

第34章 高级安全选项

本章要点：

- 强力安全方针基础
- 因特网使用三层应用设计
- 防火墙：它们是什么？不是什么
- 使用高级安全选项
- 启用数据加密与校验
- ASO支持的Radius协议设备
- 安全套接字层（SSL）协议
- 对Oracle工具包的支持
- 理解多线程服务器
- 使用Oracle连接管理器
- 实例学习

随着因特网与基于 Web 应用的到来，数据库世界发生了改变，Oracle 提供了一个新的选择以适应这个新社会的挑战。

高级安全选项（ASO）提供了创建一个高度安全环境所需要的所有工具，安全环境定义为秘密的，具有完整性与可用性，这是基础。ASO 将帮助实现这些安全原则。

34.1 强力安全方针基础

ASO 具有许多选项提供了强有力的安全数据库环境。这意味着使用这个产品可以加强计算体系结构中的脆弱点。安全性通常以一个计划开始，安全性计划通常基于企业安全方针，这个方针的基础基于预期的对计算资源及不保护这些资源的相关风险的处理。计算机安全策略的基础在于机密性、完整性与可获取性。事务的机密性包括用户的验证，与 / 或发起这个事务的服务器确认这是一个授权的事务。验证通过 ASO 支持的完整验证建立，这个验证可扩展到用户、数据库及 Web 服务器。

用户使用口令进行验证，Web 服务器与数据库使用数字符号进行验证。

数据加密保证了数据处于保密状态，ASO 支持所有加密运算法则的不同级别。

为什么我们需要一个高安全性环境？

数据库必须是安全的，可阻止计算机破坏者的侵入。在过去，数据库被限制在安全的内部网络中，由于数据库对外部世界是不可获取的，所以违背安全性是罕有的。数据库管理员应该勤奋地执行相应的安全，包括：

- 确定以相应的操作系统文件级安全保护所有的外部数据库结构。

- 通过减少操作系统超级用户口令的人员与有权注册到数据库服务器的人员，限制对数据库服务器的存取。

物理地将服务器限制在一个安全的房间中，并以适当的防火墙进行保护。

提示 像所有其他的产品服务器一样，用于因特网处理的数据库服务器必须被保护（参见第23章）。使用ASO可以实现辅助安全，满足因特网数据库的需要。

由于基于Web的计算与电子商务的出现，今天数据库管理员面对的安全性挑战是一个令人畏缩的问题。网络不再是内部的安全地带，而是因特网，一个虚拟的世界，现在允许任何注册到因特网的用户存取公司数据的王冠。

34.2 因特网使用三层应用设计

用户使用瘦客户机连接到Web服务器，然后Web服务器连接到数据库（参见图34-1）。Web服务器成为数据库的客户端。而非因特网的用户使用典型的客户-数据库服务器两层配置（参见图34-2），本章后面将讨论客户端的配置，客户端可以是Web服务器或是传统的客户PC。

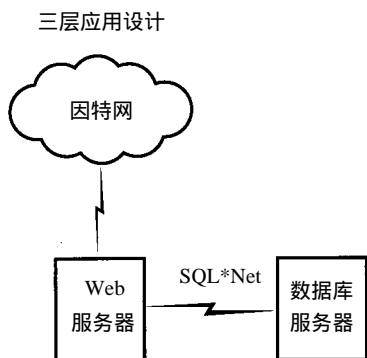


图34-1 三层应用设计

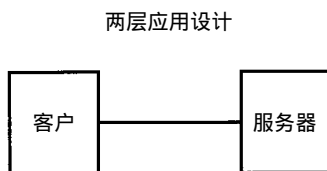
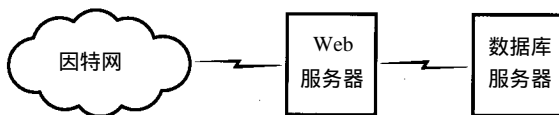


图34-2 两层应用设计

因特网会对你的公司造成威胁

因特网会对你的公司产生一个真实的威胁。计算机入侵者是活生生的，并且并不是按典型的九点至五点工作日进行工作。计算机入侵者有他们自己的协会，他们可以加入并学习怎

计算机入侵者仿效Web服务器



在计算机破坏者之后

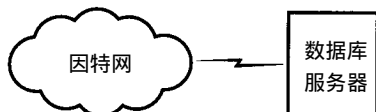


图34-3 计算机入侵者仿效一个Web服务器

样才能做得更好，做得更像一个标准的计算机专家，他们有 WEB 站点共享信息，甚至具有一本印刷出来的杂志，可以在任何稍大一点的图书链上获得。这个威胁是非常真实的，报纸上只报导了真正发生的计算机入侵中的极少部分。黑客具有探测你的网络的工具以发现你正使用的数据库服务器的类型与版本号。他们还具有工具用以猜测口令、读网络数据包及仿效 Web 服务器的 IP 地址以使你的数据库认为它正在与你的网络上的一个正式计算机进行交谈（参见图 34-3）。通过因特网存取一个公司的产品数据库需要特殊的安全性考虑，即便如此，计算机入侵者仍可进行侵袭。典型的攻击是试图击败一个或多个安全性原则，网络探测者可以查看数据包，这也许会损害你的数据的机密性。网络探测者可以使用 Windows NT 4.0 操作系统，所以即使是最可依赖的雇员也可能发现新的开发网络的方法，这些数据可能是信用卡号码或正准备与你进行交易的客户信息。

