数据库安全功能测试自动化框架设计与实现

刘泊伶[1] 叶晓俊[1] 谢 丰[2] 李 斌[2]
(清华大学软件学院 北京 100084)[1] (中国信息安全测评中心 北京 100085)[2]

摘 要 数据库安全功能独立性测试是评估者使用代表性测试用例在被测数据库管理系统(DBMS)上执行,并将DBMS内部数据修改和系统输出同预期结果作比较,完成DBMS安全功能实现的评估。给出一种DBMS安全测试自动化模型及基于STAF/STAX开源框架的实现方法。最后以安全审计组件的实现为例,在Oracle和国产DBMS上给出了其用例测试及实验方法,证明了该框架的可用性。

关键词 数据库管理系统,通用准则(CC),独立性测试,测试自动化框架
中图法分类号 TP311 **文献标识码** A

DBMS Security Independence Test Framework Design and Implementation

LIU Bo-ling[1] YE Xiao-jun[1] XIE Feng[2] LI Bin[2]
(School of Software, Tsinghua University, Beijing 100084, China)[1]
(China Information Technology Security Evaluation Center, Beijing 100085, China)[2]

Abstract Security function independent testing for DBMS is an security function evaluation process during which an evaluator runs typical test cases against the DBMS under test and compares DBMS internal metadata and actual outputs with expected outputs to accomplish the evaluation on DBMS security function implementation. This paper presented a DBMS security test automation framework and its implementation which is based on an open-source framework called STAF/STAX. We practiced the audit component security test case implementation against Oracle and a domestic-produced DBMS which well proves the usability of this framework.

Keywords DBMS, Common criteria(CC), Independent test, Test automation framework

1 引言

依据通用准则[8],数据库管理系统(DBMS)的安全功能独立性测试是指评估者依据被测产品安全目标(ST),设计不同于被测DBMS开发者提供的测试用例集,通过与被测DBMS的实际交互,检查被测DBMS的安全实现是否符合其安全目标中的安全技术要求。基于通用准则的安全功能独立性测试是一种功能符合性测试,但使用的测试用例、测试方法与测试环境等可能与开发者不同。独立性测试的难点是符合安全功能规范的测试用例集设计,包括被测产品安全功能实现是否与安全功能规范相一致、是否与安全模型一致以及是否支持某种安全模式等。

换句话说,数据库安全功能独立性测试检查的是被测DBMS的运行行为是否符合安全目标中的安全组件要求,至于DBMS支撑其行为的内部实现方式是否存在缺陷并不在独立性测试范围内,因此该测试是一个黑盒测试[6]。测试的输入是与DBMS的各种交互请求,包括通用的SQL、脚本(SHELL)命令,或DBMS相关的特殊的API;测试的输出是DBMS的响应,包括内部元数据的修改或返回到外部的各种信息。传统的手工测试通过DBMS交互工具人为地向DBMS发送请求,并将DBMS的响应与预期的结果相比较,判断是否一致,来完成DBMS的安全功能评估。但是手工方式效率低、成本高,很难满足执行大量测试用例达到完全覆盖的安全功能测试需求,因此人们一直在研究安全功能测试的自动化技术。

测试自动化技术的使用改进了软件测试的工作效率和测试质量,它有效地应用于多个软件领域。针对不同的应用,测试自动化框架的设计方法也分为多种,主要有Web或软件界面测试、软件系统行为测试以及软件系统内部测试。安全功能独立性测试是一种系统行为测试。文献[1]提出了一种测试界面功能的测试自动化框架,其思想是将界面操作和操作元素抽象为关键字和参数,测试用例基于这些关键字和参数,通过检查界面对象完成测试。文献[2]综合STAF/STAX和Fit/FitNesse形成自定义的测试自动化框架,原理是使用STAX作执行引擎,用Fit表的形式描述测试流程。该框架应用于性能测试和压力测试。文献[3]的Apex测试框架提供创建测试和执行测试的内置支持,用于Apex代码的单元测试或仿真用户行为,即对白盒测试和黑盒测试均适用。

