EJB 组件的安全机制与安全性设计

Security Mechanism and Security Design of EJB Component

魏楚元 李陶深 易嵩杰 （南宁广西大学计算机与电子信息学院 530004）

摘要：组件的安全是基于组件的软件系统的关键问题。本文从考虑中间件安全的角度，分析了组件安全问题产生的原因，结合 J2EE 平台下的 EJB 组件开发技术，深入分析了 EJB 的安全机制。参照 EJB 组件安全性设计的声明方式和可编程方式，讨论了开发安全 EJB 组件的方法，并探索了将其应用于构建安全的基于 EJB 组件的分布式应用系统。

关键词：中间件 组件 组件安全 EJB

1 引言

过去 10 年来，随着计算机技术的发展，电子商务和电子政务等应用的兴起，软件组件化的开发无疑会加快这些行业信息化的进程，但是这些行业对安全的高要求使得软件开发商必须满足企业应用的安全需求，保障计算机应用系统的安全。组件安全性是近年来国内外学术界和工业界研究的热点问题，组件安全性的研究对解决整个信息系统的安全问题至关重要，因此探索一种行之有效的开发适应性强和安全性能好的软件组件的技术或方法有重要的实践意义。国内外对软件组件的安全问题的研究也提出了多种方法，形式化方面主要有：文献[1]提出了一种实现组件体系安全的数学模型的方法，从形式上论证组件的安全性；文献[2]提出了开发安全的自适应的可信组件的方法；工程实践方面主要侧重于研究中间件的安全机制，以 CORBA、COM/DCOM、EJB 三大组件技术为主，研究在 Microsoft 的. NET 或 Sun 的 J2EE 平台下组件的安全性设计方法。本文着重讨论了 J2EE 平台下开发企业信息系统的核心技术—EJB 组件的安全性设计的方法。

2 组件的安全

开发一个复杂的软件系统，通常需要不同功能的软构件，现实中经常会采用一些已开发出来的具备所需功能的组件，但是这些组件是否含有恶意代码等形式的潜在攻击并不是轻易获知的，并且使用其它组件或者同其它组件交互的组件也需要一些方法保护自己以免遭到其它组件的恶意的侵犯。这种情况最常见的是使用从网络上下载的一些不被信任的开发者所提供的组件。因此，组件的安全也是从这个问题中产生。另一个方面就是开发人员在开发过程中对安全问题考虑不足或者缺乏正确的方法而产生的。无论是软件开发商还是企业自身更注重的是加强物理设备一级的安全，如构筑防火墙、VPN、内外网隔离等，显然这是一种很重要的保障措施。但是对于软件漏洞层出不穷，黑客对软件漏洞的扫描、入侵攻击破坏等安全隐患反映了人们对软件开发过程中加强安全机制的考虑不足，往往在开发过程中忽略了在其它层的实施相应的安全策略和技术，例如对组件自身安全性设计的考虑。组件安全性被用来封装在处理软件组件的应用和分布时出现的安全问题[3]。因此，研究组件安全性，从组件的设计上加强安全是一个重要的途径。关于软件系统中组件的安全问题，国内外都有与组件安全相关的标准和评价准则，文献[5]做了详细的描述。组件安全性是组件质量标准评价的一个重要因素，一般可以从自主访问控制、身份鉴别、数据完整性、客体重用、审计机制、强制访问控制、标记、隐蔽信道分析、可信路径与可信分析等方面来评价组件的安全性能，也为我们设计组件提出了要求和目标。

3 EJB 的安全机制

EJB（Enterprise JavaBean）是一个关于用 Java 语言开发的可部署的服务器端组件的组件体系结构。在一

个注重安全问题的企业应用中，EJB 和所有其他分布式对象一样，也必须受到安全保护。

EJB 组件的安全性在很大程度上依赖于 EJB 容器环境提供的支持。实现 EJB 安全的机制可以分为：J2EE 和 EJB 规范要求的标准安全机制，EJB 容器/服务器特有的(即供应商私有的)安全机制。

