DOI:10.3969/j.issn.1003-0972.2009.02.037

基于 JAAS 的企业身份认证平台的研究与设计

沈桂兰[1*],李玉霞[1],孙印杰[2]

(1. 北京联合大学 商务学院 电子商务系,北京 100025;2. 河南师范大学 计算机与信息技术学院,河南 新乡 453007)

摘　要:采用 JAAS 可插拔认证框架技术,研究设计了适用于企业环境的身份认证平台.该平台以改进的动态口令身份认证模块为基础认证模块,并结合 JAAS 的工作流程,具有较高的认证强度,安全性较好,同时也可以根据需要方便地选择认证方式,并具有较强的可扩展性.

关键词:JAAS;身份认证;动态口令;可扩展性

中图分类号:TP302.1;TP309.2　　**文献标志码**:A　　**文章编号**:1003-0972 (2009)02-0296-04

The Research and Design of the Enterprise Identity Authentication Platform Based on JAAS

SHEN Gui-lan[1*], LI Yu-xia[1], SUN Yin-jie[2]

(1. Department of Electronic Business, Business College of Beijing Union University, Beijing 100025, China;
2. School of Computer and Information Technology, Henan Normal University, Xinxiang 453007, China)

Abstract: An identity authentication platform fitted for enterprise was studied and designed based on JAAS technology. Based on improved dynamic password technique, a basic identity authentication module was developed by combining the JAAS technology. EISSP based on JAAS not only has high security, but also has strong scalability in that it facilitates the choice of authentication methods.

Key words: JAAS; identity authentication; dynamic password; scalability

在企业环境中,对于特定的系统资源,应该只有经过授权的合法用户才能访问.如何正确地鉴别用户的真实身份是企业信息安全的关键,是众多企业解决网络安全问题时遇到的首要问题.用户身份认证,也称用户身份鉴别或验证,是用户向计算机系统以一种安全的方式提交自己的身份证明,由系统确认用户的身份是否属实,最终拒绝或赋予用户一定权限的过程.目前应用在企业环境中的身份认证实现技术多种多样,如静态口令、动态口令、智能卡和生物特征、X.509 数字证书[1]等.每种实现技术各有特点,随强度等级不同,适用于不同安全级别的应用系统.同时,各个系统通常是独立地维护自身的认证授权机制,且管理的方式各不相同,随着系统数量的不断增加,用户认证授权的正确性将难以保证,其维护成本以及维护难度也将会随之日益增加.在企业中开发出一种统一的身份平台,可以根据应用系统的安全等级方便地选择替换不同强度的认证方式,而不需要对应用系统做过多的修改,这就显得十分重要.

本文利用 JAAS 技术,研究设计出具有可插拔模块的 ESSIP 企业身份认证平台,采用动态口令技术实现基本的认证模块,同时用户可以根据应用系统的认证需要,方便地选择不同的身份认证方式,其目的是能够提高企业中应用系统的身份认证效率,降低成本

1　JAAS 技术

JAAS (Java Authentication and Authorization Service)是 J2EE 中关于安全的规范,它提供了系统中实体认证与访问控制的安全机制[2].JAAS 包括认证和授权两方面的内容,是 PAM 框架的 Java 实现,在应用程序和底层的验证和授权机制之间加

收稿日期:2008-10-11;**修订日期** 2009-01-29;*.**通讯联系人**, E-mail:guilan.shen@bcbuu.edu.cn

基金项目:2007 年度北京市拔尖人才资助项目

作者简介:沈桂兰 (1979-),女,河南信阳人,讲师,硕士,主要从事电子商务、信息安全研究.

入一个抽象层,其优点是抽象层独立于平台的特性使开发人员可以使用各种不同的安全机制,而且不用修改应用程序级的代码. 和其他 Java Security API 相似,JAAS 通过一个可扩展的框架——服务提供者接口 SPI (Service Provider Interface)来保证程序独立于安全机制. 服务提供者接口是由一组抽象类和接口组成的.

在 JAAS 框架中,应用程序同登录上下文 LoginContext 对象进行交互,LoginContext 对象管理一个或多个 LoginModule 对象,一个 LoginModule 对象对应一种具体的认证方式. 通过加入或更换一个 LoginModule 来增加或更换对某种认证机制的支持. 这种变换是动态的,在运行时通过配置文件来指定,这就是 JAAS 的"可插拔"特性[3].