34.3 防火墙：它们是什么？不是什么

防火墙用于将安全环境与潜在的敌对环境隔离。防火墙的典型结构是通过因特网连接到一个 Web 服务器，然后由 Web 服务器通过防火墙将数据传送到数据库服务器（参见图 34-4）。防火墙担当从 Web 服务器到数据库器的入口。

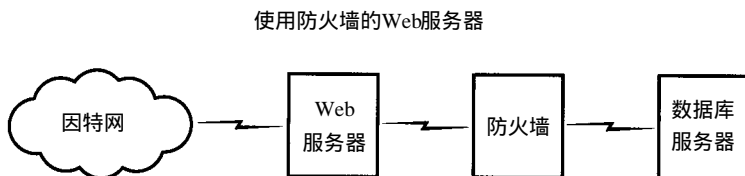


图34-4 使用防火墙的Web服务器

防火墙的目标如下：

- 锁定计算机资源的端口。
- 计算机资源间网络通信的约束。
- 约束通信的方向，即只允许网络通信沿一个方向传送。
- 限制 Web 服务器与数据库的交互类型。

34.3.1 在计算机资源中锁定端口

计算机入侵者寻找进入网络的方法，一个计算机系统上开放的端口是第一个对他们的邀请。SQL*Net 需要使用 1521 端口进行 SQL*Net 通信，不同的操作系统及 Oracle 网关可能需要使用其他的端口。如果端口是打开的，所有类型的网络通信都可以使用这个端口，而不单只是 SQL*Net。

这允许计算机入侵者使用网络协议堆栈的第七层。挑战在于以安全的方式打开 SQL*Net 使用的端口，这正是防火墙要做的事情。

34.3.2 计算机资源间网络通信的约束

因特网数据库可以被成千上万的用户访问，这些用户有些想要破坏你的公司，有些不想。防火墙允许孤立数据库所在的数据库服务器或网络。

34.3.3 约束通信方向

防火墙可以打开一个端口，允许网络通信只能入站或只能出站。图 34-5显示了工作在典型的因特网配置中的两个防火墙。防火墙 A 允许在一个指定端口的入站与出站通信，防火墙 B 只允许在一个指定端口的出站通信。这阻止了获取任何由防火墙 B 保护的网上的事务，这个产品网络可以将数据项放入因特网，而不允许任何进站通信。这阻止了计算机入侵者试图探测这个产品网络，并阻止他们损害公司的核心。

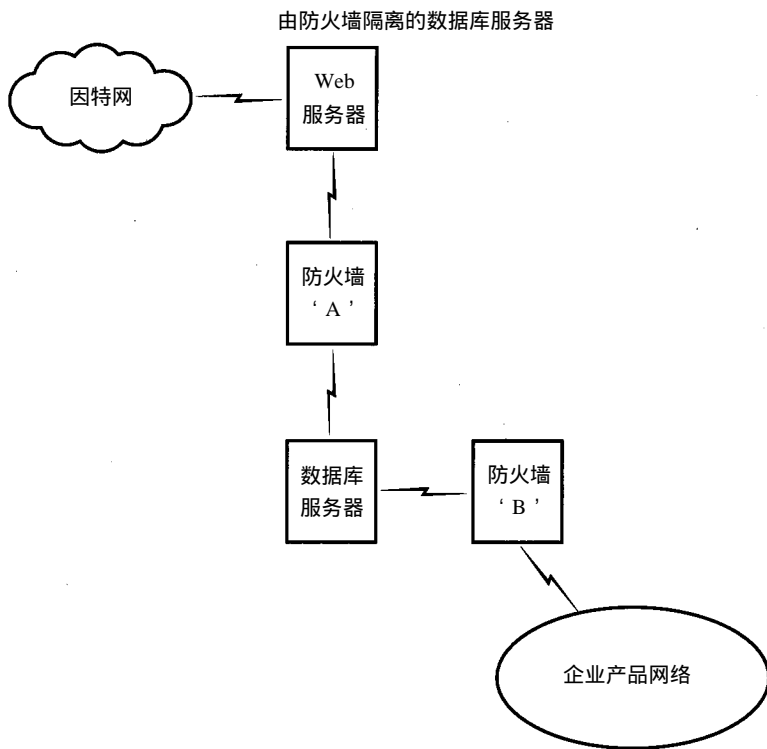


图34-5 典型的因特网配置

34.3.4 限制Web服务器与数据库交互

Web激活的数据库与公司的产品计算资源之间的事务应该被限制在仅有计算服务的必要点。例如应该关闭网络文件系统服务（NFS），这个服务已被计算机入侵者圈子开发了许多年了。

