JBoss Enterprise Application Platform 4.3

Common Criteria Configuration Guide

JBoss Enterprise Application Platform

Edition 4.3.3

Last Updated: 2017-10-13

JBoss Enterprise Application Platform 4.3 Common Criteria Configuration Guide

JBoss Enterprise Application Platform
Edition 4.3.3

Darrin Mison
Red Hat Engineering Content Services
dmison@redhat.com

Isaac Rooskov
Red Hat Engineering Content Services
irooskov@redhat.com

Joshua Wulf
Red Hat Engineering Content Services
jwulf@redhat.com

Red Hat

Legal Notice

Abstract

This book describes the configuration of JBoss EAP 4.3 used for the Common Criteria security evaluation

Table of Contents

PREFACE

CHAPTER 1. INTRODUCTION

1.1. PURPOSE OF THIS DOCUMENT

This document is a security guide for administrators and application developers who wish to use JBoss Enterprise Application Platform (JBoss EAP) 4.3.0 GA CP03 in its certified Common Criteria compliant secure configuration. It is intended to be self-contained in addressing the most important issues at a high level, and refers to other existing documentation where more details are needed. Knowledge of the Common Criteria is not required for readers of this document.

JBoss EAP Version 4.3.0 GA CP03 is the subject of this document as the Target of Evaluation (TOE) for Common Criteria certification. JBoss EAP Version 4.3.0 GA CP03 has been evaluated under Common Criteria version 3.1 at level of assurance EAL2 augmented with ALC_FLR.3. This provides assurance that the product has been structurally tested.

All usages of the term "JBoss EAP" in this document refer to the Common Criteria certified configuration of JBoss EAP Version 4.3.0 GA CP03.

Chapter 1 contains a brief introduction to the CC certification & the structure of this book.

Chapter 2 contains the requirements for deploying the certified product.

Chapter 3 contains the steps that are required in downloading &verifying the authenticity of the CC product.

Chapter 4 provides instructions on how to start the server and the different modes of operation.

Chapter 5 contains guidelines for developers creating applications for JBoss EAP

Chapter 6 contains the details of the security implementation & usage limitations of the CC product.

Should there be any discrepancy between information contained in this guide and any other product documentation, the CC Guide information takes precedence, as it addresses the requirements for the evaluated configuration of JBoss EAP.

1.2. WHAT IS A COMMON CRITERIA COMPLIANT SYSTEM?

The *Common Criteria for Information Technology Security Evaluation*, usually known as *Common Criteria* or *CC*, is an internationally-recognized standard (ISO/IEC 15408) used as the basis for independent evaluation of the security properties of an IT product.

Common Criteria provide consumers with an impartial security assurance of a product to predefined levels. These levels range from EAL1 to EAL7, each placing increased demands on the developer for evidence of testing, in turn providing increased assurance within the product for consumers.

Under the Common Criteria Recognition Arrangement (CCRA), members agree to recognize Common Criteria certificates that have been produced by any certificate authorizing participant, in accordance with the terms laid out in the CCRA. Currently, the CCRA is comprised of 22 member nations: Australia, Austria, Canada, the Czech Republic, Finland, France, Germany, Greece, Hungary, India, Israel, Italy, Japan, the Netherlands, New Zealand, Norway, the Republic of Singapore, Spain, Sweden, Turkey, the United Kingdom, and the United States. New members are expected to join in the near future.

A system can be considered to be *CC compliant* if it matches an evaluated and certified configuration. This implies various requirements concerning hardware and software, as well as requirements concerning the operating environment, users, and the ongoing operating procedures.

You can find further information on Common Criteria at http://www.commoncriteria.org.

CHAPTER 2. REQUIREMENTS FOR THE EVALUATED CONFIGURATION

2.1. SOFTWARE REQUIREMENTS

2.1.1. Java Virtual Machine

JBoss EAP is evaluated on the following Java Virtual Machines (JVMs). Only these JVMs are acceptable for the deployment of JBoss EAP.

- Sun JRE 1.5.x & 1.6.x
- BEA JRockit JRE 1.5.x & 1.6.x
- HP-UX JRE 1.5.x & 1.6.x
- IBM JRE 1.5.x & 1.6.x

2.1.2. Operating System

All of the JBoss EAP functionality in the evaluated configuration relies only on the correct operation of the Java virtual machine. Thus it can operate on any operating system that is supported by the respective Java virtual machine. This also means that any hardware supported by the aforementioned operating systems can be used.

2.1.3. Database Servers

JBoss EAP is evaluated with the following relational database systems. Only these database systems with the specific driver versions are acceptable for use with JBoss EAP.

