

---

# **ORACLE 9I应用服务器第2版的安全性**

*John Heimann, Oracle*

## **引言**

本文介绍Oracle9i应用服务器(Oracle9iAS)第2版的安全特性。首先概要介绍Web安全性的各项要求,然后讨论Oracle9iAS第2版中为满足这些要求而实现的安全性体系结构。本文对于Oracle9iAS第2版中引入的新的安全特性给予了特别关注。

## **WEB应用程序的安全性**

Web已经取代了客户/服务器方式,已成为各机构为其用户提供访问商务应用程序和数据的首选方案。这其中有很多不言自明的原因,包括更好的可伸缩性、降低成本以及拓展新用户和市场的机会。在互联网上部署商务应用的风险也得到了广泛的认识。这些风险包括:

- 只能了解有限的用户身份信息,
- 很难甚至无法控制用户的行为(如果你不知道用户是谁或如果他们并不是你机构中的成员,则很难对他们的错误行为进行处罚),
- 系统和数据更明显地暴露于恶意的用户之前,
- 不安全的网络可能造成信息泄露或破坏,
- 利用互联网特定的开放特性(如蠕虫、跨站点脚本等)进行攻击。

Web应用程序开发人员多年来一直致力于这些风险的解决。一些近来的趋势增加了Web应用程序安全性的复杂度,这些趋势包括通过单一信息门户网站部署多个应用程序、增加Web应用开发中的Java使用,以及增加已部署应用程序的复杂性和规模(如增加所服务的用户数)。

## **ORACLE9IAS安全性概述**

Oracle9iAS第一版中引入了Oracle9iAS单一登录(Single Sign-On, SSO)特性,它支持范围广泛的应用程序,并且在很多组件,特别是Oracle HTTP Server和Oracle9iAS Portal中,提供了安全性。在Oracle9iAS第2版中,Oracle引入了一个全面的安全性框架,来支持所有的Oracle9iAS组件以及部署在Oracle9iAS上的第三方和定制的应用程序。这个框架建立在Oracle9iAS SSO(用于认证)、Oracle Internet Directory(用于授权和建立用户规则)和Oracle Java Authentication and Authorization Service(JAAS) Provider(用于实现Java 2 Enterprise Edition(J2EE)应用程序中的安全性)的基础上。

## **ORACLE 9IAS的单一登录特性**

Oracle9iAS的一个重要安全特性是支持对基于Web应用程序的单一登录(SSO)。企业考虑采用SSO有很多原因。其中包括公司越来越多地部署基于Web的电子商务应用程序,供其员工、客户及合作伙伴来使用。如果没有SSO,每个用户都必须为她访问的每个应用程序维护一个单独的身份标识和密码。为每个用户维护多个账户和密码既不安全成本又高。

---

## 多个账号和密码是不安全的

大多数用户都记不住几个密码。需要维护一个以上登录账号的用户往往选择容易记忆的密码，为不同的账号选择同样的密码，当要求他更换密码时则重复使用旧密码，或者将密码写下来。所有这些做法都有损密码的安全性。将密码写下来或选择容易记忆（因此也容易被猜中）的密码加大了密码泄露的危险。要求其更换密码时重用旧的密码或在多个系统上使用同一个密码都增加了密码泄露所导致的可能损失。虽然很多系统实施密码管理机制，强制用户选择复杂的密码或阻止他们重复使用过期密码，但这些机制往往招致相反的结果：用户想出破坏它们的方法，因而更进一步地损失了安全性。例如，强制用户使用随机密码几乎就确保了密码一定会被记在纸上。

如果一名用户加入或离开了一家机构，或在该机构内所任的职务有变动，那么该用户对机构支持的应用程序进行访问时的权限也相应改变。每个用户的多个独立账号往往意味着相关的用户权限会在组织变动后继续保持一段时间。例如，用户账号和访问权限可能在用户离开该机构或改变职务以后很长时间还会保留在系统中。这就使系统存在着被不满的前员工攻击的可能性。

## 多个密码成本较高

为每个用户管理多个账号和密码成本较高。在很多企业部署中，系统管理员把相当大的一部分时间花在处理账号和密码相关的问题上，包括在用户加入机构时初建账号、在用户离职或职务变动时删除账号、密码忘记时重新设置密码。每个用户拥有多个账号也相应地要求成倍增加系统管理员。

系统管理员所需处理的问题中还包括不得不访问多个系统，通过多个而且可能是不一样的管理界面来增加或删除每个系统上的用户账号。

## ORACLE9IAS的SSO解决方案

Oracle的Web SSO技术为Web用户提供单一登录策略。它是为在Oracle9iAS提供的这样环境中工作而设计的，在这种环境下，对多个基于Web的应用程序可以通过该应用服务器进行访问。Oracle的SSO策略包括一些不同的技术。对于发展中的基于Web的应用程序领域，Oracle开发了一个SSO框架和Oracle9iAS SSO服务器，后者是特别为提供Web SSO而设计的。

Oracle SSO策略有很多好处。它提供通过标准协议了从浏览器客户机对基于Web的应用程序（包括Oracle应用和工具）进行安全单一登录的一个框架。它支持充分利用SSO框架的合作伙伴应用程序，以及支持外部应用程序，后者用于支持原有系统和第三方产品。合作伙伴应用程序运行在SSO框架之内，并依赖SSO的服务对用户进行认证；外部应用程序继续使用他们自己的用户名和密码。Oracle9iAS SSO策略建立在cookie的基础上，这些cookie是由合作伙伴的应用程序和名为Oracle9iAS SSO Server的集中化服务器产生的。

Unocal是使用Oracle9iAS的主要Oracle客户。他已经选择了Oracle9iAS SSO为其公司的信息门户网站myUnocal.com提供单一登录特性，并且正在致力于该标准的全球实施。myUnocal为全球的Unocal员工提供商务应用服务，支持Unocal商务数据和服务的统一。Oracle9iAS SSO

---

将使myUnocal员工只认证一次，即可访问那些他们有权访问的应用程序，而无需记住用于每个应用程序的单独用户名和密码。

## *组件*

### **ORACLE 9iAS SSO SERVER**

Oracle SSO技术的核心是Oracle 9iAS SSO Server。该技术最初是作为Oracle 9iAS Portal的一个组件引入Oracle 9iAS第一版的，但是在Oracle 9iAS第2版中，SSO Server是一个基础设施组件，并不要求安装Portal。