提示 当测试因特网数据库时，关闭所有的选项服务，如邮件服务、FTP、NFS。然后运行数据库，你也许会惊奇地发现你的数据库运行只需要这么少的服务。每次关闭一个服务都减少了一个计算机入侵者可以使用的区域。

防火墙怎样实现是基于公司的安全方针的，这个方针决定了防火墙设想作什么及使用何种类型的防火墙。这个方针基于一个风险分析，该风险分析显示公司的计算资源处于风险的何种级别及认为什么是可接受的风险。

提示 保护内部网络的最安全的方法是具有一个完全分开的网络用于因特网服务器。

34.3.5 防火墙类型

有三种主要的防火墙类型：筛选路由器、代理网关或防护 (guard)。

1. 筛选路由器

一个宿主/服务器使用一个路由器指引网络通信流向一个指定的网络，筛选路由器允许规则在数据包级别执行。规则必须不能太复杂，否则路由器会成为网络上的明显瓶颈。筛选路由器可以只查看包头信息并检查产生包的宿主的地址，尽管这可以进行伪造。筛选路由器可以配置为阻止来自一个指定网络的通信。它也可以用于在应用级控制通信，方法是限制端口的使用。例如，文件传输协议 (FTP) 使用端口 21。如果 FTP 应用被关闭，筛选路由器会关闭端口 21 或只允许 FTP 通信沿一条路流动。

2. 代理网关

筛选路由器只查看数据包的头，并允许有危害的传输发生，只要它们通过筛选路由器。代理网关担当因特网应用与因特网试图交互的内部网之间的真正的中间人。电子邮件，当用于因特网时，连接到公司的邮件服务器（参见图 34-6），代理网关允许因特网连接到网关，并且因特网认为它已连接到邮件应用，这允许代理网关筛选所有到来的命令并只执行允许的那些命令，这个行为对邮件服务器与 Web 服务器都是透明的。

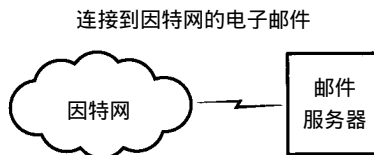


图34-6 连接到代理网关的电子邮件

3. 防火墙的防护类型

防护防火墙合并了所有代理防火墙的特性，同时增加了一些给定行为的可计算的规则。这些规则可以是合成的，并且必须经过错误检测。如果引入了错误，会危及公司计算资源的安全。

提示 更高级的防火墙现在具有 SQL*Net 的代理，这意味着当打开一个端口用于 SQL*Net 通信时，只有 SQL*Net 通信被允许通过这个端口。过去当这个端口打开用于 SQL*Net 通信时，计算机入侵者会开发这个端口作为攻击点并使用这个端口探查网络并使其曝光。

34.4 使用高级安全选项

Oracle 高级安全选项包含了一套工具，提供了在标准 Oracle 网络工具集中找不到的数据安全性与完整性级别。虽然一些特性在 Oracle 软件中可以具有，但主要的功能来自于 Oracle 网络与第三方安全性与验证产品的集成。

高级安全选项提供的功能为：

数据加密或校验和——使用高级安全选项，可以保证传过网络数据的安全性，方法是当数据流在客户端与服务器间传递时，对数据流加密。也可以保护数据在客户端与服务器之间传递时不被修改，方法是使数据包与校验和包同时传递。

校验与单独注册——高级安全选项允许将 Oracle 环境与你的结点上可以放置的其他校验或单独注册解决方案集成，Net8 支持 Kerberos、CyberSAFE、Identix TouchNetII、SecurID 的适配器，而且新的 Oracle 安全服务器将为 Oracle 资源提供验证服务，对安全

套接字层 (SSL) 使用数字验证 X.509V3。

与DCE环境集成——使用高级安全选项，可以将 Oracle环境与资源同 OSF的分布式计算环境 (DCE) 集成。

远程拨号用户校验服务 (RADIUS) 协议——这允许ASO使用所有遵守RADIUS的设备。

安全套接字层 (SSL) ——SSL是因特网的标准之一，现在极易与 ASO集成。

Oracle工具包——帮助管理公共键基础结构 (PKI)。

配置第三方验证适配器与 DCE适配器已超出了本书的范围。如果要获取这些项目的信息，参考《Oracle Advanced Security Option Administrator's Guide》与你试图安装的适配器的特定文档。

