JDBC 在 B/S 体系中的应用及其安全通信

胡 泳 (上海 同济大学计算中心 200092)

摘要:本文对 JAVA 访问数据库的接口 JDBC 在 BROWSER/SERVER 体系中的应用做了概括性的描述,并结合实例对基于中间件的(第三类)JDBC 的机制及实际应用进行了探讨,最后对 JDBC 的安全通信问题做了详尽的讨论。

关键词:WEB JDBC 安全套接层(SSL) 私有密钥算法 数字签名 签名 JAVA APPLET

随着 INTERNET 的迅速发展,国内外各企业单位都力争将自己的计算机应用系统由传统的 CLIENT/SERVER 模式向 BROWSER/SERVER 模式转变,这就需要利用到 WEB 服务器访问数据库的技术.该技术有多种实现方案,目前流行的方法主要有 CGI, API, PLUG-IN, SERVER-SIDE-SCRIPT, JDBC 等,它们各有特点,本文结合实际应用对 JAVA 数据库接口 - JDBC 在 BROWSER/SERVER 体系中的实现进行分析、论述,并就其安全通信机制及实现进行了探讨。

一、DBC 简介及分类

JDBC 即 JAVA 数据库接口,它是由 SUN 公司为 JAVA 访问数据库而制定的标准及一些 API. JAVA 语言自从其问世以来,便以其安全性,平台无关性深得业界青睐。它的数据库访问接口 JDBC 继承其特色外,并借鉴了 ODBC 的风格,使熟悉 ODBC 的人很容易掌握。

利用 JDBC 访问数据库的原理如下:浏览器从 WEB 服务器上下载 JAVA APPLET 到本地,JAVA APPLET 利用 JDBC 驱动程序直接或者通过中间件(如 IDS 服务器)与数据库建立连接,前者称为两层结构,后者称为三层结构。无论那种情况,JDBC 驱动程序起着主要作用。JDBC API 的创始人 - - - JAVASOFT 公司把 JDBC 驱动程序分为四类:

1. JDBC - ODBC 桥.它通过 ODBC 提供 JDBC 访问,客户端直接与数据库服务器建立连接,属于两层结构方式,由于 ODBC 驱动程序不能下载到客户端,所以需要在客户端安装与服务端相同的 ODBC 驱动程序,适用与企业内部网。

2. native - API partly - Java 驱动程序。它把 JDBC 调用转换成对不同数据库的 API 调用,也属于两层结构方式,需要在客户端安装针对不同数据库的 API 代码。

3. net - protocol all - Java 驱动程序。先将 JDBC 调用转化为一个与数据库无关的网络协议,然后由中间件转换针对不同数据库的调用,属于三层结构方式。此法不需要在客户端安装任何附加代码(称为客户端的"零安装"),中间件服务器可以让客户端连接多种数据库,是一种最灵活的方式。

4. native - protocol all - Java 驱动程序。将 JDBC 调用直接转换成对数据库的 API 调用,属于三层结构方式,此法也不需要在客户端安装任何附加代码,但是一种驱动程序只能对一种特定的数据库访问。

从以上讨论可看出,第三类 JDBC 驱动程序最灵活,因为只需要在服务端安装一个中间件,就可以访问多种数据库;同时对客户端的配制要求最低,只需要装一浏览器就可以访问服务端数据库,实现了客户端的"零安装",特别适用与 INTERNET。下面就对第三类 JDBC 的机制结合实例进行探讨。

二、第三类 JDBC 的实现机制及实例

基于中间件服务器的第三类 JDBC 原理图如图 1:

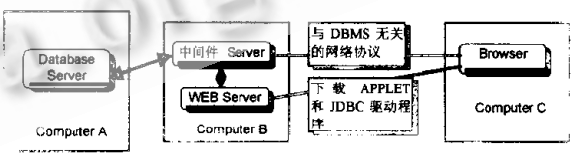

图 1 基于中间件服务器的第三类 JDBC 原理图

浏览器向 WEB 服务器发出 HTTP 请求,WEB 服务器根据 HHTP 请求中的 URL 地址将嵌入 JAVA APPLET 的 HTML 页面连同 JDBC 驱动程序传递到浏览器.JDBC 驱动程序与中间件服务器建立起一个网络连接,JDBC 的调用被转换成一个独立于数据库的网络协议(如 HTTP 协议),然后由中间件服务器转换成对数据库的调用。JAVA APPLET 对数据库的访问就在 JDBC 驱动程序与中间件服务器之间进行,中间件服务器管理着