3.1 EJB 的安全模型

EJB 的安全模型[7]是简单的：当客户端程序调用目标 Bean 的方法时，与调用客户端相关的用户身份被传递到目标端的容器中，容器检查 Bean 调用者的身份是否在与 Bean 被调用方法相关联的存取控制策略中，如果与调用者的身份匹配，容器允许它调用方法否则就拒绝它。EJB 应用中的安全管理可以配置和部署成最适合企业运作环境的应用的安全策略，EJB 规范支持 Bean 的集成者定义应用程序的逻辑安全角色，即基于特定应用的客户端身份的集合。部署者则要求针对目标运行环境中定义的特定"用户"和"角色"信息，在客户试图访问部署在应用服务器中的 EJB 应用之前，这些信息被用来认证客户，客户的身份必须是具有执行该项操作的安全角色才能被授权执行该项操作。一个安全角色就是应用程序的一个特定类型的用户为了访问应用程序资源必须具有的权限的一个语义上的分组。[4]定义在应用域中的逻辑安全角色被映射到运行域中的用户/角色，为 EJB 应用提供了一个集成的安全框架。

3.2 EJB 容器的安全

在 J2EE 平台下，EJB 的核心组件是 EJB 容器或应用服务器，EJB 组件的安全是由它们的容器负责的，不需要或者很少需要 EJB 开发人员专门编写有关安全的代码，实现了安全逻辑和业务逻辑的分离。EJB 容器提供了一个包含安全等额外服务的一个运行期环境，Bean 的提供者通常只负责提供已经定义好的实施商务逻辑的接口；应用集成者负责把这些 Bean 集成为一个完整的应用，但是并不清楚它们的内部工作方式或结构；部署者清楚地了解安全需求，但是他不能随意更改 Bean 的代码，他只负责在一个 EJB 容器或应用服务器中安装和运行已集成的应用程序。

EJB 支持基于角色的访问控制策略，角色可以将访问控制策略中的不同用户划分为用户组，具有良好的可扩展性和管理性，角色提供了允许 Bean 提供者、应用集成者和部署者的之间的任务分离所需的级别：Bean 提供者和应用集成者指定一般的角色，部署者则把个别的用户身份同这些角色关联起来。安全策略在应用程序之外的一个 XML 文件描述器中指定，由部署者在 EJB 容器中实施。图 1 描述了 Bean 提供者、应用集成者和部署者有关安全的任务。

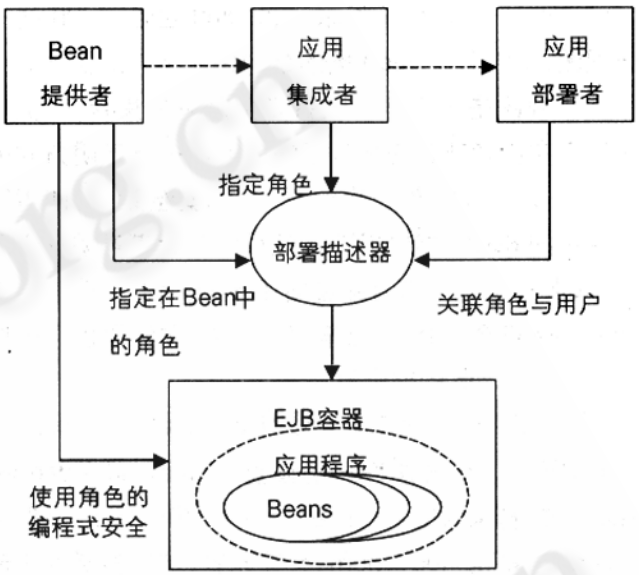

图 1　Bean 提供者、集成者、部署者

3.3 基于特定容器供应商的安全

EJB 规范规定容器供应商必须提供 EJB 组件部署者实施安全策略必要的安全机制，但并不指定必须被 EJB 容器支持和实施的某种确切的机制。EJB 容器提供的安全功能主要包括：主体认证、EJB 调用和资源管理器访问的访问授权和远程客户端的安全通信(保密性、完整性等)。EJB 容器为 EJB 提供了一个安全域和一个或多个主体领域，一个安全域可以被 EJB 容器实施、操作和管理，例如 EJB 容器可以存储 X.509 证书或者使用一个外部安全认证 Kerberos，但 EJB 并不指定安全域的范围，这个范围可以由应用的边界、EJB 服务器、操作系统、网络和企业等定义。对于 EJB 组件安全策略的实施，EJB 容器供应商都提供相应的部署工具用来简化安全管理的机制，例如用 XML 文件或者 LDAP 服务器来存储主体的身份信息。当前流行的应用服务器如 BEA 的 Weblogic 和 IBM 的 Websphere 等服务器都提供了十分灵活的安全域管理机制和管理工具，适合开发安全策略更为复杂的规模较大的企业应用。

