文章编号：2096-1472(2018)-07-32-04
DOI:10.19644/j.cnki.issn2096-1472.2018.07.009

基于EJBCA的CA证书认证中心的搭建与应用

史建宜[1]，陈新鹏[2]
(1.国网北京电力公司房山供电公司电力调度控制中心，北京 102401；
2.国家电网公司信息通信分公司信息通信调度控制中心，北京 100053)

摘　要：保障信息安全交换与传输越来越重要。CA认证中心，作为一种可信赖的第三方证书机构，为信息安全传输提供安全的密码认证服务，保证信息传输参与者的身份真实有效、信息在网络上安全传输。EJBCA是一个开源的CA系统，首先介绍了EJBCA特点和基本结构；然后详述通过EJBCA搭建与配置CA认证中心；随后讨论使用EJBCA进行证书的申请和生成；最后，通过一个简单实例说明EJBCA的应用。

关键词：EJBCA；信息安全；认证中心
中图分类号：TP393.08　**文献标识码**：A

Certificate Authority Construction and Application Based on EJBCA

SHI Jianyi[1],CHEN Xinpeng[2]
(1.*Dispatching and Control Center,Beijing Fangshan Power Supply Company,Beijing* 102401,*China*;
2.*Dispatching and Control Center,State Grid Information & Telecommunication Brach,Beijing* 100053,*China*)

Abstract: Secure information exchange and transmission is increasingly important.As a trusted third-party certificate authority,CA (Certificate Authority) provides secure password authentication services for information security transmission. It ensures that the identity of the information transmission participant is valid and effective,and the information is transmitted securely over the network.EJBCA is an open source CA.This paper introduces EJBCA,outlines the characteristics and the basic structure,details the setup configuration of certificate authority through EJBCA,discusses subsequently application and generation of certificates using EJBCA and finally explains the simple application of EJBCA by an example.

Keywords:EJBCA;information security;Certificate Authority

1 引言(Introduction)

网络的迅速发展给整个社会带来了各种各样的改变，人们通过网络进行信息交换变得越来越普遍。正因如此，人们越来越意识到信息安全交换和传输的重要性。现阶段，人们对信息安全传输有如下几个要求：身份鉴别、数据的保密性、数据的完整性、不可抵赖性。信息安全传输也面临着一系列的问题和挑战：如何保障信息传输的可靠性？如何鉴别信息来源的真实性？如何确保信息的完整性和不可抵赖性。诸如以上的问题，推动了信息安全传输技术的快速发展。

PKI是Public Key Infrastructure的缩写，是指用公开密钥的概念和技术来实施和提供安全服务的具有普通性的安全基础设施，它是国际上解决开放式互联网络信息安全的一套体系[1]。PKI主要由证书认证中心(CA)、审核注册中心(RA)、证书查询验证服务器(LDAP)等组成[2]。PKI体系支持身份认证、信息传输、信息的完整性、信息的机密性，以及操作的不可否认性[3]。其中，CA是PKI中最核心的部分，CA主要负责管理PKI结构下所有用户的证书，为用户提供有关证书的一系列服务。在本篇论文中，本人将利用开源的EJBCA自己动手搭建一个CA证书管理中心，并把它简单的应用到普通业务系统中。

2 EJBCA(Enterprise Java Bean Certificate Authority)

2.1 EJBCA简介

EJBCA是一个企业级的PKI证书颁发机构，它基于J2EE技术。它是一个强大的、高性能的、独立于平台的、灵活的、基于组件的CA证书认证中心。它可以独立使用，也可以集成在其他的J2EE应用程序中[4]。

2.2 EJBCA特点

EJBCA具有以下特点：(1)建立在J2EE规范之上；(2)支持多个CA和多级CA；(3)可以单独运行，也可以与任何J2EE

的应用程序集成；(4)安装和配置十分简单；(5)有强大的基于Web的管理界面；(6)支持个人证书申请和证书的批量生产；(7)服务端和客户端证书能够采用PKCS12、JKS或者PEM格式导出；(8)支持使用Netspace、Mozila、IE等浏览器直接申请证书；(9)由RA添加的新用户可以通过email进行提醒[5,6]。