Table 2.1. Allowed Database and JDBC Driver Versions

Database	JDBC Driver

Oracle 10g R2 and Oracle 9i	Oracle 10g R2 version 10.2.0.2.0
	
	Driver download: http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.
	
	`$ md5sum ojdbc14.jar`
	`8ae726d3a32c3cc3adbbe6793ade57f8 ojdbc14.jar`

Microsoft SQL Server 2005	Microsoft SQL Server 2005 Microsoft SQL Server 2005 driver v1.1.1501.101
	
	Driver download: http://www.microsoft.com/downloads/details.aspx?FamilyId=6D483869-816A-44CB-9787-A866235EFC7C&displaylang=en.
	
	`$ md5sum sqljdbc.jar`
	`3bc12b220fd0ed6e074eb26b938185e5 sqljdbc.jar`

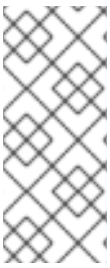
Database	JDBC Driver

MySQL v5.0	MySQL version 5.0.8
	
	Driver download: http://dev.mysql.com/downloads/connector/j/5.0.html.
	
	`$ md5sum mysql-connector-java-5.0.8.zip`
	`569f7284761b8162a2d2ac0a9786581a mysql-connector-java-5.0.8.zip`

PostgreSQL v8.1	PostgreSQL version 8.2-504
	
	Driver download: http://jdbc.postgresql.org/download.
	
	`$ md5sum postgresql-8.2-504.jdbc3.jar`
	`aa8fb66ad71300b635943a8f473a3261 postgresql-8.2-504.jdbc3.jar`

DB2 UDB 8.2.7	DB2 Universal JDBC Driver Version: 2.10.52
	
	Driver download: http://www-01.ibm.com/support/docview.wss?rs=71&uid=swg21251460
	
	`$ md5sum db2jcc.jar`
	`1ae13ee23b595de8b282a7974e5cc25c db2jcc.jar`

DB2 UDB 9.1 Fixpack 3	DB2 Universal JDBC Driver Version: 3.1.57
	
	Driver download: http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.uprun.doc/doc/r0011932.htm
	
	`$ md5sum db2jcc.jar`
	`6b33669a5c2173e65f6bb6618e935b8d db2jcc.jar`



NOTE

The MD5SUM command line examples given are accurate for most Linux and Unix operating systems. Mac OS X includes the equivalent command **md5**.

If you are using Microsoft Windows you will have to download a third party utility to perform these steps as it does not include a MD5SUM tool.

For information on how to configure each database with the JBoss EAP refer to Section 2.5.4, "Database Configuration".

2.2. PHYSICAL REQUIREMENTS

The hardware and software executing JBoss EAP as well as the software critical to security policy enforcement will be protected from unauthorized modification including unauthorized modifications by potentially hostile outsiders. Reasonable physical security measures to ensure that unauthorized

personnel do not have physical access to the hardware running the JBoss EAP software must be implemented.

2.3. PERSONNEL REQUIREMENTS

There shall be one or more competent individuals who are assigned to manage JBoss EAP, its environment and the security of the information it contains. The system administrative personnel shall not be carelessly or willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

The developer of user applications executed by JBoss EAP, including web server applications and enterprise beans, shall be trustworthy and comply with all instructions set forth by the user guidance and evaluated configuration guidance of the JBoss EAP.

2.4. CONNECTIVITY REQUIREMENTS

The operating system and the Java virtual machine operate according to their specification. These external systems shall be configured in accordance with this guidance.

Any other system with which JBoss EAP communicates is assumed to be under the same management control and operate under the same security policy constraints as JBoss EAP.

2.4.1. Cluster Connectivity Requirements

Your JBoss EAP instances must operate in a network segment that is logically separated from any other network segment using a packet filtering mechanism. This packet filter must only allow incoming communication that meets the following criteria:

- the network protocol is TCP
- the destination port is 8080 or 8443

All outgoing communication from one of the JBoss EAP instances is to be allowed.

Each cluster node communicates with the other nodes by means of standard network sockets. Whenever this occurs the client side of each connection has a port number assigned to it by the host operating system from a range of ports that are reserved for client sockets. These ports are referred to as *dynamic* or *ephemeral* ports. They are only used by a connection until it is closed. Once the connection is closed the port is made available for use by other new client connections.You should refer to your operating system documentation if you need to configure this port range.

2.5. CONFIGURATION REQUIREMENTS

The following sections describe modifications to be made to the **`production`** server configuration to comply with CC requirements. It is recommended, however, to back up the production configuration prior to making the changes shown in the following subsections.