在应用程序中使用 JAAS 验证中涉及到的核心类和接口大多数都在 javax. security. auth 包和 com. sun. security. auth 包中,主要包括:

公共类:Subject,Principal,Credential

Subject 类代表了一个验证实体,它可以是用户、管理员、Web 服务,设备或者其他的过程. 一个 Subject 对象可以关联一个或多个相关身份,每个身份由一个 Principal 对象表示. 一个 Subject 还可以包含私有、公有两组凭证,用 Credential 表示.

Principal 对象代表了 Subject 对象的身份,表示具有访问权限的一个实体,不具有持久性. 每次使用者登录时必须将它们增加到 Subject 中. 它们实现了 java. security. Principal 和 java. io. Serializable 接口.

Credential 类表示一种凭证,分为私有和公有的凭证,可以是任何对象. 用于将特定安全系统需要的验证信息,如票据,密钥或口令等和 Subject 联系起来.

认证类: LoginContext,LoginModule,CallBackHandler,Callback.

LoginContext 登录上下文类使用一个配置文件来确定使用哪个 LoginModule 对用户进行认证. 配置文件可以通过系统属性 java. security. auth. login. config 指定.

LoginModule 对象对应一种具体的认证方式. 可以通过加入或更换一个 LoginModule 来增加或更换对某种认证机制的支持.

CallBackHandler 是回调处理类,被 LoginModule 类用来和用户通信以便取得认证信息.

Callback 是回调类,用于 CallBackHandler 中保存用户的通信信息.

2 企业身份认证平台的设计

2.1 总体架构

综合考虑部署的灵活性和集中管理的高效性,设计的身份认证平台总体架构如图 1 所示. 认证平台主要由中心认证服务器 AuthCenter 和应用服务器代理 SAgent 两部分组成,AuthCenter 的作用是为用户提供可信的第三方认证,包含的主要模块有:身份认证模块、访问控制模块、认证代理和统一数据中心,包括用户信息数据库和授权策略数据库. SAgent 负责处理用户请求.

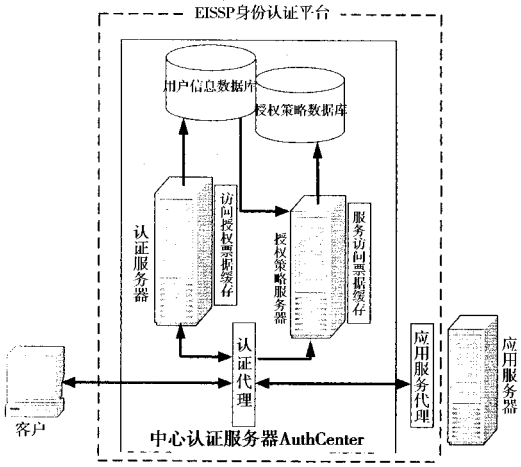

图 1 EISSP 身份认证平台系统结构图

Fig. 1 The architecture of EISSP identity platform

2.2 企业身份认证平台的各部分功能

2.2.1 应用服务器代理的功能

应用服务器代理 SAgent 的主要作用是截获用户访问请求,检查访问请求中的信息,以决定用户是否能够访问应用服务器. 与中心认证服务器 AuthCenter 中的认证代理建立安全通道,把用户信息和资源访问请求送到中心认证服务器的认证代理.

2.2.2 中心认证服务器的功能

中心认证服务器 AuthCenter 各部分的主要功能如下:

(1)身份认证模块

身份认证模块是 EISSP 身份认证平台的核心模块,它主要完成对用户身份的进行认证,并为通过认证的用户发放一个用户凭证,这两部分功能分别由身份鉴别机制和凭证获得机制来实现.

发放的用户凭证是访问授权凭证 TGT (Ticket Granting Ticket),具有时效性,获得 TGT 的用户可以在时效范围内访问平台中的访问控制模块.

身份认证模块中建立缓存机制,用于缓存发放

给用户访问授权凭证的详细信息.

(2)授权策略模块

EISSP 身份认证平台采用基于角色的访问控制技术将身份认证与访问授权分离,为应用服务器上的每一个应用资源建立一个访问控制列表 ACL (Access Control List),这种联系记录在统一用户中心的授权策略数据库当中.用户凭借通过身份认证后获得的访问授权凭证可直接查询授权策略数据库,来决定该用户对所请求的资源是否具有访问权限[3].如果有,授权策略模块发放给用户用于访问应用资源的服务访问票据 ST (Service Ticket).否则,返回给用户一个拒绝访问请求的信息.