安全功能独立性测试是数据库管理系统信息安全评估的重要内容,其测试自动化是提高安全评估工作效率的重要手

到稿日期:2011-03-21 返修日期:2011-06-23 本文受国家"核高基"科技重大专项(2009ZX01045-004-001-03)资助。
刘泊伶(1983-),女,硕士生,主要研究方向为数据库安全,E-mail:liu-bl09@mails.tsinghua.edu.cn;叶晓俊(1964-),男,教授,主要研究方向为数据库技术、数据安全。

段。文献[4]展示了一款 DBMS 效率测试自动化框架，包括测试数据库生成器、用例构建自动化和用例执行自动化工具。该框架用于执行大表连接查询，其设计适用于测试 DBMS 的性能特性。文献[5]给出了一种基于保护轮廓/安全目标的安全功能自动化测试方法及工具实现。该方法依据安全模型生成相应的测试向量(用例集)，再依据安全目标确定测试数据，并用这些测试数据实例化测试步骤模板，形成相应的测试用例，交给测试运行工具即驱动程序执行。该方法在 Oracle DBMS 上进行了实验验证，但其行为模型的建立需要定义输入变量、输出变量以及中间变量等众多变量，建模过程比较繁琐，且它必须先定义安全行为模型产生的测试向量。

本文提出一种覆盖测试用例开发、自动执行和测试结果分析比较的 DBMS 安全功能测试自动化过程模型。基于 STAF/STAX[7] 开源框架，给出了支持这个过程模型的工具(DBMS Security Function Testing Tool，DSFT2)实现技术，并通过基于数据库保护轮廓(DBMS PP)的某个审计组件独立性测试展示了该工具在具体 DBMS 产品安全功能评估上的应用过程及其相关技术与方法。

2 数据库安全功能自动化测试

根据基于规约的测试特点，基于通用准则的 DBMS 安全功能自动化测试过程模型的基本过程分 3 步(见图 1)。

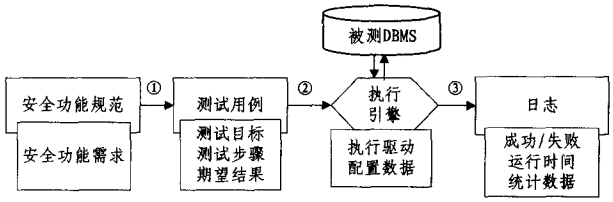

图 1 DBMS 安全功能测试自动化基本过程模型

第 1 步 安全功能符合性测试用例集开发。根据安全目标和安全功能规范中定义的安全功能组件开发测试用例集，每个测试用例描述包括测试目标、测试步骤和预期结果。测试目标描述了该测试用例覆盖的安全目标功能组件或组件元素；测试步骤给出了发送给 DBMS 的动作列表，即数据库请求序列；预期结果描述了期望被测 DBMS 的响应信息、DBMS 内部状态、异常提示等。

第 2 步 测试脚本开发与测试场景设计。一是使用 DBMS 的开发接口或 SQL 语言选择合适的开发环境，编写测试用例运行的程序或脚本；二是依据测试目标设计测试场景，配置参数输入到执行引擎的测试用例子集，即根据安全目标和测试场景配置数据将上面开发的测试用例映射为可执行的测试脚本，这样由测试引擎中的驱动程序执行测试脚本。

第 3 步 自动化测试与结果分析。使用自动化执行引擎依据测试场景描述驱动相应测试脚本的执行顺序运行，即执行引擎将测试脚本发送给被测 DBMS，通过它们的交互，生成测试日志。最后依据测试输出和测试用例集期望结果的对比分析，判断测试用例的成功或失败，并对成功/失败的用例数进行统计分析，依据通用准则评估方法(CEM)生成测试报告。

3 基于 STAX 的 DBMS 测试自动化实现

STAF/STAX[7] 是一款分布式开源自动化框架。STAF 及其调用代理提供了重用组件任务，其目的是简化自动化测试过程管理。基于这个框架实现的每个 DBMS 测试用例-STAF 的服务提供了 DBMS 安全功能测试自动化过程特定的功能集(比如记录日志、文件传输等)。STAX 提供了支持图 1 过程模型第二步的测试用例自动化执行和管理引擎。