Oracle 9iAS SSO Server对用户进行认证，并将他们的身份标识安全地传递给合作伙伴应用程序。用户在特定时间段（通常是一天）内第一次访问系统时，它提示用户输入用户名和密码，并对用户给出的密码进行验证。

Oracle 9iAS SSO Server使用cookie，这是一些由Web服务器存储在浏览器客户机上的格式化信息。Cookie使Web服务器能够存储和获取关于客户机用户的相关信息，在无状态的Web环境中有效地维护客户机的状态信息。Cookie得到了目前所有浏览器的支持，不过也可以被用户设为无效（在这种情况下Oracle 9iAS SSO Server不提供SSO）。Cookie可以是持久型的，这意味着它们被保存在客户机的硬盘上，即使关闭了浏览器也不受影响；它们也可以是非持久型的，这种情况下浏览器关闭时它们就会被删除。当用户通过了Oracle 9iAS SSO Server的认证时，该服务器在用户的浏览器中设置一个SSO的cookie。当浏览器访问Oracle 9iAS SSO Server时，有效的SSO cookie存在就表明用户已经通过认证。

### **合作伙伴应用程序**

合作伙伴应用程序是运行在SSO框架中的应用程序。特别是，它们是为将用户认证的责任委托给Oracle 9iAS SSO Server而设计（或被改制）的。它们接受由Oracle 9iAS SSO Server提供的用户身份标识。既然合作伙伴应用程序利用Oracle 9iAS SSO Server的认证服务，它们不需要实施自己的认证模块。合作伙伴应用程序的用户管理被简化，因为无需为这些应用程序管理密码。将一个应用程序部署为合作伙伴应用程序可以同时降低开发和管理成本。

### **MOD\_OSSO**

Mod\_osso是在Oracle 9iAS第2版中引入的一个新特性。它是Oracle HTTP Server的一个扩展，可以使HTTP Server成为一个SSO合作伙伴应用程序。在HTTP Server下运行的应用程序，如servlets，可以从mod\_osso中以一个Apache报头的方式获得用户的经过认证的身份。因此，Mod\_osso可以使应用程序无需内嵌特定的合作应用程序逻辑就可以加入Oracle 9iAS SSO框架。这是使运行在Oracle 9iAS上的应用程序加入Oracle 9iAS SSO的推荐方法。

### **外部应用程序**

外部应用程序是那些保留自己的用户名和密码，不将用户认证责任委托给Oracle 9iAS SSO Server的应用程序。这些应用程序没有按在SSO框架中运行的要求进行开发和修改。

典型的外部应用程序是一个由第三方开发或部署的应用程序,如要求用户名和密码才能访问并定制类似电子邮件服务的门户网站。

既然外部应用程序不利用Oracle®iAS SSO Server的认证机制,它们就必须实施自己的认证模块并管理它们的用户密码。因为不同的外部应用程序可能有不同的Web表单来输入用户名和密码,所以支持外部应用程序可能要求对Oracle®iAS SSO Server进行应用程序特定的定制工作。用户或系统管理员可能在安装或改变用户名和密码时被要求进行一些特定的操作,如通过特殊的注册页面来输入这些信息。设置这些注册页面可能要求在Oracle®iAS SSO Server中开发应用程序特定的功能。

虽然使用mod\_osso 来部署应用程序或将它们部署为合作伙伴应用程序更好,但是对原有的应用程序进行翻新并不总是实用的。就是因为这个原因, Oracle®iAS SSO Server支持外部应用程序。在Oracle®iAS SSO Server中同时支持合作伙伴应用程序和外部应用程序为系统集成商提供了最大的灵活性,因为新的应用程序和原有的Web应用程序在一个SSO框架中都得到了支持。

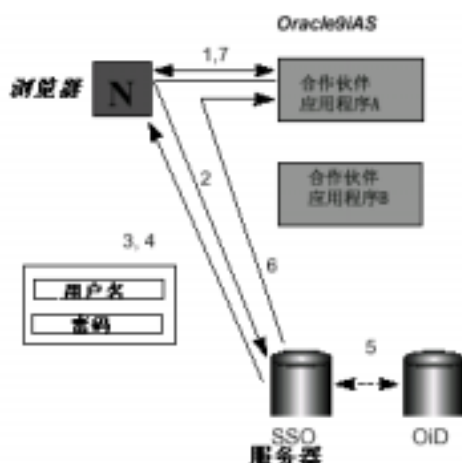
## 功能概述

### 初始认证

当一位用户试图第一次访问一个合作伙伴应用程序时,他被该应用程序重定向到Oracle®iAS SSO Server。Oracle®iAS SSO Server通过检查来确定该用户是否有一个有效的SSO cookie集;如果没有,则它要求用户提交用户名和密码进行认证。在用户提交后, Oracle®iAS SSO Server进行密码验证并在用户的浏览器中设置一个SSO cookie。该cookie被用来在与Oracle®iAS SSO Server之间的后续HTTP交互中验证客户对Oracle®iAS SSO Server的登录。

SSO cookie由Oracle®iAS SSO Server进行加密,因此不能被第三方设置或读取。Cookie在一段特定时间后就过期,(这段时间的长短取决于管理员的设置,一般是8个小时。)或者当用户关闭浏览器时被删除。与合作伙伴cookie不同,SSO cookie不是持久型的,在浏览器关闭时就会被删除。

图1: 对Oracle®iAS SSO Server和合作应用程序登录的认证



1. 用户访问合作伙伴应用程序A, 它确定用户没有被认证 (没有应用程序A的cookie) ;
2. 应用程序A将用户重定向到SSO Server
3. SSO Server显示用户名和密码页面
4. SSO Server验证密码并设置SSO cookie用于对SSO Server登录的认证
5. SSO证书可能被存储在外部服务器中 (未来)
6. SSO Server通过认证用户登录合作伙伴应用程序的加密令牌将用户重定向到合作伙伴应用程序
7. 合作伙伴应用程序A设置应用程序A cookie

---

注意，Oracle®iAS SSO Server、浏览器客户机和应用程序之间的交互都是通过标准HTTP进行的。除了它们要支持cookie以外，对客户机没有特别要求。建议在Oracle®iAS SSO Server和客户机之间启用SSL，以避免用户名、密码和SSO cookies被第三方截取，后者可能用这些信息来欺骗Oracle®iAS SSO Server。

