用JAAS和JSSE实现Java安全性Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/629/2021_2022__E7_94_A8JAAS_E5_92_8C_c104_629766.htm 从早期所谓的 Java 沙箱到 JDK 1.4 引入的健壮的、全功能的安全体系结构，安全性一直是 Java 平台的基本组件。从那时到现在，Java 语言的设计者收到了来自团体的大量关于安全的 Java 应用程序(或者企业环境)可以做什么和不做什么的意见，他们自己也添加了若干技巧。 可以说随着 J2EE Web 应用程序安全体系结构的引入，我们不断从近 10 年的反复试验有所收获，事实也表明了这一点。J2EE 安全框架由三个 API 组成：Java 认证和授权服务(JAAS)、Java 安全套接字扩展(JSSE)和 Java 加密扩展(Java Cryptography Extension，JCE)。虽然 JCE 是一个有意思和重要的 API，但是它与我们所关注的安全 Web 应用程序开发的“三大项”——认证、授权和传输——并不特别相关。所以在本月的专栏中我们将集中讲述 JAAS 和 JSSE。 JAAS 和 JSSE 概述 JAAS 提供了一种灵活的、说明性的机制，用于对用户进行认证并验证他们访问安全资源的能力。JSSE 定义了通过安全套接字层(SSL)进行安全 Web 通信的一种全 Java 的机制。通过结合这两种技术，可以使我们的应用程序： 验证用户就是他或者她所宣称的那个人(认证)。 保证允许他或者她访问所要求的资源(授权)。 通过安全网络连接进行完整的信息交换(传输)。 现在，我们来看每一个基础的功能组件。 用 JAAS 进行认证 JAAS 建立在一种称为可插入的认证模块(Pluggable Authentication Module，PAM)的安全体系结构之上。PAM 的体系结构是 模块化的，这意味着它设计为可以通过交换模块

，支持从一个安全协议组件无缝地转换到另一个协议组件。这个框架中定义良好的接口使得无需改变或者干扰任何现有的登录服务就可以加入多种认证技术和授权机制。PAM 体系结构可以集成范围广泛的认证技术，包括 RSA、DCE、Kerberos 以及 S/Key，因而 JAAS 也可以集成这些技术。此外，这个框架与基于智能卡的认证系统和 LDAP 认证兼容。就像许多 Java 2 平台技术一样，JAAS API 定义了应用程序代码与将要执行业务逻辑的物理实现之间干净的抽象。这个抽象层不用重新编译现有的应用程序代码就可以作为登录模块的运行时替代。特别是，应用程序写到 LoginContext API，而认证技术提供程序则写到 LoginModule 接口。在运行时，LoginContext 将读取配置文件以确定应使用哪一个(一些)登录模块对访问特定应用程序的用户进行认证。 JAAS 所使用的认证方案以两种非常重要的实体为基础：principal 和 subject。实际被认证的人或者服务称为 subject。 principal是一个惟一的实体，比如个人或者组的名字、帐号、社会安全号或者类似的惟一标识。为了惟一标识一个 subject(这是认证的关键部分)，一个或者多个 principal 必须与这个 subject 相关联。最后，一个 subject 可能拥有安全相关的属性，称为 凭证(credential)。凭证可以是从简单的密码到复杂的加密密钥的任何东西。应用程序通过实例化一个 LoginContext 对象开始认证过程。LoginContext 查询一个配置文件以确定进行认证所使用的一种(或者多种)认证技术以及相应的一个(或者多个) LoginModule 。一个非常简单的 LoginModule 可能会提示输入用户名和密码并对它们进行验证。高级一点的可能会使用现有的操作系统登录身份进行身份验证。理论上，甚至可以将

一个 JAAS LoginModule 构建成与指纹识别器或者虹膜扫描仪交互。 用 JAAS 进行授权 认证只是 Java 安全框架任务的一半。当用户的身份被确认后，必须对他或者她的访问权限进行检查。只有确认了适当的权限后，用户才可以访问安全的系统或者资源。 换一种说法，验证了用户或者服务的身份后，就创建一个 Subject 对象来表示经过验证的实体。然后 JAAS 将这个对象传递给任何为保护对敏感系统或资源的访问而建立的授权组件。 要确定授权，可以向 Java 2 Security Manager 提供 Subject 及其 Principal s，以及 Subject 要执行的特权操作(读/写到文件系统、数据库访问，等等)。Security Manager 会咨询与 Principal s 和权限相关联的策略文件。 如果一个 Subject 的 Principal s 具有执行指定操作的权限，那么就对这个 Subject 授权并允许操作，否则就会拒绝这项操作并抛出一个 SecurityException 。 用 JSSE 进行安全传输 有了 JAAS ，我们就可以识别访问系统的用户并限制他们只能访问授权使用的那部分系统。虽然 JAAS 是迈向安全 Web 应用程序坚实的第一步，但是如果没有安全传输，那么应用程序安全性仍然是不完整的。 这里，我们仍然是以明文形式——即 HTTP 、TCP/IP、FTP等——传递安全信息(包括认证信息)。所以我们需要保证数据在传输时不会被未授权的人访问。我们还需要保证数据在到达之前，没有在传输过程中修改过，不管这种修改是有意的还是无意的。我们可以利用安全套接字层(SSL)和传输层安全性(Transport Layer Security，TLS)协议实现这两种功能。 SSL 和 TLS 不是特定于 Java 的协议，它们是为维护通过套接字的数据的完整性和私密性而设计的网络层协议。Java 安全套接字扩展(JSSE)利用 SSL/TLS 可以进行安全

的 Internet 通信，它提供了一个具有完整功能的应用程序框架——一个 Java 版本的 SSL 和 TLS 协议，这些功能包括数据加密、服务器认证、消息完整性，等等。使用 JSSE，我们可以定义运行任意应用程序协议——包括 HTTP、TCP/IP、FTP，甚至 Telnet——的客户机与服务器之间的安全套接字连接。从数据加密的角度看，JSSE 结合了许多与 JCE 中使用的同样的概念和算法。不过更重要的是，在简单流套接字 API 背后，它会在必要时自动使用它们。 要利用 JSSE API，我们只需要做简单的几件事。首先我们需要获得 JSSE 提供程序(请参阅参考资料)。其次，我们需要从一个 JSSE 套接字工厂而不是直接从 java.net.Socket 类获得套接字。客户端代码从 SSLSocketFactory 获取套接字，而服务器端代码从 SSLServerSocketFactory 获取套接字。通过从这些工厂获取套接字，我们就可以利用 JSSE 提供程序提供的框架，而不是像 java.net 包允许我们所作的那样，简单地创建标准的、不安全的套接字。 有关 JSSE 的更多细节，请参阅 参考资料。 结束语 Java 平台以其岩石般坚固的安全性闻名。每一年 Java 安全性框架都会变得更灵活和更健壮，JAAS 和 JSSE 的加入表明这个传统将会继续发扬光大。 本月，我们快速回顾了保证 Java Web 应用程序安全的技术。JAAS 提供了对用户进行认证和控制访问资源的模块化机制。JSSE 提供了 SSL 和 TSL 协议的 Java 实现以支持数据完整性和私密性。下个月，我们将探讨 Servlet 过滤器的全新世界。那时再见，祝您探索快乐! 编辑特别推荐: 指点一下：到底该不该去考JAVA认证? Java认证权威问答精华集 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com