2.3 EJBCA基本结构

EJBCA包括证书认证中心(CA)、审核注册中心(RA)、证书查询验证服务器(LDAP)和数据库等，如图1所示。

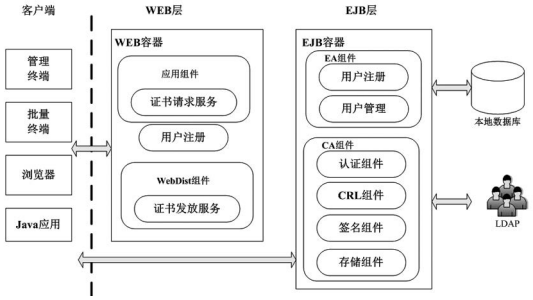

图1 EJBCA基本结构

Fig.1 Basic architecture of EJBCA

(1)证书认证中心(CA)：提供验证用户证书申请、签发证书、定义发布证书失效列表、响应用户证书吊销请求等功能。

(2)审核注册中心(RA)：相当于CA的一个代理结构，RA提供证书申请的登记和审计工作，同时还产生、验证和分发密钥。

(3)证书查询验证服务器(LDAP)：LDAP服务器提供目录浏览服务，负责将注册机构服务器传送过来的用户信息，以及数字证书存储到服务器上，用户可以通过访问LDAP服务器获得证书和证书失效列表。

(4)数据库：保存用户信息、CA信息、用户证书信息，以及作废证书信息、存储日志和统计信息[7]。

2.4 EJBCA与其他类似软件比较

与EJBCA功能相似的还有另外两种软件：OpenSSL和Jcssource。Openssl可以生成CA证书和密钥，但是，所需要的参数非常多，不提供web方式的管理界面，使用非常不方便。Jcssource提供Web方式的证书管理界面，但是证书只能在指定的web服务器上使用。EJBCA可以通过web方式生成CA证书和密钥，也可以通过web方式对证书和密钥的生命周期进行管理。在使用web服务器方面，它需要web服务器必须完全支持J2EE的所有标准，默认使用JBOSS作为web服务器。

3 EJBCA搭建与配置(Construction and configuration of EJBCA)

3.1 EJBCA搭建环境和所需软件

在搭建EJBCA之前，需要做一系列准备工作。我选择Ubuntu为操作系统，数据库选用MySQL，Web应用服务器使用默认的JBOSS。除此之外，还需要另外的一些软件或工作包，它们分别是JDK、ANT、JDBC驱动。以上所述的软件或工具包，在此次搭建过程中，所选用的版本分别为Ubuntu 11.10、JBOSS5.1.0、JDK 1.6.0、Apache-ant 1.8.4、mysql-connector-java-5.1.22。当然还有最为重要的EJBCA，我选用的版本为EJBCA_4_0_12。在确定搭建中所需要的软件和工具包之后，我们就可以开始进行搭建工作了。

3.2 前提准备

在安装EJBCA之前，我们需要把EJBCA安装所需要的一些软件和项目包提前安装好，这样可以为安装EJBCA做好前期准备，我把前期准备主要简单的分为四部分。它们分别是安装JDK、安装ANT、安装JBOSS、安装MySQL。下面将对它们的安装过程进行详细的讲解。

3.2.1 安装JDK

JDK是十分重要的，正如我们前面所述，EJBCA是基于J2EE规范之上的，所以JDK是必不可少的。把JDK安装到/opt目录下，它的安装路径为/opt/jdk。

3.2.2 安装ANT

在安装EJBCA时，我们需要对EJBCA进行编译、打包、部署等操作，这些工作都需要借助ANT来帮助完成。所以，在搭建EJBCA之前，我们需要安装ANT编译打包工具。把ANT安装到/opt目录下，它的安装路径为/opt/ant。

3.2.3 安装JBOSS

我们选择EJBCA默认选用的JBOSS为Web应用服务器。当安装EJBCA时，我们要把EJBCA的服务包发布到JBOSS中，通过JBOSS，我们就可以访问EJBCA所能提供的服务了。安装过程如下：

(1)安装JBOSS：把JBOSS安装到/opt目录下，安装之后的路径为opt/jboss-5.1.0.GA。

(2)修改配置文件：我们需要修改的配置文件位于opt/jboss-5.1.0.GA/server/default/deploy/jbossweb.sar目录下，配置文件名称为server.xml，将其中的PORT端口由默认的8080修改为80，ADDRESS由原始的${jboss.bind.address}修改为0.0.0.0。配置完成之后，JBOSS就允许其他IP地址可以通过80端口访问所发布的项目了。

3.2.4 安装MySQL

MySQL作为EJBCA的数据库，为其存储用户信息、证书信息、日志信息等。其安装过程如下所述：

(1)安装MySQL：使用命令apt-get install mysql自动安装MySQL数据库，在安装过程中提示设置数据库的密码，设置为123456。

(2)JDBC驱动：解压缩mysql-connector-java的安装包，将解压缩后文件夹下的mysql-connector-java-5.1.22-bin.jar拷贝到JBOSS默认的发布路径(opt/jboss-5.1.0.GA/server/default/deploy)下的lib目录中。当我们把此jar包拷贝到JBOSS中的lib文件夹后，JAVA应用程序就可以正常的访问