注意 有两个Oracle高级安全选项的版本：一个版本用于本土使用（美国与加拿大），可以使用现在能用的加密的最高级别。另一个版本适用于其他国家，称为出口使用版本。出口版本必须使用加密的最低级别，这是法律规定的。询问Oracle销售商以决定你的结点可以使用哪种相应的软件。

34.5 启用数据加密与校验

ASO使用RSA数据安全的RC4或数据加密标准 (DES) 加密数据。一个为Net8会话随机产生的键值提供了所有网络通信的安全性，这种加密适用于所有的 Net8网络，包括网关。

为启用数据流加密或数据检验和，必须在客户端与服务器的 sqlnet.ora文件中设置一些参数。在客户端与服务器端各有一套参数。注意如果数据库服务器也作为客户端使用，必须同时设置客户端与服务器端参数。可以按如下方式配置 sqlnet.ora文件：使用文本编辑器编辑这个文件，或使用网络管理或NET助手编辑缺省的资源文件。

SQLNET.ENCRYPTION_SERVER与SQLNET.ENCRYPTION_CLIENT参数指明一个连接是否需要加密，服务器与客户端的值一起进行评估以确定会话的配置，合法的值为：

ACCEPTED——这台机器不加密这个会话，除非其他的机器要求对它加密。

REJECTED——这个机器不加密，即使其他端点要求加密。如果其他的机器 REQUIRES一个加密会话，这两台机器将不能连接。

REQUESTED——这台机器试图加密这个会话，但如果其他的机器不加密这台机器仍可连接。

REQUIRED——这台机器只接受一个加密连接。

SQLNET.ENCRYPTION_TYPES_SERVER与SQLNET.ENCRYPTION_TYPES_CLIENT参数指明客户端与服务器机器可以使用的加密运算法则。如果指定了多于一条运算法则，机器将逐一测试，从第一个开始，到最后一个。在会话中使用的真正的运算法则是由客户端与服务器协商决定的。如果由于服务器与客户端没有一个共同的算法不能协商一条运算法则，连接失败。

合法的加密类型是：

RC4_40~RSA RC4 (40位键值大小) 本土的与国际的

RC4_56~RSA RC4 (56位键值大小) 只有本土的

RC4_128~RSA RC4 (128位键值大小) 只有本土的

DES~标准DES (56位键值大小) 只有本土的

DES40~DES40 (40位键值大小) 本土的与国际的

3DES（使用SSL的三倍DES）

为指定校验和行为，使用SQLNET.CRYPTO_CHECKSUM_SERVER与SQLNET.CRYPTO_CHECKSUM_CLIENT参数。与加密参数相似，这些参数接受ACCEPTED、REJECTED、REQUESTED与REQUIRED作为合法值，当协商连接时以相同的方式运转。

最后一套参数是：SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER与SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT，这些参数指明了产生校验和值的运算法则的类型。当前这些参数只支持MD5作为合法值。

最后，SQLNET.CRYPTO_SEED参数在客户端计算机上配置以种下用密码写的键值，这是一个从10个到70个字符长的数字值。这个系列数越长越随机，校验和键值越坚固。当使用加密或校验和时，必须为这个参数指定一个值。

34.6 ASO支持的RADIUS协议设备

支持RADIUS协议意味着更多的授权厂商可以与Oracle共同工作，给予了实现安全计划更多的机会。Oracle服务器作为一个RADIUS客户并有权连接到RADIUS服务器（参见图34-7）。客户使用ASO注册到Oracle8i服务器，数据库请求RADIUS服务器对该请求的验证，RADIUS服务器接受或者拒绝这个请求，该回答送给Oracle8i服务器，然后它执行适当的行为。Oracle8i服务器真正执行了透明的代理验证。ASO可以提供客户化的Java类，所以可以定制你的安全政策。RADIUS选项可以在因特网机制中实现，方法是使用一个作为RADIUS客