JDBC 驱动程序的运行。现在我们以 IDS SERVER 为例说明。

IDS SERVER 是典型的中间件服务器，由 IDSSOFTWARE 公司开发，IDS SERVER 运行时使用默认端口号 12，有自己的宿主目录(缺省为 File \ wwwroot \ )，所以可以和标准的 WWW 服务器运行在同一台机器上，不会引起冲突；也可以作为独立的 WEB 服务器，提供标准的 HTTP 服务。它提供了 IDS JDBC 驱动程序，是用 100% 纯 JAVA 写成的，使用的独立于数据库的网络协议是 HTTP，因为 IDS SERVER 是利用 HTTP 工作的。为了使 IDS SERVER 与 WEB 服务器(如 IIS)协同工作，用户开发的 JAVA 类必须和 IDS SERVER 提供的 JDBC 驱动程序一起放在 IDS SERVER 的宿主目录下，而 HTML 页面存放在 WEB 服务器的目录下，同时在 HTML 页面中使用如下调用：< Applet code = "Myclass. class" codebase = "100.0.0.1:12/classes" > </Applet >，其中 Myclass. class 是用户开发的 JAVA CLASS，100.0.0.1 是 WEB SERVER 和 IDS SERVER 所在机器的 IP 地址，12 是 IDS SERVER 的 TCP 端口号，CLASSES 目录下存放 JDBC 驱动程序和用户代码。采用这种方式解决了非信任的 JAVA 类只能同下载它的 WEB 服务器建立网络连接的问题。一旦浏览器同 WEB 服务器建立起网络连接，JAVA APPLET 中的 JDBC 调用被转换成标准的 HTTP 协议，通过 TCP 端口号 12 同 IDS SERVER 进行通信，IDS SERVER 再把它转换成相关的 DBMS 协议利用 ODBC 机制同数据库进行通信。其原理图如图 2：

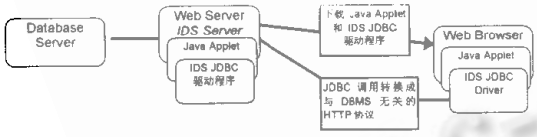

图 2 IDS SERVER 工作原理图

三、JDBC 安全通信的讨论及实现

INTERNET 是完全开放的网络，其上传送的信息也是公开的，有许多途径可以截取、篡改它们，如果这些信息是公司的机密数据而且未经任何加密处理是非常危险的。基于第三类 JDBC 的数据库访问也同样存在安全性问题，因为 JDBC 驱动程序与中间件服务器通信时使用没有安全措施 HTTP 协议，数据在传输过程中很容易被截取，所以有必要考虑它的安全通信。

目前实现网络安全通信的方法很多，一种常用的方法是使用安全套接层(SSL)协议。SSL 协议是由 NETSCAPE 公司提出的并且被广泛认同的安全通信工业标准。SSL 协议可以由公开密钥算法或者私有密钥算法实现。由于私有密钥算法较简单而且效率比较高，广泛的用于数据加密，本文重点讨论利用私有密钥算法实现 SSL 协议的方法。

私有密钥算法的基本思想是通信的两端各持有一个完全相同的密钥，传输方利用密钥加密数据，接收方利用密钥解密数据。该法效率很高，适用于大量信息的情况，缺点是在客户端必须安装密钥及相关信息，而且客户端获取密钥时有被第三者盗取的危险。弥补的方法是：利用公开密钥算法传送私有密钥，然后利用私有密钥算法传送数据。第三类 JDBC 的安全通信利用该算法的实现可以描述为：在客户端和中间件服务器上都装上相同的密钥，传输数据前利用一定的机制加密，到达目的地后解密。由于下载到客户端的 JAVA 类需要利用本地的密钥加密数据，就可能要访问本机的文件系统，然而为了安全，非信任的 JAVA APPLET 是严禁访问本机的文件系统的，所以在服务端需要对 JAVA 类进行“数字签名”，实现方法稍后再作讨论。

在 SSL 协议中，客户端与服务端的安全连接称为一个“会话(session)”，一个会话中包含许多安全连接建立时产生的加密信息，基于公开密钥算法的 SSL 协议中，当一个连接结束时，这些信息将自动消失；基于私有密钥算法的 SSL 协议中，我们称会话为“静态会话”，其中包含了密钥和其他的 SSL 信息。静态会话是以静态会话文件方式存在的，在服务端和客户端各有一个副本。从上面的论述中可以看出，利用 SSL 协议实现 JDBC 驱动程序与中间件服务器安全通信的关键在于密钥的生成及获取和 JAVA APPLET 的“数字签名”，下面以 IDS SERVER 为例来说明具体实现。

1. 密钥的生成

IDS SERVER 利用 SSL 协议实现 JDBC 驱动程序与中间件服务器安全通信的机制称为安全 JDBC(SECURE JDBC)。首先需要为安全通信生成密钥。Session. exe 是一个可以生成静态会话文件命令行命令，每个静态会话文件包含了随机且唯一的私有密钥，调用格式如下：

session < cipher-spec > < session-file > [life-span]

cipher-spec 是生成私有密钥采用的算法，如 DES，DES40，BLOWFISH 等，session-file 是生成的静态会话文件名，以 . SES 结尾，life-span 是会话文件的有效时间。IDS SERVER 需要在 IDSSERVER \ SECURITY \ 目录下运行 SESSION. EXE 命令，而且会话文件都存放在该目录下，因为它从这里读取会话文件。对于 IDS SERVER