后台的MySQL数据库。

(3)创建EJBCA数据库：以用户名root，密码123456，登录MySQL数据库。创建一个名字为EJBCA的数据库，并为此数据库的所用用户授权。我们在安装完MySQL之后，就立刻为EJBCA创建数据库，是为后期EJBCA构建证书提供方便。

3.3 EJBCA安装

在做好所有前提准备之后，我们就可以开始安装和配置EJBCA了。EJBCA安装主要可以分为四个部分：安装EJBCA、配置环境变量、配置EJBCA属性文件、构建EJBCA。下面我们对每一部分进行更加详细的讲解和说明[8]。

3.3.1 安装EJBCA

将EJBCA安装到/opt目录下，安装成功之后的路径为/opt/ejbca_4_0_12。

3.3.2 设置环境变量

到此步骤，所有需要安装的软件和项目包都已经安装完毕，所以，我们需要把这些软件的安装路径配置到环境变量中。我们把环境变量配置到/etc/profile文件中，我们之所以选择把环境变量设置在/etc/profile文件中，是因为/etc/profile是属于系统级别的环境变量，在此文件中设置的环境变量对所有用户都起作用。所需要配置的环境变量内容如下：

①export JAVA_HOME=/opt/jdk

②export JBOSS_HOME=/opt/jboss-5.1.0.GA

③export APPSRV_HOME=/opt/jboss-5.1.0.GA

④export J2EE_HOME=/opt/jboss-5.1.0.GA

⑤export ANT_HOME=/opt/ant

⑥export PATH=$JAVA_HOME/bin:$JBOSS_HOME/bin:$ANT_HOME/bin:$PATH

⑦export EJBCA_HOME=/opt/ejbca_4_0_12

⑧export ANT_OPTS=-Xmx640m

3.3.3 配置EJBCA属性文件

EJBCA的配置文件在/opt/ejbca_4_0_12.conf目录下。我们总共需要配置三个文件。

(1)ejbca.properties文件：首先，拷贝ejbca.properties.sample，命名为ejbca.properties。然后，修改ejbca.properties文件，需要修改项如下

appserver.type=jboss；

appserver.home=${env.APPSRV_HOME}；

java.ver=15；

ca.name=Yeeach Root CA；

ca.dn=CN=Yeeach Root CA,O=Yeeach,C=CN

以上需要修改的几项，如果在文件中是注释状态，则取消注释即可，如果文件中不存在，则需要添加。

(2)database.properties文件：首先，拷贝database.properties.sample，命名为database.properties。然后，修改database.properties文件，需要修改项如下：

database.name=mysql；

datasource.mapping=mySQL；database.url=jdbc:mysql://127.0.0.1:3306/ejbca?characterEncoding=UTF-8；

database.driver=com.mysql.jdbc.Driver；

database.username=ejbca；

database.password=ejbca

(3)web.properties文件：首先，拷贝web.properties.sample，命名为web.properties。然后，修改web.properties文件，需要修改项如下：

java.trustpassword=密码；

superadmin.password=密码；

httpsserver.hostname=10.5.110.199(JBOSS服务器IP地址)；

httpserver.pubhttp=80(访问Jboss服务的端口)

(4)构建ejbca

①拷贝jar包：把EJBCA安装路径lib目录下的以bc开头的jar都拷贝到JBOSS的默认发布项目路径下的lib文件夹中；

②ant bootstrap：编译和构建EJBCA，并将EJBCA发布到JBOSS中，此步骤执行成功之后，我们将在JBOSS默认发布服务的路径下看到EJBCA的服务包(ejbca.ear)。

③./run.sh：启动JBOSS服务，此时，我们就可以通过http://10.5.110.199/ejbca访问EJBCA服务了，如图2所示。

图2 EJBCA服务

Fig.2 EJBCA services

④Ant-isntall：开启一个新的终端，执行ant-isntall命令构建证书信息，生成CA、JBOSS服务器、客户端浏览器证书。如果执行成功，在/opt/ejbca_4_0_12下生成一个p12文件夹，其中包含三个文件：superadmin.p12、tomcat.jks和truststore.jks。