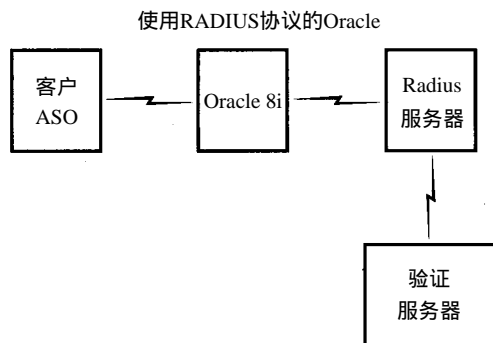


图34-7 使用RADIUS协议的Oracle

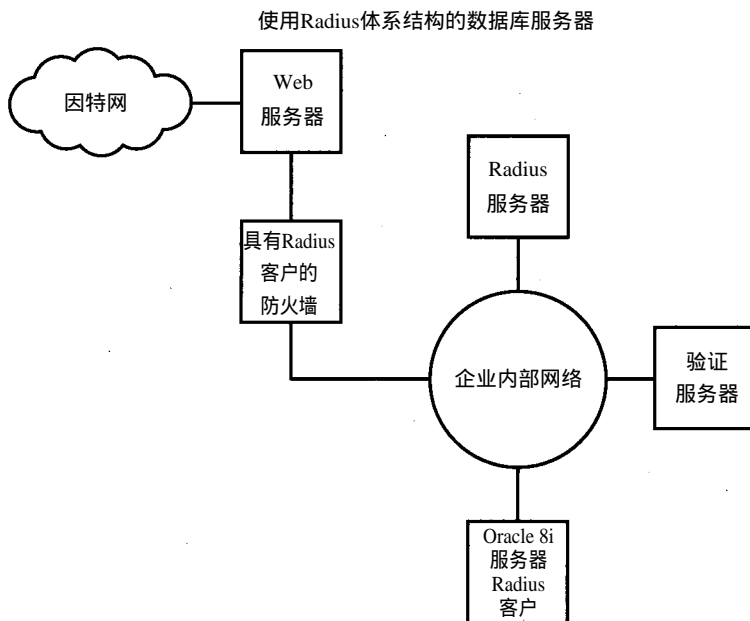


图34-8 使用RADIUS体系结构的数据库服务器

户端的防火墙（参见图 34-8）。验证服务器可以是任何RADIUS协议意见服务器，如ActivCard、SecureId、kerberos、Biometrics等等。

ASO对RADIUS的高级特性包括质询—响应验证与记帐，并包含以下步骤：

- 1) RADIUS服务器向应用服务器质询。
- 2) 质询传送到客户端。
- 3) 质询呈现给最终用户。
- 4) 用户然后对质询提供一个响应。
- 5) 响应传送到RADIUS服务器。
- 6) RADIUS服务器然后证实这个响应并发出拒绝或接受最终用户的消息。

RADIUS记帐提供了创建审计报告的能力，审计报告可以用于安全性及资源的使用情况。

34.7 安全套接字层协议

安全套接字层（SSL）支持已合并入Oracle 8i数据库与ASO。SSL是因特网标准之一，现在更易于ASO集成。SSL提供了使用X.509数字证明的验证并加密网络通信。当一个因特网会话的所有组件都使用SSL时，就拥有了一个产业强度的安全解决方案。必要条件是因特网会话的所有层都支持SSL，包括客户层、Web服务器层与数据库层。SSL与Net8连接共同工作，并使用IIOP与企业级JavaBeans（EJB）。确保数据完整性由密码组处理。客户与服务器都有一套密码组，它们互相商议传输时使用哪个组，参见显示SSL体系结构的图34-9。

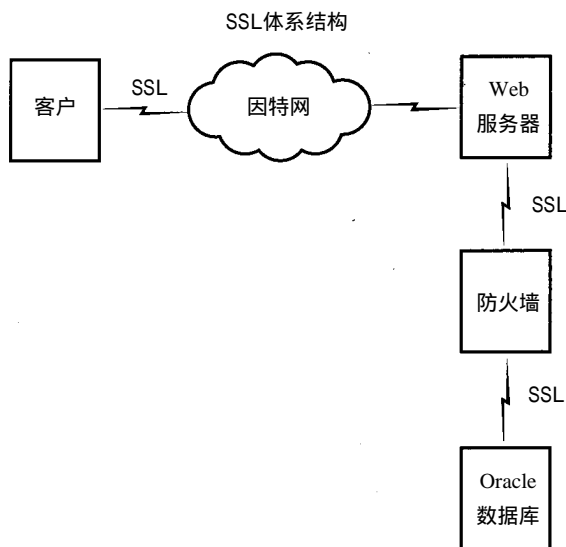


图34-9 SSL体系结构

34.8 Oracle工具包的支持

支持Oracle工具包帮助管理公共键基础结构（PKI）。Oracle工具包管理器允许用户与系统管理员管理包的内容。一个包可以含有用户证明及信任点数（证明权限的可信度）。当配置时，包可以放置在一个由口令保护的因特网路径下，这个口令可以不使用计算资源的方式与用户通信，这完成了只允许用户知道口令的安全方法。

一个工具包可以配置为具有对公司的普通信任点数，并且可以使用集中管理计划进行管理，使用工具包的过程如下：

- 1) 安装Oracle工具包。
- 2) 要求工具包的口令。
- 3) 口令用于创建一个公有键与一个私有键，私有键存储在工具包中。
- 4) Oracle包管理器请求证明（可以从Oracle或非Oracle CA产生）。
- 5) 证明下载并存储在工具包中。
- 6) SSL连接初始化。
- 7) 使用SSL检索用户的身份，服务器验证这个身份。