4 EJB 组件安全性设计

J2EE 提供两种基于 EJB 容器的安全的形式：可编程的安全性和声明的安全性。这两种方式提供实现基于角色的访问控制策略。

4.1 声明方式的安全性

J2EE 部署描述文件 ejb－jar. xml 包含了 EJB 的安全视图，也包含了 Bean 类的结构化和参考的信息。一个安全视图是一个包含逻辑安全角色的集合，它提供了一个实施和部署安全策略的框架。执行声明方式的授权一般分两步：首先声明 Bean 的安全策略，即声明受保护的 Bean 方法的许可，然后为部署者声明安全角色。EJB 规范使用名为部署描述器的 XML 文件来支持用户部署安全策略，部署描述器 XML 文件提供了一些元素来定义 EJB 的安全策略。

（1）声明方法许可：主要是对 <method－permission> 元素进行方法许可的设置，<method－permission> 嵌套 <role－name>、<method>、<ejb－name>、<method－name> 等元素，可以灵活的声明 Bean 的安全策略。声明方法许可设置完成后，EJB 容器自动在运行期针对 Bean 方法进行安全检查。

（2）声明安全角色：<security－role－ref> 元素是安全角色的引用声明标志，它又包含 <role－name> 和 <role－link> 两个子元素，<role－name> 声明一个逻辑的安全角色；<role－link> 将 role－name 抽象安全角色映射为定义在部署描述符中真正的安全角色。<role－name> 元素的值是一个角色名称，角色描述基本是抽象的，并且独立于实际的用户和它们的安全性质，Bean 方法的存取控制策略可以使用简单的角色或方法配对来指定需要调用该方法的角色。声明安全角色后，部署员可以使用容器工具将角色映射到要点中去。

4.2 可编程方式的安全性

EJB 规范不推荐在组件之内实现编程方式的安全逻辑控制，但并非所有的安全策略都可以用声明的方式表达，EJB 体系结构允许通过使用 javax. ejb. EntityContext 接口的 isCallerInRole（String roleName）和 getCallerPrincipal（）方法，提供一种对一个调用者安全上下文的可编程访问的方式。图 2 描述了 EJB 上下文提供的安全方法接口以及实体 Bean 和会话 Bean 调用与继承安全方法的关系。

一般编程方式授权可以分为三步：

（1）编写编程方式安全逻辑。EJB 对象有两个和安全相关的方法：isCallerInRole（String roleName）检查当前调用者是否符合某一特定的安全角色；getCallerPrincipal（）方法检索当前调用者的安全信息要点，如在一个存储安全信息的数据库中检索标识调用者的名称，以判断该用户是否被授权。安全逻辑主要通过编写调用这两个主要方法实现。

（2）声明 Bean 所使用的抽象安全角色。在部署描述中声明 Bean 代码所用到的安全角色，类似于声明方式地声明抽象安全角色，用 <security－role－ref> 和 <role－name> 元素实现声明，<security－role－ref> 元素定义 Bean 从属的安全角色，以方便部署者去实现这个角色。

（3）将抽象安全角色映射到实际角色。Bean 的部署者负责产生最终的应用系统真正使用的安全角色。通过 <role－name> 和 <role－link> 元素将抽象安全角色映射到实际角色。

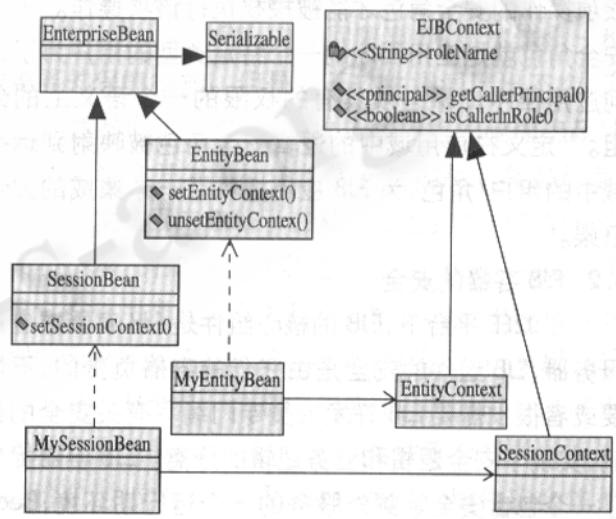

图 2　EJB 的安全 API

4.3 两种方式的比较

前面讨论了声明方式和编程方式实现 EJB 安全策略的方法。从一致性原则和事务处理角度来看，安全作为一种中间件服务，实现安全策略和商务逻辑的分离是较为理想的方法。对于声明方式，由 EJB 容器执行所有的安全操作，客户可以在部署描述中声明自己