基于 STAX 开发 IT 产品安全功能独立性自动化测试工具，安全评估人员在工具修改中主要完成两方面的任务。

1)测试场景描述：根据 DBMS PP/ST 的安全功能要求以及安全组件的测试场景设计，使用 XML 描述每个测试场景相关的测试单元中的测试用例序列。

2)STAX 服务实现：面向 STAF 服务调用需求，将安全功能测试用例集中与 DBMS 安全功能交互所需要的 DBMS API 和各种功能调用(如 SQL 中的安全控制语句)，使用 STAX 服务开发工具进行实现，并以服务方式部署到 STAF/STAX 运行环境中。

图 2 给出了基于 STAX 开发技术的数据库安全功能测试工具(DSFT2)实现框架。对应于图 1 的 DBMS 安全功能测试自动化基本过程模型的 3 个步骤，自动化测试实施分 3 个层次：测试场景的描述、测试引擎的驱动和 DBMS 安全功能测试服务。

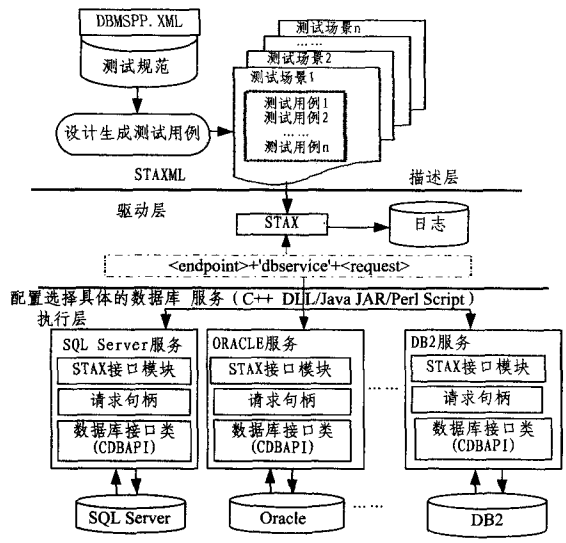

图 2 DSFT2 框架

层次 1 测试场景描述。通过 STAXML 描述独立性测试相关的测试流程，获取测试主机名、测试数据库名、测试用户名以及测试密码等配置数据。安全功能组件的测试通过测试场景来组织测试用例，即测试用例用 STAXML 描述，多个测试用例组成在一起形成的一个作业(STAX job)对应一个测试场景。测试人员在 STAF 配置文件中完成测试场景参数的赋值工作：将服务名指定为具体 DBMS 的某测试服务，对其他测试变量进行赋值。

层次 2 测试引擎驱动。驱动任务由 STAX 引擎完成。STAX 引擎驱动测试用例执行的过程是：以 STAX job 为输入，识别、解析 STAX job 描述的测试流程，向执行层传递 STAX job 中指定的命令并接收响应。在 STAX 执行测试的过程中会产生测试日志，日志的内容是实际交互的请求-响应序列，由指定的日志服务将日志内容输出到文件。STAX 根据配置文件中定义好的日志输出参数(如日志目录的路径)，生成测试日志。

层次 3 测试服务的运行：STAX 的测试执行过程就是运行 DBMS 安全服务的过程。支持 DBMS 功能的测试服务

是通过 STAX 接口模块接受 STAX 的服务请求、验证并处理请求、定义数据库命令、向被测数据库发送命令并获取数据库的响应、将数据库的响应封装为 STAX 识别的格式并通过 STAX 接口模块返回到 STAX。因此，面对不同测试 DBMS 安全评估需求，使用 STAX 服务就可封装安全功能符合性测试用例集的测试执行服务。

4 工具实践

4.1 设计测试用例

测试场景的描述可由通用准则第 2 部分[8]、DBMS 保护轮廓或安全目标的安全组件或更低层级的“元素”进行组织描述。该场景描述可以抽象为一个五元组：

(环境 e，主体 s，操作 op，客体 o，结果 r)