### ***对合作伙伴应用程序登录的认证***

一旦用户经过了认证并且设置了SSO cookie，Oracle®iAS SSO Server将用户重新转给合作伙伴应用程序，并且在合作伙伴应用程序URL中提供一个包含用户身份的加密令牌。令牌用一个密钥加密，该密钥只有Oracle®iAS SSO Server和该合作伙伴应用程序知道。这使合作伙伴应用程序确信这个令牌是可信的，并且是由Oracle®iAS SSO Server生成的。

当合作伙伴应用程序接收并解开URL令牌之后，它可以确定是否给通过了验证的该用户授权访问该应用程序。要进行访问授权，它在用户的浏览器中设置一个合作伙伴应用程序cookie。这个合作伙伴应用程序cookie使该应用程序能够识别客户机用户并为其授予访问权限，而无需将用户重定向到Oracle®iAS SSO Server进行认证。合作伙伴应用程序cookie，像SSO cookies一样，在经过应用程序特定的时间段后即过期。与SSO cookie不同，合作伙伴cookie可以是持久型的，也可以是非持久型的（即在浏览器关闭后也可以继续存在，也可以不存在）。一个合作伙伴应用程序cookie的过期时间由该应用程序来决定，可能与SSO cookie的过期时间不同。

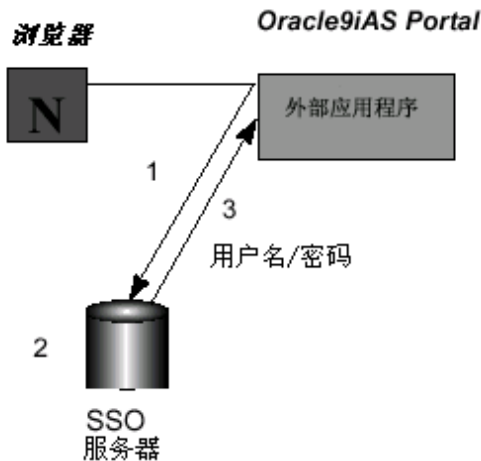
和SSO cookie的情况一样，推荐使用SSL加密以保护浏览器和合作伙伴应用程序之间的cookie交换。

### ***对外部应用程序登录的认证***

外部应用程序不能直接从Oracle®iAS SSO Server接受经过认证的身份。Oracle®iAS SSO Server为外部应用程序提供SSO功能，来支持通过Web表单的认证过程。Oracle®iAS SSO Server通过一种安全密码库机制为外部应用程序提供SSO。该密码库在Oracle®iAS SSO Server的一张数据表中维护着应用程序特定的用户名和密码。对该数据表的访问受到Oracle®iAS SSO Server的限制，密码也通过加密得到进一步的保护。当一位经过了Oracle®iAS SSO Server认证的用户需要访问一个外部应用程序时，Oracle®iAS SSO Server从密码库中获取该应用程序特定的用户名和密码，把它们形成正确的Web表单格式，并将它们提交给该应用程序。这个过程对于用户来说是透明的。

注意，Oracle®iAS SSO Server和外部应用程序之间的SSL加密可以用来防止应用程序密码在网络中泄露。

**图2：** 对外部应用程序登录的验证



1. 访问外部应用程序的客户请求被重定向到SSO Server
2. SSO Server查找外部应用程序的用户名/密码
3. 外部应用程序的用户名/密码被发送给该外部应用程序

### LDAP的集成

支持 Lightweight Directory Access Protocol (LDAP, 轻量级目录访问协议) 的目录越来越多地被用作企业内用户信息的单一信息源。这些目录为规定 (创建和配置) 和管理在企业内使用多个应用程序或服务器的用户提供一种方便的机制。这

是因为LDAP得到了广泛的支持的是互联网标准的协议, 还因为LDAP目录可用作用户信息的单一信息源, 可方便地在整个企业范围内进行访问。Oracle Internet Directory (OID) 特别适合这种类型的应用, 因为它提供安全、可伸缩的、高性能和高可用性目录服务(要了解关于OID的更多信息, 请参考题为“通过OID实现基于目录的安全性”一节)。

Oracle Oracle9iAS SSO Server使用OID来验证SSO用户名和密码。当一名用户提交一个SSO用户名和密码作为初始认证的一部分时, Oracle9iAS SSO Server将该用户名和密码与OID中维护的用户名和密码进行比较。如果比较结果一致, 则SSO用户名和密码就认为是经过了验证。注意, 在Oracle9iAS第一版中维护OID中的用户身份不是必须的 (即是可选的), 而在Oracle9iAS第2版中目前这是默认的。管理OID中的用户名和密码与在一个单一的标准集中式LDAP资料库中管理组件的用户信息的整体Oracle9i平台 (数据库和应用服务器) 策略是一致的。

在Oracle9iAS第2版中, OID通过一个Password Verifier API提供可扩展的认证功能。Password Verifier API允许OID接受用户认证数据 (以<用户名>/<密码校验器>的格式), 并通过定制或第三方的认证机制检查其合法性。因为Oracle9iAS SSO依靠OID来检查认证数据的合法性, 所以OID Password Verifier提供为Oracle9iAS SSO可扩展的认证功能, 并使它能支持广泛的认证技术。

### 第三方认证的集成

Oracle9iAS SSO为第三方认证和单一登录的集成提供了一个API; 这个特性在Oracle9iAS第1.0.2.2版中已被引入。这个API使SSO能配置成可从一个可信任的外部认证机制获取用户身份, 并能将Oracle9iAS集成到第三方产品 (如来自Netegrity 公司的Siteminder®) 提供的SSO框架中。

### PKI支持

PKI认证开始取代很多应用程序中的密码。在基于Web的应用程序中, PKI认证作为安全套接层 (SSL) 会话建立的一部分, 一般是通过X.509证书的交换来实现的。PKI本身可用来提供SSO, 因为拥有证书的用户无需输入密码就可以通过对访问多个应用程序