的安全要求,但是这种方式有很多限制,它仅仅可能声明被容器实现所支持的安全策略,要实现对安全策略更为灵活的定义是不可能的,这极大的限制了安全模型的实施和更为细致复杂的安全策略的定义。从理想的角度,我们希望采用声明方式的安全检查方法来实施安全策略,但目前 EJB 规范并没有完全支持这种方式的实现,特别没有提供一种可移植方式以实现实例级的授权。对于编程方式,由编程人员在 Bean 中编写安全检查程序代码,这种实施安全策略的方法显得更为灵活,可以弥补声明方式的一些缺陷。在实际的应用中,可以结合使用这两种方式。

4.4 使用 Run – as 方法

应用集成者需要指定在一个 EJB 的方法执行期间,方法调用者的身份是否应该被使用或者一个特定的 run – as 身份应该被使用。< security – identity > 元素描述了安全身份的概念,它的值是 use – caller – identity 或 run – as。在部署描述器中定义安全身份操作对于应用集成者是可选的。应用集成者能在部署描述器中使用 run – as 元素为一个企业 Bean 定义一个 run – as 身份,run – as 身份作为一个整体应用于企业 Bean,即应用于企业 Bean 的本地、组件、Web 服务终端接口等所有方法、一个消息驱动 Bean 的所有消息监听方法、实现 TimedObject 接口的企业 Bean 的 ejbTimeout 方法和将会轮流调用的 Bean 的所有内部方法。应用集成者一般通过以下两种方式指定 run – as 身份:

(1) 使用 role – name 元素定义安全角色的名字;

(2) 有选择地使用描述元素提供主体的描述,主体被期望一定是根据它的安全角色而定的 run – as 身份。

5 结束语

组件的安全研究是当前研究热点问题。EJB 组件的安全类似于 CORBA 安全,它不仅提供中间件层的安全,而且也提供应用层的安全,EJB 安全体系允许每一个参与方在部署描述器中定义安全策略,适合于 EJB 组件的开发、集成和部署。J2EE 是一整套功能非常健壮开发企业级中间件的规范,它包含了 Java 认证和授权服务(JAAS),Java 安全套接扩展(JSSE)和 Java 加密扩展(JCE)等组成的 Java 安全扩展来保障 J2EE 开发企业级应用的安全。EJB 是 J2EE 极其重要的组成部分。采用 J2EE 架构开发适合企业分布式环境的应用系统具有很多优点,在安全保障方面,EJB 的功能组件具有更好的可靠的安全性能,因此开发功能强大、具有较好安全性的 EJB 组件,不仅符合软件复用和大规模软件构件开发的潮流,而且结合 J2EE 的安全规范,可以更为容易的实施安全策略的部署,构建满足企业应用安全需求的更为强大的安全体系,保障信息系统的安全运行。

参考文献

1 McCullough D. A Hookup Theorem for Multilevel Security [J], Software Engineering, IEEE Transactions on, 1990, 16(6).

2 Ulrich Lang. Access Policies for Middleware [EB/OL], http://www.cl.cam.ac.uk/TechReports/.

3 Bharath Kumar. Component Security [J], http://purana.csa.iisc.ernet.in/mbk/jammin/component-security.PDF.

4 Mrunal G. Dhond J2EE Security and Enterprise JavaBeans. [EB/OL] http://www.cis.ksu.edu/~mdhond.

5 Common Criteria Project Sponsoring Orgnaisations. Common Criteria for Information Security Evaluation Part 3: Security assurance requirements. Version 2.1, 1999.

6 Sun Microsystems. EJB2.1 specification [EB/OL], http://java.sun.com/products/ejb/.

7 仙人掌工作室, EJB 的安全机制 [EB/OL], http://www.ccw.com.cn/htm/center/prog/02_3_1_6.asp.