这个五元组描述了在 e 的环境下，主体 s 在客体 o 上进行了 op 操作，以及预期的结果是 r。测试用例集设计即是根据安全功能组件元素和预期的测试输出，设计不同的测试场景，以五元组对安全目标中的功能规范产生的测试用例进行形式化描述。

对于测试场景五元组中的每个元，依据被测数据库的状态赋予有代表性的值。赋值后的元组成为可脚本化的测试用例集。要赋予哪些值是依据安全功能组件的元素中的参数说明和被测 DBMS 的实例数据。这样测试人员就根据参数说明，结合测试目标和 DBMS 状态，使用 STAXML 描述相应的值或列表。当参数值是一个列表给定的范围域时，需要从该域中选择参数值。选择参数值的一种方法是类别划分(Category Partition，CP)[12]。CP 方法建议测试员将参数域分成若干子集(称为划分)，划分的依据是同子集中所有的点可从测试目标中导出相似的行为，测试元应从每个划分中选一个值构成参数的值的集合。

以安全审计类层次中的一个元素为例说明如何设计测试场景并由参数值生成测试用例。安全审计类中的数据生成子类包含审计数据产生组件 FAU_GEN. 1 和用户身份关联组件 FAU_GEN. 2。以组件 FAU_GEN. 1 为例，它包含了两个元素：要求在哪些场合下产生记录的元素 FAU_GEN. 1. 1 和要求产生的审计记录包含哪些内容的元素 FAU_GEN. 1. 2。

元素是安全规范定义中最小的层级，描述了一个比较具体的、可操作的安全要求。因此把测试场景的描述对象定位到元素一级，将元素作为测试场景的描述内容。以 Oracle 安全目标中对应的元素 FAU_GEN. 1. 1 为例，其 FAU_GEN. 1. 1 的内容如图 3 所示。

系统应能为下述可审计事件产生审计记录：

1)审计功能的启动和关闭；

2)有关审计最小级别的所有可审计事件；

3)特殊的使用许可(比如，通常被授权管理者用来获得访问控制策略的事件)。

图 3 Oracle ST 中 FAU_GEN. 1. 1 的内容

其中最小级别的审计事件表，在 US DBMS PP[9] 里的表 8 中定义了相应的审计事件，因此这里不再列出。事件中定义了操作，并隐式地给出了该操作在被测 DBMS 中对应的主体和客体。例如，开启审计功能事件，定义了开启操作，同时给出了执行开启操作的主体应是有开启权限的角色，客体是审计功能子系统。

根据安全组件每个元素描述的场景，基于数据库安全模式最佳实践[11]，就可整理符合上述元组语义的测试场景五元组实例(即对元组的各个要素进行赋值)，得到部分测试用例，如表 1 所列。

表 1 从场景中抽取元素设计的测试用例

ID	E	S	OP	O	R
#1	DBMS 能审计	DBA	关闭	DBMS	产生审计记录
#2	DBMS 能审计	DBA	开启	DBMS	产生审计记录
#3	u 能选择 t	u	SEL	t	产生审计记录
#4	u 能插入 t	u	UPD	t	产生审计记录
#5	u 能更新 t	u	INS	t	产生审计记录
#6	u 能删除 t	u	DEL	t	产生审计记录
…	…	…	..	…	…

测试环境在本文中指的是测试某个安全功能的数据库状态。比如，对于表 1 的 FAU_GEN. 1. 1 的测试用例，所有需要用到的环境是测试数据库审计功能应该设置的审计选项、一个具有连接和访问表权限的用户、一系列用来产生审计数据的数据表及其数据。由于场景下的多个测试用例需要的环境没有冲突，因此可以将它们所需要的这些测试环境在场景的条件中实现，当所有用例都执行完毕后一并清除环境。一个场景中的用例的相关性在于它们从不同的方面描述该场景，通常共用这个场景的测试环境、DBMS 安全状态以及测试条件等。

4.2 测试用例实现

在 STAF 框架中的测试用例需要用 STAXML 语言编写成 job 的形式，并以 STAX 服务实现测试用例。也就是说，通过编写测试用例的可执行 STAXML 文档，即测试脚本，实现测试用例的自动化执行程序。