---

的认证。

在Oracle9iAS第2版中，用户可以通过PKI进行对Oracle9iAS SSO Server的访问认证。这可以为Oracle9iAS SSO Server支持的基于Web的应用程序以及其他支持PKI的应用程序提供单一登录。用户将通过SSL利用客户机和服务器的X.509证书交换来对Oracle9iAS HTTP Server的访问进行认证，而不是通过提供一个SSO用户名和密码进行认证。Oracle9iAS SSO Server可以从HTTP Server获得用户的SSL验证证书，并且在Oracle Internet Directory (OID)中查找到证书。如果用户的该证书被找到，则OID将向Oracle9iAS SSO Server返回用户的SSO身份。对访问合作伙伴应用程序和外部应用程序的认证则由Oracle9iAS SSO Server通过前面所述的基于cookie的方法来实现。

这个方法的好处是，在Oracle9iAS SSO Server框架中运行的应用程序在Oracle9iAS SSO Server具备PKI支持之后也自动支持PKI。Oracle9iAS SSO Server和OID负责名字映射。另外，由于获得和检查一个cookie的处理工作量远远少于执行一个SSL交换，所以使用PKI用于对登录SSO框架的初始认证和将cookie用于对合作伙伴应用程序登录的认证，比只用PKI认证的方案要性能高一些。Web应用程序的特点是其很多会话时间短，对于这种应用程序来说，这可以提高服务器的性能和吞吐量。

最后，在Oracle9iAS SSO Server中启用PKI支持允许用户通过PKI进行对访问Oracle应用程序进行认证的Oracle策略。由于Oracle应用程序通过Oracle9iAS Portal中的Application Portlet加入SSO框架，因此对Oracle9iAS SSO中的PKI支持允许对访问Oracle应用程序执行PKI认证。

### **其他安全性增强**

在Oracle9iAS第2版中，SSO包括了很多增强特性，它们提高了SSO解决方案的灵活性和安全性。这些增强中包括单一退出 (Single Sign-off)、质疑应用 (Paranoid Application) 支持和全局性非活动状态检测 (Global Inactivity Detection)。

#### *单一退出*

单一退出特性使用户不仅能终止一个SSO会话，还能终止与合作伙伴应用程序的会话。该特性非常重要，因为合作伙伴应用程序的超时时间可能比SSO会话要长，所以如果合作伙伴应用程序cookie的有效期比SSO cookie长，则用户可能没有有效的SSO会话（因为SSO会话的cookie已经过期），但是还继续拥有有效的合作伙伴应用程序会话。单一退出特性使用户能通过一次操作（例如下班回家的时候）从SSO会话和所有合作伙伴应用程序中退出。

#### *质疑应用支持*

质疑应用支持特性允许合作伙伴应用程序强制SSO服务器对一个用户进行重新认证，而无论SSO cookie是否依然有效。在该特性引入之前，如果一个合作伙伴应用程序的会话超时但SSO会话依然有效，则合作伙伴应用程序将用户重定向到SSO服务器，但SSO服务器不对用户进行重新认证，而只是简单地将用户身份返回。这个质疑应用特性

---

允许敏感的（多疑的）应用程序比SSO服务器要求更频繁的认证。它还允许事件驱动的认证，因此当用户在一个应用程序中执行某些特别敏感的操作时就可以强制对其进行重新认证。

#### 全局性非活动状态检测

合作伙伴应用程序可以通过上面所述的质疑应用特性基于非活动状态使一个会话停止。在9iAS第2版中还可以配置全局性非活动状态检测，这样，当在指定的时间段内未能成功地使用任何一个合作伙伴应用程序将会导致SSO会话的超时（结束）。该特性可以用来实施安全性策略，即如果用户在一定时间段内没有进行任何操作，则要求对其进行重新认证。

### 三层集成

Oracle9iAS SSO Server为Web客户机访问Web服务器提供单一登录特性。Web服务器越来越多地在三层体系结构中被作为中间层进行部署，在这一层中，它们可以提供对后端层数据库的访问。要求访问数据库的Web应用程序用户无需提供数据库用户名和密码来访问其中存储的数据。虽然Oracle9iAS SSO Server不支持非基于Web的应用程序，但Oracle9i数据库提供了为支持通过三层体系结构安全访问数据库而专门设计的特性。要了解更多信息，请参考Oracle白皮书《保护Oracle8i的三层体系结构系统》和《Oracle8i中的数据库安全性》。

### SSO总结

Oracle9iAS SSO提供了一个认证框架，它在部署多个Web应用程序时非常有用。SSO可以改善用户的操作体验；降低应用开发成本，因为应用程序不再需要它们自己的认证机制；降低系统管理成本；而且由于排除了多个密码的安全隐患，系统安全性可以得到提高。

### 通过OID实现基于目录的安全性

Oracle已经在LDAP方面进行了标准化，作为Oracle产品的通用机制来管理企业内用户和服务相关的信息。为支持这一点，Oracle已经开发了非常灵活、可靠和安全的符合LDAPv3标准的目录，即建立在经过验证的Oracle9i数据库技术基础上的Oracle Internet Directory (OID)。OID为Oracle产品提供LDAP目录服务，反过来Oracle产品确保它们对OID的LDAP实施。

### 目录授权

Oracle产品使用OID来管理安全性信息，特别是在各Oracle企业组件间共享的信息。这些信息包括用户身份、认证数据（如SSO密码）以及授权数据（如角色或组员资格）。在Oracle9iAS第2版中，OID是Oracle9iAS基础设施的核心组件，是用户、认证和授权信息的公共资料库。它取代了Oracle9iAS第1版中组件特定的资料库。OID提供一个描述用户权限的公用的LDAPv3标准框架。使用OID不仅为访问Oracle9iAS组件提供了一个公用的

---

权限管理机制，还提供了用于集成管理Oracle9iAS与其他符合LDAP标准的企业组件之间权限的工具。

为了在OID中管理权限，Oracle9iAS第2版还引入了代理管理服务（Delegated Administrative Services, DAS），这是一个应用程序，它允许系统管理员以及合适的用户管理OID中的用户信息。DAS包括一个基于Web的GUI应用程序和一组用于管理OID数据的API。

### 可扩展的认证

OID提供了一个集中的用户密码信息安全资料库。在Oracle9iAS第2版中，OID通过一个密码验证器（Password Verifier）API支持可扩展的认证。该API支持各种定制的和第三方认证机制，在Oracle9iAS SSO的一节中已做过介绍。

### 第三方目录支持

为提供跨多个应用程序的基于单一目录的安全构架，客户可能需要将OID与其他第三方目录产品进行集成。因此Oracle9iAS第2版引入了目录集成平台（Directory Integration Platform, DIP）。DIP提供了一个框架，用于在OID和第三方目录间建立连接器（连接软件），并且支持OID和其他目录之间的参照和同步。

