大致可分为 40 多种，如APPEND_OPEN、DELETE等。实际的读操作请求或者是写操作请求是由上面的多种操作请求组成，如读操作是由CHDIR、CLOSE等 8 个操作组成。对每个客体的定义操作并不是都包括上面有的操作，如对DIR类型的客体并没有EXECUTE操作。

3.2 ACL 自动测试的实现

ACL就是自主访问控制模型[8]，这种模型是目前在实际系统中应用较多的模型。它是在确认主体身份及所属组的基础上，根据访问者的身份和授权来决定访问模式，对访问进行限定的一种控制策略。即主体(一般指用户)能够自己决定是否将访问控制权限的某个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。

RSBAC 中的 ACL 模块是使用访问控制列表机制来实现的，每一个客体相对于每个主体都一个权限向量(64 位)，其中每一位表示主体是否具有与该位对应的对客体的访问操作。系统中所有的客体都有一个默认的权限向量的值，该默认的权限向量是从该客体的父对象继承过来的，是否继承可以通过掩码来限制。另外 ACL 中还有该模型 3 种特有的权限(Supervisor、Access Control 和 Forward)。一般来说，安全用户(一般指用户 ID 为 400 的用户)始终具有 Supervisor 权限，该用户的这个权限不能取消，因此安全用户始终具有管理访问控制属性的权限。

ACL 模块中的访问控制属性主要就是客体对应于主体的权限向量，RSBAC 提供了一个 RSYS_acl_get_rights 系统调用获得权限向量，但是 RSBAC 并没有提供逻辑访问的决策结果的接口，因此需要修改 RSBAC 的核心代码，增加系统调用 RSYS_cc_acl_adf 获得每个逻辑访问的决策结果，这

个结果的返回值有 4 种可能：NOT_GRANTED，GRANTED，DO_NOT_CARE，UNDEFINED。它们分别表示不允许、允许、系统不关心该逻辑访问和系统不可能进行的逻辑访问。由于权限只有 0(不允许)和 1(允许)两个值，而决策结果有 4 个值，因此需要根据规则来判断属性和决策结果是否一致。

规则如表 1 所示，如果客体对某个用户具有 Supervisor 权限，决策结果就不出现 NOT_GRANDTED，就可以认为系统处于安全状态，而不需要考虑其余权限是否一致。

表 1 ACL 权限与决策结果规则

权限 / 决策结果	0	1
NOT_GRANTED	一致	不一致
GRANTED	不一致	一致
DO_NOT_CARE	一致	一致
UNDEFINED	一致	一致

RSBAC 中 ACL 模块的主体即系统中的所有的用户，而客体覆盖的范围比较广，包括 FILE、DIR 等 15 种。其中 FILE、DIR 等客体是在整个系统中以文件的形式存在的，因此可以直接遍历整个系统找到这些客体，对每个可能的逻辑访问分别使用系统调用 RSYS_acl_get_rights 和 RSYS_cc_acl_adf，获得权限和决策结果，并对结果进行判断是否一致。对不是以文件形式存在的客体需要逐一判断，这些具体的课题都可以通过 RSBAC 提供的一些函数或系统调用接口获得。经过测试，RSBAC 的初始化状态是安全的。

验证了系统处于安全状态之后，可以进行下一步测试了，即要验证每个操作都是安全操作。需要验证的主要操作有加入主体和客体，修改客体对应于某个主体的访问权限向量。由于该模块的客体比较复杂，除了前 5 种客体变化比较频繁，后面的客体基本是很少变化的。因此，这里只考虑了类型为文件的客体，其余客体也可使用类似的方法来完成测试：

(1)在系统中任意增加一个文件(如 test_file 文件)。

(2)按照上面遍历系统的方法测试验证所有的主体对该文件的逻辑访问都是安全的。

(3)在系统中任意增加一个用户(如 test_user 用户)。

(4)获得新增用户对某一文件(如 test_file 文件)的访问权限(该权限应该是系统默认的权限)，验证该权限和决策结果都是一致的。

(5)不断修改文件指定的用户(如 test_user 用户)对某一文件(如 test_file 文件)的访问控制权限，验证所有可能的权限和决策结果是一致的。

4 结论

本文分析了基于 GFAC 的访问控制，并扩展了该访问控制框架，提出了基于测试的通用访问控制框架，总结了访问控制自动测试的一般步骤：从分析访问控制模型的具体实现出发，增加测试接口，并验证模型初始状态的安全性，并测试在所有可能的状态转换过程都是安全的，实现了 RSBAC 环境下自主访问控制的主要功能的测试。本文研究的重点是测试执行和测试结果验证的自动化方法，而测试案例的生成是通过对访问控制功能的描述的分析来产生的。如何根据描述有效的自动生成测试案例是今后研究的重点。

参考文献

1 The International Organization for Standardization. ISO/IEC 15408-2-1999(E) Common Criteria for Information Technology Security Evaluation——Part 2: Security Functional Requirements[S]. 1999.

2 NSA. Protection Profile for Multilevel Operating Systems in Environments Requiring Medium Robustness(Version 1.22)[Z]. 2001.

3 Stocks P, Carrington D. Test Templates: A Specification-based Testing Framework[C]// Proceedings of the 15th International Conference on Software Engineering, 1993.

4 Stocks P, Carrington D. Test Template Framework: A Specification Based Case Study[C]// Proceedings of the International Symposium on Software Testing and Analysis, 1993: 11-18.

5 Abrams M, Eggers K, LaPadula L, et al. A Generalized Framework for Access Control: An Informal Description[C]// Proceedings of the 13th National Computer Security Conference, Washington, 1990.

6 ISO. ISO/IEC 10181-3-1996 Security Framework for Open Systems: Access Control Framework[S]. 1996.

7 Amon O. The Rule Set Based Access Control Linux Kernel Security Extension[C]// Proc. of the 8th International Linux Kongress, 2001.

8 Ravi S S. Access Control: The Neglected Frontier[C]// Proceedings of the 1st Australian Conference on Information Security and Privacy, Wollongong, Australia, 1996.

（上接第 160 页）

3 结束语

本文研究了如何利用超椭圆曲线密码体制设计、实现一类高效的可追踪公平离线电子现金方案。文中提出的方案在保证系统数据信息安全性的基础上，充分发挥了超椭圆曲线密码系统密钥量小、效率高的优势。在同等安全强度下，方案可以用较小的开销(所需计算量、存储量、带宽、软件和硬件实现的规模等)和时延(加密和签名速度快)实现较高的安全性，有广阔的应用前景。

参考文献

1 Stallings W. Cryptography and Network Security Principles and Practice[M]. New Jersey: Prentice Hall Inc., 1999.

2 Nyberg K, Rueppel R. Message Recovery for Signature Schemes Based on Discrete Logarithm[J]. Design Codes and Cryptography, 1996, 7(1/2): 61-81.

3 Blundo C, Desantis A. Perfectly Secure Key Distribution for Dynamic Conferences[C]// Advances in Cryptology-Crypto’92. New York: Springer-Verlag, 1993: 471-486.

4 Avanzi R M. Aspects of Hyper-elliptic Curves over Large Prime Fields in Software Implementations, International Association for Cryptology Research 2004[C]// LNCS 3156. Springer-Verlag, 2004: 148-162.

5 周宣武，杨晓元. 网络中基于椭圆曲线密码的密钥管理方案[J]. 计算机工程, 2004, 30(11): 98-101.

6 张方国. 超椭圆曲线密码体制的研究[D]. 西安: 西安电子科技大学, 2001.