授权策略模块中同样建立缓存机制,用于缓存发放给用户的服务访问票据的详细信息.

(3)认证代理 AuthAgent

认证代理接收并鉴别应用服务器代理发送来的用户信息和访问请求,做出如下处理:对于未获得用户凭证的访问请求,发送用户信息到认证模块进行身份验证;对于已获得用户凭证的请求,依据用户凭证,查询授权策略数据库,决定用户对资源的访问权限.

(4)数据中心 Database Center

统一的数据中心由用户认证数据库和授权策略数据库组成.用户认证数据库记录了发放给用户的口令令牌的信息、用户的身份信息、用户组的信息、角色信息等等,是用户通过认证和发放认证凭证的依据.授权策略数据库记录了注册过的受保护资源的各项信息,是认证代理控制用户访问资源时的依据.

3　身份认证平台设计实现

3.1　动态口令身份认证模块

目前一些著名的动态口令安全产品,如 RSA 的 SecureID[3],Vasco 的 DigiPass[4] 系列采用的都是基于时间同步的动态口令生成技术,基本实现原理是:合法用户持有的动态口令令牌中置有和认证服务器同步的内部时钟,另外令牌还内置有密钥和动态口令生成算法.动态口令是以用户登录的当前时间作为变动因子进行运算得到的.用户进行身份认证时,同时输入静态口令和动态口令,客户端把结合后的口令传送到认证服务器,认证服务器确认静态口令的合法性后从加密的数据库中提取该用户所对应的密钥,采用和令牌中相同的算法计算出验证口令,若验证口令和动态口令相同或在设置的时间窗范围则通过验证.

这种方法在令牌内置时钟和认证服务器时钟同步的情况下能做出正确的判断.但是,一旦发生了时钟偏移,需进行时钟校正的方法相当繁琐.EISSP 平台的身份认证模块设计实现的基于时间同步的动态口令算法模块 DyLoginModule,采用了 Triple-DES 对称加密算法作为动态口令的生成算法,在令牌中以时间为因子进行加密生成动态口令,而在认证服务器中解密动态口令提取时间因子,将提取的时间和认证服务器的内部时钟进行比较,从而进行用户的身份认证.认证原理如图 2 所示.算法模块包括密钥产生模块,口令生成和验证模块[5].

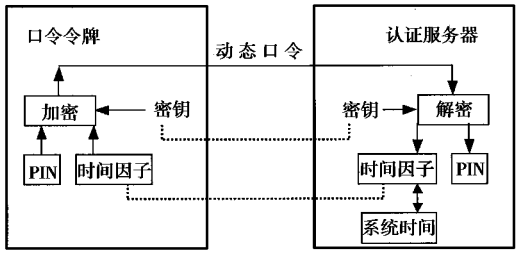

图 2　时间同步动态口令认证原理

Fig. 2 Authentication principle of the dynamic password based on time synchronization

DyLoginModule 模块中,进行比较验证的不是最终生成的动态口令,而是口令令牌中的时间和当前系统的时间,在认证服务器中只需经过一次解密就可以提取动态口令中的时间因子,时间窗大小设为,当满足时,用户即可通过认证,同时在认证服务器中记录下该用户用对应的正负时间偏差,以便下次认证时作同步校正.这种方法避免了为进行多次验证口令生成与比较的繁琐,简化了认证和同步校正的过程,另外当出现不同的情况时,只需要调整时间窗的大小即可.

3.2　工作流程

EISSP 平台的 JAAS 认证的工作流程如图 3 所示,用户提交代表自己身份的凭证,JAAS 创建登录下上文 LoginContext,LoginContext 读取配置文件以确定应该使用哪一个或哪一些登录模块对访问特定应用程序的用户进行认证.在登录时,LoginContext 对象调用每个 LoginModule 对象的 login()方法,实现拦截验证,每个 login()方法进行验证操作活动,获得一个 CallbackHandler 对象,CallbackHandler 对象通过使用一个或多个 Callback 方法同用户进行交互,获得用户输入.验证用户凭证成功后,创建 Session 会话对象,方便用户后继的访问请

求得到许可.