## ORACLE HTTP服务器的安全性

Oracle HTTP服务器是Oracle9iAS的Oracle web服务器组件。它建立在源代码开放的Apache web服务器的基础上。Apache服务器是被最广泛采用的Web服务器产品之一，它支持大量的现有应用程序，提供一个灵活、很好理解的安全性模型。Apache是一个经过了严格验证的平台，可以在其上部署安全的应用程序。熟悉Apache的客户会发现通过Oracle HTTP Server可以很容易地构建和部署安全的Web应用程序。

### HTTP服务器安全性服务概述

Oracle HTTP Server通过各种标准和Oracle独有的增强特性（或Apache社区中被称为“mods”）对Apache进行了扩展。它使拥有Web浏览器的用户能够通过标准的Web协议访问Oracle9iAS。它提供了一个基本的HTTP侦听器功能（HTTP和安全HTTP，或者叫做HTTPS），以及通过不同的界面提供对静态网页和动态内容的访问。

Oracle HTTP Server安全性服务包括根据通过基本的质询/响应操作、客户机提供的X.509证书以及IP或主机名地址确定的用户身份来限制或允许对文件和服务的访问。

Oracle HTTP Server安全性的另一个重要特性是保护客户机和服务器之间交换的数据。这是通过SSL协议来实现的，该协议还提供数据完整性以及用户和HTTP服务器的严格认证。

另外，Oracle HTTP Server提供检测和分析入侵企图日志及其它特性。它提供与其他Oracle9iAS组件和产品（如Oracle8i数据库）的集成。通过这种方法，Oracle HTTP Server

---

为构建Web应用程序提供了一组全面的安全性服务。

## 访问控制

当URL请求抵达Oracle HTTP服务器时，这些请求要经过很多步骤的处理，这是通过对所有流行Web服务器/侦听器都通用的mod/插件体系结构来实现的。在请求处理周期的早期就进行访问控制。

Oracle HTTP Server的访问控制建立在Apache访问控制机制的基础上，后者使服务器管理员能够限制对服务器上特定文件、目录或URL的访问。对于服务器上每个受限制的对象，管理员可以通过一个指令 (*directive*) 来规定对该对象的访问是禁止或允许要取决于请求者相关的一个或多个属性 (*attributes*) 的值。管理员可以根据用户属性 (如主机名、IP地址或浏览器类型) 来配置指令，如deny、allow和order来控制进一步的处理。约束控制可以分别通过<files>、<directory>和<location>等配置指令应用到特定的文件、目录或URL格式上。

在下面的例子中，来自192.168.1.\*范围的任何IP地址或主机名为us.oracle.com的请求可被允许访问目录/internalonly/中的文件。

```
<Directory /internalonly/>
order deny,allow
deny from all
allow from 192.168.1.* us.oracle.com
</Directory>
```

注意，根据主机名来允许或限制Internet访问并不是一个提供安全性的好办法，因为主机名是很容易伪造的。采用IP地址也一样，对它的破坏也不是太难。这样，在许多情况下通过IP和主机名来进行企业内部网的访问控制就比较合理，因为内部网中的IP和主机名限制是有效的。

虽然Oracle HTTP Server是建立在开放资源Apache Server基础之上的，但它包含了一些增强的访问控制，它们可以提高安全性。例如，Apache Server通过后缀为htaccess的文件提供对每个目录/文件夹访问的约束。对这些文件的处理在Oracle HTTP Server中的默认设置中是禁止的，因为.htaccess的处理会造成安全性和性能的下降。

## 用户认证和授权

在很多应用程序中，能够根据用户身份来控制对Web服务器上资源的访问，这是人们很希望的。Oracle HTTP Server提供了多种用户认证的机制，包括在单一套接层 (SSL) 上使用X.509 证书、用户名/密码 (如在基本认证中) 和其他方式进行的客户机认证。服务器管理员可以通过Apache指令来规定，对特定URL的访问只限于必须通过特定机制进行认证的特定用户。

一旦用户通过了认证，就可以应用一些附加的规则来限制或允许对URL请求的进一

---

步处理。基于用户身份的访问控制指令可以与基于IP地址或主机名（如上所述）的指令结合起来，这样用户请求必须满足两个指令才能进行进一步处理。例如，你可以限制对一个特定URL的访问，使得只有来自企业内部网、符合特定命名方式的用户能够访问它。为此，你可以配置用于URL的指令，以便要求用户认证、拒绝除特定名字以外用户的访问，并且要求IP地址必须处于一定的范围内（确保用户在企业内部网中）。

这样，Oracle HTTP Server中实施的Apache指令访问控制机制在管理用户对对象的访问方面提供了非常大的灵活性。

## MOD\_OSSO

Mod\_ossso是Oracle9iAS第2版中Oracle HTTP Server的一个全新特性，使HTTP Server能成为支持SSO的合作伙伴应用程序。Mod\_ossso在本文Oracle9iAS SSO部分有更全面的介绍。

## 安全套接层（SSL）

安全套接层提供Oracle9iAS HTTP Server与客户机浏览器之间点到点的安全性。SSL提供的与安全性相关的服务包括认证、授权、机密性和数据完整性。这些都在以下讨论。

### SSL机密性

SSL提供的主要服务是机密性：消息被加密，因而第三方无法读取和理解。SSL使用一组标准的加密机制来加密数据以及在通信设备间分发密钥。加密的特定集合、完整性保护、选择的密钥分发算法，以及所用的加密密钥的长度等合起来定义为加密套件（*ciphersuite*）。Oracle9iAS SSL实施支持一组范围很广的标准加密套件。特别是，Oracle9iAS支持将X.509用于认证和密钥分发（也被称为PKI认证）的那些加密套件。

Oracle HTTP Server允许SSL会话进行高速缓存，这样在两个IP地址之间的多个信息交换可以在一个会话中完成。会话高速缓存对于提高性能是非常重要的。SSL会话的建立非常消耗CPU，大约要占用多达90%的可用CPU资源。SSL会话高速缓存是通过SSLSessionCache指令来指定的，其参数指定了维护SSL会话信息的文件或共享内存段。