测试人员编写测试用例的 STAXML 脚本，以控制测试流程和使用相关功能。在控制流程中，会调用到 STAF 的服务。在这里，服务相当于一个类，服务中支持的请求相当于类中的方法。同样地，数据库服务就相当于支持数据库操作的类，使用这样的“数据库类”，我们可以对数据库的对象(如表、视图等)进行各种操作(如插入、删除等)。

比如，用 STAXML 描述图 4 所示的测试用例的流程。首先，流程中的各个结点，如管理员做的数据库 DDL 操作(准备工作)等可以在 STAXML 通过调用数据库 SQL 请求服务来执行，格式为“端点＋服务＋请求”。其次，流程的顺序可以用顺序标签 sequence 来表示，测试用例的结构可以用标签 testcase 标记。其中需要对用户名等测试数据进行的赋值操作，可以用标签 script 标记，然后在标记范围内编写 python 脚本，执行变量赋值、表达式运算等操作。通过描述流程和流程中的各个 SQL 请求，就实现了用例的脚本化。

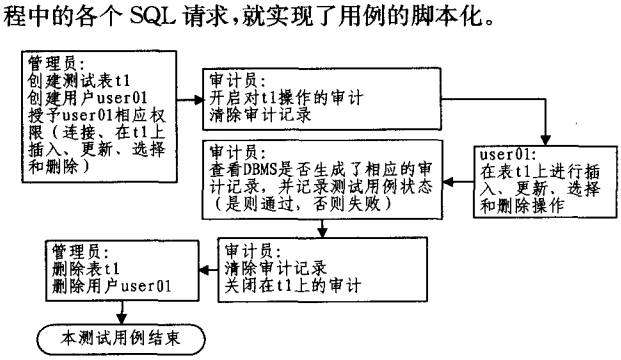

图 4 安全审计数据生成的测试用例的流程

4.3 测试环境配置

测试环境配置的目的是为测试用例自动化运行做准备。

它包括3个方面的工作:修改被测DBMS的设置、安装DBMS测试服务文件、修改STAF/STAX运行配置文件。

修改被测DBMS设置是指测试场景必须在DBMS服务器上执行的特权操作,或者只能通过DBMS提供的工具进行的操作。例如,修改DBMS的错误连接数目,再重启DBMS使修改生效,就需要在DBMS服务器上修改其测试环境配置参数,包括数据库配置文件。

安装DBMS测试服务文件是指把待测DBMS的STAF服务文件拷贝到测试机的STAF安装目录的bin目录下,以使得STAX能够找到使用STAXML测试场景描述的测试用例程序。

修改配置文件是要在测试机的配置文件staf.cfg中设置所有测试脚本要用到的参数值,比如DBMS服务名映射到的具体的DBMS服务文件、本机上各个job共用的变量、本机的自定义机器名以及日志所在的目录等。

4.4 运行及结果

运行测试用例有两种方式:图形界面的STAX监控和交互式输入命令。相应地,查看运行日志也有两种方式:使用图形界面的STAX监控器进行查看和在命令窗口中输入命令进行查看。仍然以FAU_GEN.1.1为例,分别在Oracle DBMS和一款国产数据库上自动化运行了#3-#6这4条测试用例,运行前配置变量如表2、表3所列。

表2 运行时通用变量(与具体DBMS无关的变量)

终端	服务名	服务器名	DBA	DBA psd	DB服务
local	dbserv	testdb	testdba	testpsd	dbs

表3 运行时DBMS专用变量(针对具体DBMS的变量)

变量名	Oracle数据库	国产数据库
审计表	sys.aud$	sysauditrecords
审计视图	user_audit_trail	sysauditrecords
主体字段名	username	username

安全审计测试的方法是触发审计事件的发生,通过检验被测DBMS在审计事件发生后的状态(如生成了相应的审计记录)是否符合期望来判断安全审计需求是否满足。在实际测试中,各个DBMS用的审计方式可能不同,因此可能需要更新变量赋值或者局部更新控制流程。其中的难点在于不同的DBMS审计记录内容与格式不同,从而影响测试自动化。目前解决的方法是采用模糊比较,即挑选关键的字符串进行部分匹配。