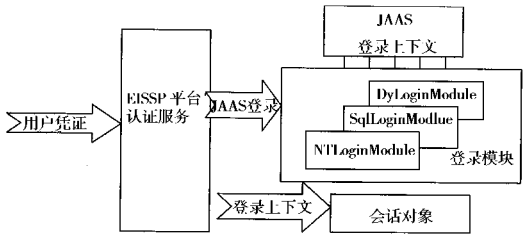

图 3　EISSP 平台 JAAS 认证工作流图

Fig. 3 JAAS authentication flow in EISSP platform

3.3　实施步骤

实现 JAAS 验证要进行以下工作.

第一步:设计登录配置文件. 在登录配置文件 EisspLogin. config 中加入申请:

```
AuthApp1{
cn. eissp. auth. login. DyLoginModule requried
driver = " com. microsoft. jdbc. sqlserver. SQLServerDriver"
url = " jdbc: microsoft: sqlserver://localhost: 1433;DatabaseName = EISSP" ," sa" ," " "
debug = " true"
}
```

第二步:EISSP 中心认证服务器的身份验证模块 cn. eissp. auth. AuthModuleHandler 包含了 JAAS 认证的整体部分,在方法 authenticate (ServletRequest request, String username, String password) 中创建登录上下文 LoginContext,传递一个执行认证对象的类名,和一个 CallbackHandler 对象. 这里需要用到前面的配置文件和后面实现的类.

第三步:实现登录模块的类和反馈处理类. 实现配置文件中的 DyLoginModule 需要实现 initialize ()、login ()、logout ()、commit ()、abort ()等方法.

在 DyLoginModule 类的 login ()方法中调用 CallbackHander 类,用于处理各个反馈类与用户交互. 进行以下的操作:

(a)创建两个 Callback 对象. 这些对象从用户输入中获取用户名/动态口令. 程序中我们使用了 JAAS 中的两个 Callback 类:NameCallback 和 PasswordCallback.

(b)通过将 callbacks 作为参数传递给 CallbackHandler 的 handle ()方法来激活 Callback.

(c)通过 Callback 对象获得用户名/动态口令.

(d)在 dyValidate ()方法调用动态口令 DyLoginModule 模块中的算法,根据用户的 pin 码,提取加密数据库用户对应的 Tripe-DES 密钥,对获得的用户名动态口令进行解密操作,提取时间因子,并考虑时钟偏移因素和当前系统时间进行比较,以确定用户的合法性.

编写反馈处理类 EisspCallbackHandler 实现了 CallbackHander 接口.

3.4　其他认证模块

在 ESSIP 身份认证平台中,除了采用 DyLoginModule 以外,在实际应用中可以根据企业中被保护资源的安全级别,采用其他认证模块,在更换认证模块时只要修改配置文件 EisspLogin. conf 即可. 管理员根据需要,既可以采用自己编写登录模块,也可以采用 JAAS 提供的登录模块. 如,用于读取 NT 当前登录用户的身份标志信息的 NTLoginModule,采用密钥库的验证机制的 KeyStoreLoginModule,使用 Kerberos 协议验证用户的 Krb5LoginModule 等.

4　结论

本文采用 JAAS 认证框架开发 EISSP 身份认证平台,可以方便用户根据需要选择更换合适的身份认证方式. 首先,用户能很方便地开发出应用于本系统的身份认证模块,现有的应用服务几乎不用作任何的修改,避免了旧系统的改造;第二,用户只需要更改配置文件,就可以方便地更换现有的认证方式或增加新的身份认证方式. 该认证平台具有良好的可扩展性以及良好的应用价值,目前已应用到电子商务平台的开发和设计中.

参考文献:

[1] 李　辉. 计算机安全学[M]. 北京: 机械工业出版社, 2005:106-107.
[2] 徐　甜. Java 平台及应用 Java 技术的安全问题研究[J]. 微计算机信息,2007 (18): 216-218.
[3] 王　昕. 基于 JAAS 的安全认证技术在 Web 应用系统中的实现[J]. 北京建筑工程学院学报, 2008,24 (2):55-58.
[4] Rsa S. *RSA SecurID Authenticators* [EB/OL]. (2004-11-10)[2008-02-22]. http://www. rsasecurity. com.
[5] 沈桂兰,李　辉. 一种改进的动态口令技术[J]. 福建电脑, 2005 (11): 269-270.

责任编辑:任长江