34.9 理解多线程服务器

缺省情况下，用户通过使用专有服务器进程连接到 Oracle数据库服务器。这意味着对于每个用户连接，一个关联进程替这个用户进程处理工作，如从数据文件将需要的数据载入到数据块缓冲区，并将数据库的查询结果返回给用户。这是提供连接到数据库的最快与最简单的方法。然而，在上百甚至上千个用户同时连接的情况下，维护这些专有服务器进程的开销代价过高，而且专有服务器进程消耗数据库服务器上相同量的资源，不管它们是活动的还是空闲的。在多数情况下，如果有大量用户连接到数据库但事实上很少从数据库存取数据，维护这些专有服务器进程占用的服务器资源很浪费。多线程服务器（MTS）可以节约这些开销。

以最简单的术语说，多线程服务器允许许多用户会话共享一组服务器进程，因而减少了支持一个大量用户所必要的资源开销。这个结构也允许降低这些服务器会话的全部空闲时间。例如，如果有100个同步用户连接，但平均在每一时刻每10个用户连接只有一个是活动的，可以通过分配10个服务器进程给用户使用来使资源极大化，这使10个服务器进程在任何时间都是活动的，而不是当使用专有服务器进程时，10个进程是活动的，而90个进程处于空闲状态。

34.9.1 多线程服务器结构

当使用MTS时，需要了解在体系结构中有一些不同之处。回想第6章当使用专有服务器进程连接到Oracle数据库时，监听器使用专有服务器进程连接到用户会话，管理用户对Oracle数据库的连接。在MTS环境下，监听器的行为稍有不同。不是分散并连接用户会话到一个专有服务器进程，它将用户连接传送给一个或多个调度器进程，这些调度器进程负责将用户进程命令放入请求队列，并从请求队列检索用户进程命令的结果，请示和响应队列都保留在SGA中。

共享服务器进程不直接与调度器或服务器进程交互，而是监控请求队列。当一个新的命令放入队列后，他们读这个命令，以下面的方式处理它：读数据块缓冲区中的适当的数据块并向数据库提交命令，然后将结果放入调度器响应队列中。所有的调度器将它们的请求放入同一个请求队列，每个调度器也有它自己的响应队列。共享服务器进程确保用户命令的结果为执行命令的调度器放入正确的响应队列中。通过这种方法，共享服务器进程可以非常有效地一起工作，处理所有的用户会话请求，调度器只需要检索并操作它们所维护的用户会话的数据。

多线程服务器的使用也改变了对SGA内存区的分配。因为没有专用服务器进程，用户的

会话数据与游标状态信息存储在 SGA 中,而不是 PGA 中。由于这个原因,SGA 应该相应地进行调整。注意这并不是运行 MTS 的附加代价,而是调整在内存的什么地方存放数据。

34.9.2 配置多线程服务器

多线程服务器主要通过每个数据库的 init.ora 文件中的参数配置,以下是配置使能 MTS 的参数。

MTS_SERVICE 与调度器进程相关的调度器进程服务的名字。监听器将对一个服务的请求传送到相应的基于这个参数值的调度器。通常将这个值设为数据库的名字 (db_name init.ora 参数的值)。

MTS_SERVERS 指定当实例启动时共享服务器进程的数量。

MTS_MAX_SERVERS 指定一次运行的共享服务器进程的最大数量。共享服务器进程按需要被分配和消灭,但它们的数量不能超过这个数,或低于 MTS_SERVERS 的值。

MTS_DISPATCHERS 在实例启动时定义协议与分配的调度器的数量。指定使用不同协议的多个调度器,需要分别使用 MTS_DISPATCHERS 指明每个协议。

MTS_MAX_DISPATCHERS 指定一次运行的调度器最大数量。调度器进程基于系统负载被分配和消灭。

MTS_LISTENER_ADDRESS 调度器进程监听的地址。与监听器地址相同。

一个示例 init.ora 文件的 MTS 参数如清单 34-1 所示。

清单34-1 INIT.ORA

```

MTS_SERVICE = PROD01          # Database name is PROD01.
MTS_SERVERS = 3                # Start 3 shared server processes at
instance start.
MTS_MAX_SERVERS = 10           # Never start more than 10 shared server processes.
MTS_DISPATCHERS = "tcp,3"      # Start 3 dispatchers that use the TCP/IP protocol.
MTS_DISPATCHERS = "spx,1"      # Start 1 dispatcher that uses SPX.
MTS_MAX_DISPATCHERS = 10       # Never start more than 10 dispatcher processes.
MTS_LISTENER_ADDRESS = "(address=(protocol=tcp)(address=dbserver)(port=1521))"
MTS_LISTENER_ADDRESS = "(address=(protocol=spx)(service=novellserver))"