表4 测试用例运行情况

安全组件ID	安全组件名称	测试场景数(安全元素数)	测试用例数
FAU_GEN.1	审计数据产生	2	7
FAU_GEN.2	用户身份关联	1	3
FAU_SAR.1	安全审计查阅	1	2
FAU_SAR.2	限制审计查阅	1	3
FAU_SAR.3	可选审计查阅	1	2
FAU_SEL.1	审计事件选择	1	9
FAU_STG.1	审计事件存储保护	2	2
FAU_STG.4	审计数据防丢失	1	2
FAU_ARP.1	安全告警	1	2
FAU_SAA.1	潜在侵害分析	2	3
总计		13	35

本次实验依据DBMS PP设计了安全审计功能的35条测试用例,编写了相应的STAXML脚本。执行过程中测试用例覆盖安全需求的情况见表4。由于设计用例是从每一条安全元素中抽取场景要素而形成测试用例,因此每个安全元素要求至少有一个测试用例覆盖。

结束语 测试用例集设计和测试用例的自动化执行是保证数据库安全功能独立性测试正确性和有效性的关键。依据通用准则评估方法,基于开源测试框架,本文设计了一个支持DBMS安全功能独立性测试的模型和技术方法,最后以DBMS PP中的某个安全组件测试评估为例说明了我们实现的DSFT2工具的自动化测试流程。通过实践,我们认为:

测试框架能支持DBMS的安全功能独立性测试。依据DBMS PP设计测试用例集,基于STAXML开发测试服务,最后基于STAX引擎自动化驱动测试,验证DBMS的安全功能实现是否符合其安全目标/保护轮廓。

为保证测试工具与通用准则评估方法(CEM)[10]的一致性,测试工具的产品形态是下一步工作的重点。另外,基于规范的测试用例自动化生成是保证测试用例完备性的关键。基于自动机等形式化模型,通过对测试场景的形式化建模,再依据基于模型的测试用例生成方法自动生成测试用例是安全功能独立性测试工作深入的难点和亮点。

参考文献

[1] Zhu X C, Zhou B, Li J F, et al. A test automation solution on gui functional test[C]// 6th IEEE International Conference on Industrial Informatics(INDIN). July 2008:1413-1418

[2] Kim E H, Na J C, Ryoo S M. Test automation framework for implementing continuous integration[C]// 6th International Conference on Information Technology: New Generations. April 2009:784-789

[3] Mathew R, Spraetz R. Test automation on a SaaS platform[C]// 2nd International Conference on Software Testing, Verification, and Validation(ICST). April 2009:317-325

[4] Lo E, Binnig C, Kossmann D, et al. A framework for testing DBMS features[J]. VLDB Journal, 2010, 19(2):203-230

[5] Chandramouli R, Blackburn M. Automated testing of security functions using a combined model & interface driven approach [C]//37th International Conference on System Sciences. Janua-ry 2004:4779-4788

[6] Laurie W. Testing Overview and Black-box Testing Techniques [EB/OL]. http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf, 2006

[7] IBM Inc. Getting Started With STAF [EB/OL]. http://staf.sourceforge.net/current/STAFGS.pdf, 2009

[8] ISO/IEC 15408-2-2009. Common Criteria for Information Technology Security Evaluation[S]. 2009

[9] NIST. US Government Protection Profile for Database Management Systems Version 1.3 [EB/OL]. http://www.niap-ccevs.org/pp/pp_dbms_v1.3.pdf, 2010

[10] ISO/IEC 18045-2009. Common Methodology for Information Technology Security Evaluation[S]. 2009

[11] Sturm A, Abramov J, Shoval P. Validating and Implementing Security Patterns for Database Applications[EB/OL]. http://grace-center.jp/downloads/GRACE-TR-2009-07.pdf#page=42, 2009

[12] Ostrand T J, Balcer M J. The category-partitionmethod for specifying and generating fuctional tests[J]. Communications of the ACM, 1988, 31(6):676-686