### SSL客户机认证

SSL还可以用来提供使用X.509证书的客户机认证，作为PKI（Public Key Infrastructure，公钥基础设施）部署的一部分。Oracle HTTP Server可以配置成根据客户机X.509证书中的信息来限制对文件和服务的访问。可用来制定访问决策的信息包括客户端证书中的识别名（Distinguished Name, DN）、识别名中的描述信息、以及证书信任机构（即签发用户证书的证书权威机构）。SSL可以配置成接受它承认的信任机构，或由它承认的信任机构签发的证书等。认证可以建立在部分或全部识别名清单或含通配符的识别名基础上。

一旦进行SSL认证，证书中的信息便可用于如上所述的<directory>、<files>和

---

<location>等指令中。SSL认证可以与基本认证和基于主机的访问控制结合起来。通过这种方法，你可以允许或限制通过了SSL和基本认证相结合的用户对文件和服务进行访问，并且将这些限制与基于主机的访问控制结合起来。

Oracle9iAS第2版的一个新特性是支持对Oracle9iAS SSO访问的SSL客户机认证。请参考Oracle9iAS SSO中的PKI支持部分以了解更多信息。

### SSL环境变量

Oracle9iAS将那些与SSL会话相关的安全性信息（被称为环境变量）发送给Web服务器应用程序如CGI脚本、servlet和Perl脚本。应用程序可以使用这些环境变量，根据用户信息或代表用户建立SSL会话类型，对用户请求执行额外的访问控制或授权。

环境变量包括以下信息：

- HTTPS请求中的URL
- SSL会话中使用的加密密钥的长度
- SSL会话中使用的加密套件
- 来自客户机证书的认识名

### SSL日志

Oracle HTTP服务器还提供SSL相关信息的日志。它可以用来确定是否发生过入侵企图，以及这些入侵是否成功了。它还可以用来确定入侵攻击的来源或其他目的。

### 对ORACLE数据库的安全访问

通过Oracle9iAS可以使用一个Oracle数据库后端资料库来很轻松地构建三层系统。Oracle9iAS提供了很多访问数据库和调用数据库上应用程序的机制。最通用的机构是肥客户端JDBC和mod\_plsql，这是一个Oracle HTTP Server的插件，允许Oracle9iAS调用用Oracle数据库编程语言PL/SQL编写的数据库应用程序。因为JDBC（当运行在一个肥客户机上时）和mod\_plsql通过Oracle客户/服务器联网协议访问Oracle数据库，所以使用肥客户端JDBC或mod\_plsql的开发人员可以利用一个Oracle9i选项Oracle Advanced Security来保护Oracle9iAS和Oracle9i数据库之间交换的数据。Oracle Advanced Security提供数据加密、完整性保护以及对Oracle数据库客户机和服务器的高级认证服务。它支持业界标准加密协议（如SSL）和标准的加密算法，包括RSA的RC4、DES和3DES。

注意，Oracle与很多防火墙厂商合作，以确保由Oracle Advanced Security加密的数据得到所有领先商业防火墙产品的支持。Oracle Advanced Security确保Oracle9iAS和Oracle9i之间交换的数据是安全的，免受访问公司内部信息者的内部攻击者。

Oracle9i还提供了一个名为代理认证的特性。该特性是为解决与三层应用程序设计相关的性能问题而设计的。特别是，这使得每次Oracle9iAS切换用户背景资料时无需退出并重新登陆就可以访问Oracle9i数据库并获得特定的Oracle9i用户权限。另外，它还能

---

解决在代表已通过认证的用户访问数据库时授予中间层应用服务器的有限（而非全面的）信任度的安全问题。过去，应用程序的设计人员不得不为中间层授予超级用户的权限（如SYS或root），使它可以代表任何用户访问数据库，或者在中间层存储数据库用户密码。而这两种方案都不安全。

代理认证允许Oracle9iAS建立一个经过认证的与Oracle9i服务器的单一会话（例如用肥客户机JDBC或mod\_plsql），并且代表多个Oracle9i用户进行操作，而无需为会话中的每个用户提交单独的认证证书。应用服务器必须指定它代表哪个用户进行操作，并且还必须由Oracle9i为其分配权限以代表那个用户。另外，在制定访问决策或写事件的审核记录时，Oracle9i既可以使用经过认证的Oracle9iAS身份，也可以使用Oracle9iAS执行代理认证时所代表的用户的身份。代理认证允许Oracle9i给中间层Oracle9iAS授以有限的信任度，而无需为其授予对数据库的超级用户权限或在Oracle9iAS中存储多个数据库用户密码。

### Oracle9iAS中的JAVA安全性

Java，特别是Java2 Enterprise Edition已经成为很多新Web应用程序的开发环境选择。Java2 Enterprise Edition定义了一个Java2 Security Model（安全性模型）和一个安全性框架，被称为Java Authentication and Authorization Service（JAAS）。Oracle9iAS通过一个完全符合J2EE的JAAS Provider来实施这个框架。JAAS Provider使得用户认证、授权和委托服务对于应用开发人员是可访问的，并且允许他们将这些服务集成到J2EE应用环境中。

### JAVA2 Security Model

Java2 Security Model是由Sun Microsystems公司的Java部门定义的。它是基于功能的，并且允许开发人员规定保护域以及与这些域相关的安全性策略。安全性策略规定与运行在这些域里的Java类相关的权限集。这些权限定义了分配给特定对象的特定访问类型；如对目录/salaries/的读权限。

### JAAS Provider

Oracle9iAS JAAS Provider实施Java2 Security Model，使应用开发人员能够从JAAS提供的一组标准认证服务中获得通过了认证的用户（负责人）身份，并管理负责人访问对象的权限。它还支持权限委托，用于管理负责人调用的方法的权限。

### JAAS认证

Oracle9iAS JAAS Provider支持一个灵活的认证框架。它提供基于SSL和SSO的特别认证机制，而且允许开发人员通过标准的JAAS Login Module API集成定制的认证模块。

### SSL认证

SSL认证允许拥有客户机X.509v3证书的用户通过这些证书对访问JAAS从而对访问J2EE应用程序进行认证。SSL认证使用Oracle HTTP Server中的mod\_oss1从客户机X.509证书获得一个经过认证的用户身份，同时通过SSL交换成为合法的身份。然后这个身份