```

除配置调度器与共享服务器进程之外,也可以控制客户端的 MTS 行为。因为某些工作不能使用共享服务器进程运行(如直接加载卸出与 SQL*Loader 执行),有些工具使用专有服务器进程运行得更好(如批量或过程化处理密集工作),需要强制使用专有服务器。可以通过将 sqlnet.ora 参数 USE_DEDICATED_SERVER 设为 TRUE,在客户端全局使用专有服务器进程。为一个独立的 TNS 别名指定使用一个专有服务器,将 tnsnames.ora 文件的 SERVER 参数设为 DEDICATED。例如,清单 34-2 指定了两个 TNS 别名连接到同一个数据库。然而,第二个别名使用专有服务器,而第一个使用共享服务器。

清单34-2 连接到相同数据库的两个 TNS 别名

```

PROD_MTS =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = tcp.world)
      (PROTOCOL = tcp)
      (HOST = dbprod)
      (PORT = 1521)

```

```

    )
  )
  (CONNECT_DATA = (SID = PROD01)
)
)

PROD_BATCH =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = tcp.world)
      (PROTOCOL = tcp)
      (HOST = dbprod)
      (PORT = 1521)
      (SERVER = DEDICATED)
    )
  )
  (CONNECT_DATA = (SID = PROD01)
)
)
)

```

34.9.3 管理多线程服务器

在通常的操作中，Oracle服务器按需要启动和关闭共享服务器与调度器进程。这基于进程上所加的负载。然而，可能因为性能问题有必要手工管理 MTS操作或监控。由于这个原因，有一些可用的数据字典视图展示与MTS工作有关的信息。V\$DISPATCHERS与V\$SHARED_SERVERS显示了调度器与共享服务器进程运行时的信息。V\$MTS显示了所有的统计数字，如MTS连接的最大数量、服务器启动的数量、服务器杀死的数量及服务器的最高水位标记；V\$QUEUE显示请求队列与响应队列的信息；最后 V\$CIRCUIT显示使用调度器或服务器的用户连接信息。这些性能视图允许检查你的 MTS配置，必要时进行调整或改进。

可以使用适当的 ALTER SYSTEM命令在实例运行时调整共享服务器或调度器进程的数量。可以使用适当的命令增加或减少进程的数量。注意如果试图中断服务器进程或调度器，Oracle服务器只有在使用它们的用户会话关闭后才会中断它们。

下例显示这些命令是怎样使用的：

```
ALTER SYSTEM SET MTS_DISPATCHERS = 'spx,10';
```

它在运行10个SPX调度器时启动或停止SPX调度器

```
ALTER SYSTEM SET MTS_SERVERS = 5;
```

它在运行5个共享服务器进程时启动或停止共享服务器进程。

34.10 使用Oracle连接管理器

Oracle连接管理器是一个新的 Net8选项，由Oracle8的企业版提供。它提供了以下方面的增强服务：处理连接池、存取控制及多协议支持。在具有成百上千同时连接的环境中它是完美的，并提供了与多线程服务器相同的益处。

使用网络管理器或Net8助手配置连接管理器。还可以通过直接编辑 cman.ora文件配置连接管理器。

34.10.1 配置连接多路技术

多路技术连接（或集中化连接，如 Oracle文档中所提到的）是这样一种进程：在一个单一

的网络连接上路由多个离散的连接。当连接是多路复用的时，服务器只消耗很少的资源，因为多路复用连接只使用一个单一的连接，可以在配置使用 MTS 的环境中使用集中化管理。

在使用动态发现配置的 Oracle 名字环境中，当连接管理器在线时，集中化自动发生。要手工使能集中化，必须将连接管理器正监听的地址与端口放入 tnsnames.ora 文件中。

为了设定连接管理器监听什么，在 cman.ora 中加入下面一行：

```
cman=(address=(protocol=tcp)(host=HOSTNAME)(port=PORT))
```

其中 HOSTNAME 是连接管理器正在运行的计算机的主机名，PORT 是连接管理器应该监听的端口。缺省情况下，连接管理器在端口 1600 监听。

为配置客户端使用连接管理器池，在地址列表中指定连接管理器信息并将 SOURCE_ROUTE 参数设为 yes，如：

```
(description =  
  (address_list =  
    (address = (protocol=tcp)(host=conman)(port=1600))  
    (address = (protocol=tcp)(host=dbserver)(port=1580))  
  )  
  (connect_data=(sid=db1))  
  (source_route = yes))
```

注意 SOURCE_ROUTE 参数指出客户端必须通过多个目的地以到达最终目的地，使用连接管理器表明客户端在游历到真正的数据库之前，必须首先到达连接管理器所在的机器。