来说，可以根据客户的多少生成任意多个会话文件，每个会话文件包含一个不同的密钥和一个会话 ID 号，当 IDS SERVER 启动时，把所有的会话文件中的会话 ID 号都调入内存，根据会话 ID 号来决定对于不同的网络连接使用那一个私有密钥，这样可以保证每个客户有唯一的密钥，增强了安全性．以下是 Session.exe 命令的使用例子：

C:\ IDSServer \ Security > session blowfish kjohnson 180

C:\ IDSServer \ Security > session des40 auditor 7

会话文件的另一个拷贝要安装在客户端上．为了使服务端生成的密钥安全的到达客户端，可以使用公开密钥算法对会话文件加密后再传送。会话文件到达客户端后，为了安全应该把它存放在一个受本地操作系统保护的目录中，然后利用 ClientInstaller class 进行会话文件的安装，ClientInstaller. class 可以在 IDS 服务器中的 IDSServer \ Security 目录下找到．安装过程中最重要的是建立 Java.security 文件，它里面包含了 JDK1.1 提供的 JAVA 加密机制 API 的安全设置，而且 IDS JDBC 驱动程序利用它来寻找客户端的会话文件。

当客户端的 JAVA APPLET 调用 JDBC 驱动程序与 IDS SERVER 建立连接时，它要读取本地会话文件，用该文件的会话 ID 号与 IDS SERVER 内存中的会话 ID 号比较，如果有匹配的，网络连接成功，双方根据会话 ID 号利用会话文件中的密钥进行数据的加密和解密，否则，网络连接建立将失败。

2. JAVA APPLET 的"数字签名"

由于 JAVA APPLET 要读取本机文件，为了克服非信任的 JAVA APPLET 禁止访问本地文件系统的限制，需要对 JAVA APPLET 进行"数字签名"．所谓"数字签名"就是数据发送端对要传送的数据进行某种处理，使接收端通过逆处理后能够确认数据的确来自期望的发送端，如果数据在传送过程中被篡改，接收端可以发现。在实现上，先把要签名的数据（如 JAVA 类，JDBC 驱动程序）生成文档，然后对文档签名。一般有两种文档技术：JAR 和 Cabinet 文件格式，JAR 是 JAVASOFT 公司发明的，受 JDK1.1 的支持；Cabinet 是 MICROSOFT 公司推出的。两者的工作原理都是把许多文件结和成 JAR 或者 Cabinet 文件（称为档案 ACHIVES），而原来的文件都可以从这些档案中恢复。所以对 JAVA APPLET 数字签名就是由相关的 JAVA CLASS 和其他一些资源生成档案，然后对档案签名。

生成档案的工具主要包括 Cabinet SDK 和 jar，前者用来生成 Cabinet 格式的档案，可以从 Microsoft SDK for Java，ActiveX SDK 和 Visual J + + 中得到，后者用于 JAR 格式，可以从 JDK1.1 中得到。下面是两个命令的例子：

D:\ TEMP \ packing > cabarc - s 6144 - r - p n . . \ SecureJDBC. cab * . *

D:\ TEMP \ packing > jar cf . . \ SecureJDBC.jar *

执行后生成 SecureJDBC. cab 和 SecureJDBC. jar 两个档案，具体参数个以查看两命令的帮助．需要注意的是 JDBC 驱动程序必须包含在档案中。

对 Cabine 格式档案签名，需要使用 Code Signing Kit from Microsoft，里面提供了 signcode. exe，使用例子如下：

D:\ TEMP > signcode - spc IDSCert. spc - k IDSKey SecureJDBC. cab

IDSCert. spc 是认证文件，可以从认证部门（CA）获取，也可以利用工具生成临时的测试版本。

对 JAR 格式档案的签名可以用 JDK1.1 提供的 javakey. exe。

至此，一个能够进行安全通信的体系已经建好，为了使 JDBC 驱动程序与 IDS SERVER 间的能利用安全机制（Secure JDBC）进行通信，只要在 Java Class 中的连接 URL 部分使用"SSL = 1"的标记即可：

Jdbc. ids://myserver: 12/conn? dsn = IDSExamples&uid = Username&pwd = password&ssl = 1

而 HTML 页面中 APPLET 的调用如下：

```
<applet code = "MyApplet.class" archive = "my.jar">
    <param name = "cabbase" value = "my.cab">
</applet>
```

因为 NAVAGATOR 与 EXPLORE 只支持各自的文档格式，所以需要使用 * .jar 和 * .cab 两种文档。

四、结束语

JAVA 语言本身固有的优点决定了基于 JDBC 的 WEB 访问数据库方式具有很大的优势，而第三类 JDBC 驱动程序更具有灵活性，已经得到了许多第三方厂商的支持，其应用范围会越来越广泛．安全通信是当今 INTERNET 网上的热门话题，是实现电子商务的关键所在，JDBC 驱动程序与中间件服务器的安全通信也必然越来越成熟，是我们应重点研究的课题。

参考文献

[1] 王克宏，丁锂，孙元．Java 语言 SQL 接口——JDBC 编程技术．北京：清华大学出版社，1997

[2] http://splash.javasoft.com/jdbc/jdbc.drivers.html

[3] http://www.idssoftware.com/

（来稿时间：1998 年 12 月）