---

由JAAS提供给Java应用程序。

### SSO认证

SSO认证允许Java应用程序使用Oracle9iAS SSO进行用户认证。在这种情况下，经过认证的用户身份是从mod\_osso中获取的，并且通过JAAS提供给Java应用程序。

### 定制认证

Oracle9iAS JAAS Provider支持标准的JAAS Login Module API，该API允许开发人员将定制的认证方法集成到JAAS中。

### JAAS授权

除了提供一个全面的基于角色的授权访问控制模型之外，Oracle9iAS JAAS的实施还为开发人员管理授权时提供了体系结构上的灵活性。包括使用LDAP、通过基于XML的API使用文件系统的集中管理授权等都是可选用的。和标准的JAAS principals.xml相比，这些机制更加安全。

#### 基于LDAP的授权

Oracle9iAS JAAS用户信息和授权信息可以存储在OID中，后者是Oracle9iAS的可扩展的、安全的LDAP目录。当用户群很大时，管理LDAP中的授权非常有用，并且可伸缩性和集中管理变得很关键。

#### 基于XML的授权

Oracle9iAS JAAS Provider还支持将XML用作编码机制的授权API的快速、轻型地实施。这个API允许Java开发人员从操作系统文件而不是从OID中安全地获得用户和角色信息。这种选择对于Oracle9iAS的轻型部署很有用，因为在这种情况下不需要考虑扩展到大的用户群体。

需要注意，和principals.xml不同，其中保存的用户密码没有加密，使用Oracle9iAS JAAS XML授权时，密码经过了加密。另外，和principals.xml不同的是，Oracle实施提供了对基于角色的访问控制和Java2权限模型的全面支持。Oracle9iAS提供了移植工具，帮助从基于principals.xml的用户管理转移到Oracle9iAS JAAS XML实施。

### JAAS委托

Oracle9iAS JAAS Provider支持权限委托，允许以特定用户的权限来运行一个Java应用程序。RunAsClient和RunAsID都支持。RunAsClient意味着可以将一个Java应用程序（如enterprise bean、servlet、JSP）配置成以当前客户机用户的相关权限来运行。RunAsID表示可以将一个bean、servlet或JSP配置成以特定用户的权限来运行（如以“DBAdmin”身份运行）。这可以使开发人员在他们的应用程序中执行最低权限原则，使用户仅具有执

---

行某个功能所需的一些权限，因为用户只能在制定得很好的业务规则环境内（例如，一个enterprise bean）运用其权限。

### 访问数据库的Java应用程序的安全性

除了保护Web客户机和Java应用程序通过HTTPS（如在Oracle9iAS HTTP服务器安全性一节所描述）交换的信息外，Oracle9iAS还可以保护使用Oracle Advanced Security协议在Java应用程序和后端数据库之间交换的信息以及Oracle9i代理认证。这些特性已经在Oracle HTTP Server安全性的一节中讨论过。

### ORACLE9 IAS PORTAL的安全性

Oracle9iAS Portal是企业门户类的Oracle产品的重要组成部分。这个不断兴起的一类Web产品为企业内部网上的业务相关信息提供了一个网关。虽然最初是针对企业门户市场而设计的，但Oracle9iAS Portal可以扩展到能提供对更大的、互联网规模的社区的访问。

Oracle9iAS Portal使Oracle9iAS客户能够组织他们的Web内容和应用程序，并且以一种逻辑性很强的、一致的Web门户网站格式来提供给用户。它还提供一组工具，用来新建和管理用户及其对Oracle9iAS Portal内容的访问。

企业门户作为现有市场空间的合并和扩展，可以利用三个强大的组件：

- Oracle数据库固有的强大的信息管理技术
- 管理重要商务数据的范围广泛的应用程序，包括企业资源规划(ERP)、客户关系管理(CRM)和商务智能(BI)。
- 一个利用该技术将应用程序与内部网中其他数据中心连接在一起的框架(Oracle9iAS Portal)

内置于Oracle9iAS Portal中的功能提供了一个跨多个Oracle产品和应用程序的公用框架。购买了支持门户的Oracle产品的客户可以根据商务的需要和轻重缓急很容易地逐渐扩展门户与其他应用的集成。

这部分提供了对Oracle9iAS Portal安全性特性和体系结构的概述。它主要讨论用户和用户组的概念和表示，以及用户对数据库模式的关系。然后讨论认证、会话管理和授权等问题，以及体系结构中实施这些特性的各种组件。

### Oracle9iAS PORTAL安全性概述

Oracle9iAS Portal提供了一个将不同应用程序集成到一个单一、统一门户环境中的安全平台，并提供有效管理这个环境的各种管理工具和接口。

Oracle9iAS Portal为用户访问门户网站上的内容和应用程序提供了一个基于用户权限和分组的全面和可扩展的授权模型。它还提供了一个将应用程序连接到门户认证框架中的灵活的集成模型，使它们可以作为门户、合作伙伴或外部应用程序进行部署。Oracle9iAS Portal还通过其事件日志服务支持对安全性相关的事件审计。

---

## 门户网站的用户

在互联网计算模型中，可能有数百万的用户要访问一个门户网站，所以尽可能简便地表示用户是非常重要的。为了以安全、可伸缩和容错的方式管理大量用户和大量数据，Oracle*iAS* Portal充分利用了Oracle数据库的安全和数据管理技术。

Oracle*iAS* Portal定义了它自己的用户账号，之所以被称之为“简便的”，是因为它们并不是每一个都在Oracle数据库中有一个与其相关的独特的数据库模式。相反，每个Oracle*iAS* Portal用户账号唯一地与一个Oracle*iAS* SSO用户账号相对应。Oracle*iAS* Portal提供了一个管理机制，通过该机制来管理支持Oracle*iAS* SSO的应用程序的用户。

### 已安装的用户

当安装Oracle*iAS* Portal时，将建立一组默认的用户账号，其中包括门户网站管理员和一个公共（Public）账号。公共账号允许特定的门户网站内容能够被公开访问；也就是说，允许这些内容被没有自己的Oracle*iAS* Portal用户账号的用户访问，或者被那些虽然有账号但是还没有登录的用户访问。在默认情况下，建立两个门户网站管理账号：一个是portal，这是为同时还是相关Oracle数据库的数据库管理员（DBA）的Oracle*iAS* Portal管理员建立的；另一个是portal\_admin，它是为不需要DBA权限的Oracle*iAS* Portal管理员建立的。