34.10.2 配置多协议支持

连接管理器的多协议支持代替了 Oracle7 的多协议交换。为使连接管理器可以处理多个协议，在连接管理器运行的机器上安装所有的协议。连接管理器依赖客户端 tnsnames.ora 文件的配置自动路由查询。源路由地址指明了在 tnsnames.ora 文件中遍历的协议是所有需要使能多协议连接的协议。例如：

```
(description =  
  (address_list =  
    (address = (protocol = spx)(service=conmansrv))  
    (address = (protocol = tcp)(host = dbserver)(port = 1580))  
  )  
  (connect_data = (sid = db))  
  (source_route = yes))
```

34.11 实例学习

一个公司刚刚开始使用因特网，所以他们安装了一台连接到因特网的 Web 服务器，并允许 Web 服务器通过防火墙与数据库服务器通信。这是一个独立的网络，没有连接到企业内部网（参见图 34-10）。这为管理层提出了一些真正的问题，因为数据库没有连接到内部网络，对数据库的远程管理变得十分困难。备份与恢复数据库并不适合他们当前的公司模型，不得不开发附加的硬件与过程。安全小组与数据库小组举行了一个会议，决定将数据库连接到内部网络是否是可行的。安全小组迅速指出如果计算机入侵者危及了内部网络的安全，他们会对公司造成极大的破坏，所以认为风险是相当高的。在进一步审阅图表后，讨论了一系列替代意见。替代意见选择在数据库服务器服务的因特网应用与公司的内部网络之间，需要一个

附加的防火墙，这个防火墙只允许 1521 端口存取内部网络，并且只允许 SQL*Net 通信（参见图 34-11）。数字证明也将使用以验证 SQL*Net 通信是从公司内部的数据库服务器产生的，而不是计算机入侵者产生的。这个解决方案允许 DBA 以有效的方法管理数据库，而不会危及公司的安全。

实例学习

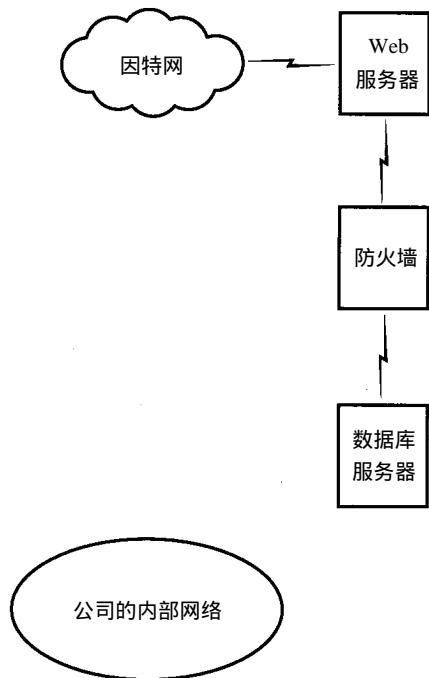


图34-10 实例学习

实例学习——替代实现

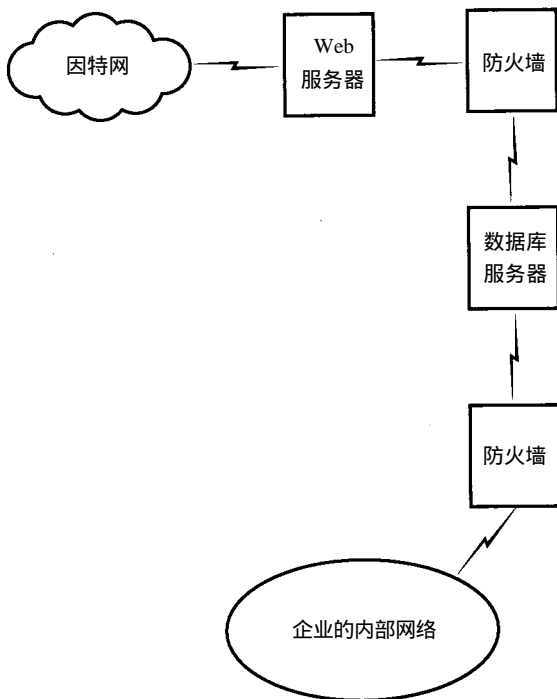


图34-11 实例学习——替代实现

34.12 小结

高级安全选项（ASO）提供了一套非常完整的工具保护你的数据库。安全性总是以一个计划开始，并包括达到这个计划结果的许多检验与平衡。ASO 在与安全性工业标准集成方面的能力是非常强的。因特网改变了每一件事；现在具有计算机入侵者打击公司的计算资源每个核心的可能性，你给了他们一个大门进入你的电子堡垒。商业需求要求你允许因特网存取以简化商务，但基于任何给定因特网系统的风险分析必须成功。ASO 给了我们工具，但现在必须使用它们建立一个安全金字塔保护公司的基础结构。