⑤./shutdown.sh-S：关闭JBOSS服务。

⑥ant deploy：重新部署一次EJBCA服务，此次部署过程中将配置web服务器的证书文件。部署成功之后，将在server.xml(位于/opt/jboss-5.1.0.GA/server/default/deploy/jbossweb.sar目录下)文件中看到JBOSS已经开启8442和8443端口服务，并且相应的密钥库路径、密钥库密码、验证客户端证书的文件路径和密码，都已经配置完成了，代码如下。其中keystore.jks是密码库，truststore.jks是验证客户端证书库。

<Connector port="8442" address="0.0.0.0" maxThreads="150"
SSLEnabled="true"
keystoreFile="${jboss.server.home.dir}/conf/keystore/keystore.jks"
keystorePass="serverpwd" sslProtocal="TLS"
truststoreFile="${jboss.server.home.dir}/conf/keystore/truststore.jks"
truststorePass="shijianyi" truststoreType="JKS" URLEncoding="UTF-8"/>
<Connector port="8443" address="0.0.0.0" maxThreads="150"
SSLEnabled="true"
keystoreFile="${jboss.server.home.dir}/conf/keystore/keystore.jks"
keystorePass="serverpwd" sslProtocal="TLS"
truststoreFile="${jboss.server.home.dir}/conf/keystore/truststore.jks"
truststorePass="shijianyi"truststoreType="JKS" URLEncoding="UTF-8"/>

⑦./run.sh：再次启动JBOSS

⑧客户端浏览器中导入证书：把生成的证书superadmin.p12拷贝到客户端，进行安装。经过以上步骤，我们就可以通过IP地址https://10.5.110.199:8443/ejbca，以web方式对CA进行管理了，通过http协议和80端口访问EJBCA服务，如图3所示。

图3 访问ejbca服务

Fig.3 Access to ejbca services

4 证书的申请和生成(Application and generation of certificates)

当EJBCA搭建成功之后，在EJBCA的主界面中，单击“Administration”，进入CA证书管理界面。在此，我们主要介绍证书的申请和生成。用户单击左侧列表中的RA Function下的“Add End Entity”，进入用户注册界面，用户需要填写个人信息给CA管理员。CA管理员在审核用户所提交资料的真实性、完整性，以及用户名的唯一性后通过RA将用户资料保存在MySQL数据库中，同时通知用户审核已核准。之后，用户需要进行生成证书请求。在左侧列表中选择Enroll下的“Create KeyStore”，以提交证书申请时的用户名和密码登录EJBCA页面，登录成功之后，选择key的长度等信息，如图4所示，点击“OK”，完成生成证书请求。EJBCA在接到用户生成证书的申请后会调用CA模块，而CA模块通过获取保存在数据库中的用户信息和CA信息，签发用户的个人证书，如图5所示。用户在客户端将此证书导入浏览器，当我们再次访问https://serverIP:8443/ejbca的时候，浏览器会提示让我们选择以那个证书访问系统，如图6所示，服务器根据我们所选择的证书，登录系统。因为yi这个用户只是普通的注册用户，没有管理证书的权限，所以当我们点击列表Miscellaneous下的Administration时，系统会提示我们“您没有权限访问此页面”，从而我们可以知道通过证书可以实现权限管理。以上，我们简述了证书的申请和生成过程。

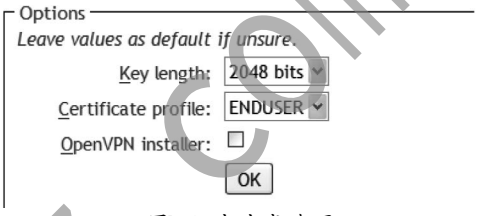

图4 证书生成选项

Fig.4 Certificate generation option

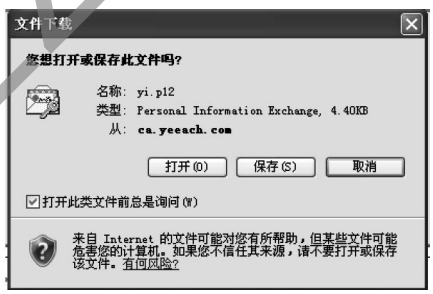

图5 生成证书

Fig.5 Generating certificate

图6 选择证书进行服务访问

Fig.6 Select a certificate for service access

5 EJBCA简单应用(EJBCA simple application)

在完成EJBCA的搭建和证书的生成之后，我们将这个证书应用到实际项目中。在本小节中，我开发了一个非常简单的Java Web Project，项目名称为PKI_JSP。在此项目的根目录下有一个key.jsp文件，在此JSP文件中，主要功能是显示证书信息。将此项目导出为PKI_JSP.war项目包。将此PKI_JSP.war发布到/opt/jboss-5.1.0.GA/opt/jboss-5.1.0.GA/server/default/deploy/目

(下转第31页)