### 新建用户

在Oracle*iAS*第2版中，Portal已经改用OID来管理用户。用户管理是通过新的OID DAS框架来实现的。

## 门户网站的分组支持

Oracle*iAS* Portal支持组别，可用于两个主要目的。分组提供了一种方便的授权手段，通过一次操作为一群用户授予相应的权限。另外，Portal系统中的特定属性可以与一个分组相关联，并且如果一个用户在其参数选择中指定了一个默认分组，那么这些属性均适用于这个用户的会话。例如，这些属性包括一个分组的默认主页，或一种默认的风格。

注意，分组可以由用户组成，也可以包括其他分组。这样可以构建层次化的分组。如果一个用户属于一个分组的子分组，那么按分组的级别关系他也就是父分组的成员。因此授予父分组的权限也适用于子分组的用户。

### 新建分组

要新建一个分组，用户必须具有新建分组的权限。在默认情况下，该权限可授予任何经过认证的用户，虽然可限定授予特定的用户。当一个用户新建一个分组时，他需要定义分组的名称、分组的简短描述、分组的范围是否局限于一个特定的内容领域、以及是否向其他用户隐藏该分组的存在。然后他指定分组的成员，如果他得到了Oracle*iAS*

---

Portal管理员的授权，则还可以为分组成员授予一定的权限。

一个分组可以指定为私有的。这意味着只有被分配为该分组拥有者的用户才能够在所有的分组列表中看到这个分组。其他用户，包括该分组的成员都无法看到这个分组。然而，私有属性并不会对分组的行为有任何影响，例如通过分组成员资格为用户提供权限。

## 门户网站认证

Oracle*iAS* Portal是作为一个Oracle*iAS* SSO合作伙伴应用程序来实施的，其目的是为了用户认证。注意，没有经过认证的用户可以通过公共用户账号访问Oracle*iAS* Portal上的一定内容。

## 门户网站授权

授权是根据用户的身份对访问Oracle*iAS* Portal不同内容领域进行访问的过程。一旦认证过程完成后用户的身份被确定，Oracle*iAS* Portal便根据其身份确定用户的相应权限。Oracle*iAS* Portal提供了一组扩展的权限，用于为门户网站中每个对象定义访问控制清单。下面各节描述了这个模型。

### 门户网站权限

Oracle*iAS* Portal规定的权限范围很广，可分配给用户以便对门户网站上特定对象（如网页或文件夹）进行特定类型的访问。这些权限在指定的对象类型中是分级的，较高的权限包容所有比它们低的权限。这样，如果一个用户被直接指定了关于一些特定页面的一定权限，那么她便拥有了对这些页面的所有较低级的权限。这种层次或累积方式使用应用程序可以检查一个指定的用户是否拥有执行某些操作的最低权限。

有两种类型的权限，全局权限（*Global Privileges*）和实例特定的权限（*Instance-Specific Privileges*），它们为在门户网站环境下管理对对象的访问提供了一个方便的机制。

### 全局权限

全局权限适用于特定类型的所有对象。例如，如果你被授予了ANY\_PAGE/EDIT权限，那些你可以编辑系统中的任何页面，无论你对那个页面是否拥有明确的实例特定的权限。

### 实例特定的权限

实例特定的权限被授予一个用户或分组，以限定对一个对象的特定实例，如一个特定的页面或特定的文件夹或项目，进行操作的权限。

## 应用程序集成

---

Oracle*iAS* Portal的一个关键特性是能够将不同的应用程序集成到它的框架中，为使用它来访问其商务应用程序的用户提供一种无缝的体验。Oracle*iAS* Portal依靠Oracle*iAS* SSO来实现这一点，并且支持门户网站安全性与应用程序安全性之间各种程度的集成。

当一个应用程序是在Oracle*iAS* Portal上实施的并且直接从Oracle*iAS* Portal获得用户身份时，这种集成是最紧密的。这样的应用程序被称为门户网站应用程序。Oracle*iAS* Portal本身就是Oracle*iAS* SSO的合作伙伴应用程序，可以从SSO服务器中获取用户的身份。一旦用户通过Oracle*iAS* SSO得到了认证，那么门户网站应用程序便直接从Oracle*iAS* Portal获得用户身份。

其他可以通过Oracle*iAS* Portal访问的应用程序可以是Oracle*iAS* SSO合作伙伴应用程序或外部应用程序，如Oracle*iAS* SSO节中所描述的那样。合作伙伴应用程序无需在门户网站上实施，但是必须加入也使用Oracle*iAS* Portal的Oracle*iAS* SSO认证框架中。外部应用程序保持它们自己的认证机制，不会在Oracle*iAS* SSO提供的认证框架中直接共享。

## 对HTTPS的支持

为了提高安全性，一些安装可能要求使用SSL。Oracle*iAS* SSO Server和Oracle*iAS* Portal都可以在HTTPS模式下运行。另一方面，为了保证性能，有时候更好的办法是将Oracle*iAS* SSO Server配置为运行在HTTPS模式下，而Oracle*iAS* Portal运行在HTTP模式下。

## 审计

除了监测系统未经授权就被使用的活动以外，对安全性相关的事件进行审计往往是确保被授权的用户遵守系统使用原则的最有效方法，从而“保证诚实的用户确实诚实”。

Oracle*iAS* Portal有一个日志服务，用来记录特定的安全性事件，也可以调用该服务来记录由门户网站应用程序定义的任意事件。Oracle*iAS* Portal应用程序的创建功能报表特性可用来查看日志数据。事件日志可用于审计安全性相关的事件，以及检测破坏系统安全性或违背安全策略来使用系统的可能企图。

## 结论

部署Web应用程序时安全性非常重要。Oracle*iAS*第2版提供了使用基于Apache的Oracle HTTP Server、Oracle的J2EE体系结构和Oracle*iAS* Portal来构建Web应用程序的一个可靠框架。Oracle*iAS*安全性建立在Apache提供的经过严格测试并和高度可配置的基本Web安全性服务基础上，并增加了Web单一登录、基于目录的授权和用户管理以及Java2安全性服务等全套特性，并且通过门户网站安全性和应用程序集成机制对它们进行了进一步地扩展。Oracle*iAS*还通过Oracle Advanced Security支持对Oracle数据库系统的安全访问。这些特性确保了Oracle*iAS*是构建和部署安全Web应用程序时应用服务器的最佳选择。