Backing up the production configuration simply involves making a copy of the ***`${JBOSS_HOME}`*`/server/production`** directory. If you are using Microsoft Windows you can simply use Windows Explorer to make a copy of this folder using copy-paste and rename the copy to **`production.backup`**. Under UNIX or Linux you can issue the command:

```
cp -pr ${JBOSS_HOME}/server/production
${JBOSS_HOME}/server/production.backup
```

In an emergency you can always retrieve the original files from the installation files.

2.5.1. Setup Configuration

The following configuration steps must be performed to ensure compliance with Common Criteria requirements.

1. Disable Simple Network Management Protocol (SNMP)

   Delete the directory ***${JBOSS_HOME}*/server/production/deploy/snmp-adaptor.sar**

   ```
   $ rm -rf ${JBOSS_HOME}/server/production/deploy/snmp-adaptor.sar
   ```

2. Disable Remote Method Invocation (RMI) under the Internet Inter-ORB Protocol (IIOP)

   To disable RMI/IIOP delete following files:

   - ***${JBOSS_HOME}*/server/production/conf/jacorb.properties**
   - ***${JBOSS_HOME}*/server/production/deploy/iiop-service.xml**
   - ***${JBOSS_HOME}*/server/production/lib/jacorb.jar**
   - ***${JBOSS_HOME}*/server/production/lib/jboss-iiop.jar**

   ```
   $ rm ${JBOSS_HOME}/server/production/conf/jacorb.properties
   $ rm ${JBOSS_HOME}/server/production/deploy/iiop-service.xml
   $ rm ${JBOSS_HOME}/server/production/lib/jacorb.jar
   $ rm ${JBOSS_HOME}/server/production/lib/jboss-iiop.jar
   ```

3. Disable AJP from JBoss Web.

   Comment out the following section from ***${JBOSS_HOME}*/server/production/deploy/jboss-web.deployer/server.xml**:

   ```
   <Connector port="8009" address="${jboss.bind.address}"
   protocol="AJP/1.3" emptySessionPath="true"
   enableLookups="false" redirectPort="8443" />
   ```

4. Disable Clustering High-Availability JNDI service (port 1102)

   1. delete the file ***${JBOSS_HOME}*/server/production/deploy/hajndi-jms-ds.xml**

      ```
      rm ${JBOSS_HOME}/server/production/deploy/hajndi-jms-ds.xml
      ```

   2. copy **jms-ds.xml** from **default** configuration to **production**:

      ```
      cp -p ${JBOSS_HOME}/server/default/deploy/jms-ds.xml
      ${JBOSS_HOME}/server/production/deploy/
      ```

   3. From the file ***${JBOSS_HOME}*/server/production/deploy/cluster-service.xml** comment out the following MBean definitions:

      ```
      <mbean code="org.jboss.ha.jndi.HANamingService"
      ```

```
  name="jboss:service=HAJNDI">
```

```
<mbean
code="org.jboss.invocation.unified.server.UnifiedInvokerHA"
  name="jboss:service=invoker,type=unifiedha">
```

```
<mbean code="org.jboss.invocation.pooled.server.PooledInvokerHA"
  name="jboss:service=invoker,type=pooledha">
```

```
<mbean
 code="org.jboss.cache.invalidation.bridges.JGCacheInvalidationBr
idge"
 name="jboss.cache:service=InvalidationBridge,type=JavaGroups">
```

5. Use password hashing and do not store plain text passwords on the server.

   You should refer to the JBoss Enterprise Application Platform Configuration Guide, Chapter 8, Section 5.3.2 Password Hashing, for details on configuring this: http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.3/doc/Server_Configuration_Guide/html/Using_JBoss_Login_Modules-Password_Hashing.html

2.5.2. Configuring Audit Logging

Audit logging can be configured to print authentication and authorization information for each thread and EJB call.

IMPORTANT

The logging of individual requests is a resource intensive activity. It is recommended that you test the impact that this will have on your server and application performance before enabling this level of logging on a production server.

You enable this level of logging by making the following changes to **`${JBOSS_HOME}/server/production/conf/jboss-log4.xml`**:

1. Set the logging level of the **`SecurityInterceptor`** class to **`TRACE`** by adding the following element to the root element:

```
<category name="org.jboss.ejb.plugins.SecurityInterceptor">
      <priority value="TRACE" />
</category>
```

2. Update the ConversionPattern parameter in the appender/layout element to show thread information by replacing the Default Pattern with the Full Pattern:

```
<!--The full pattern: Date MS Priority [Category] (Thread:NDC)
Message -->
<param name="ConversionPattern" value="%d %-5r %-5p [%c] (%t:%x)
%m%n"/>
```

If you need additional logging for web-based requests, uncomment the **AccessLogValve** in **deploy/jboss-web.deployer/server.xml**.

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
  prefix="localhost_access_log." suffix=".log"
  pattern="common" directory="${jboss.server.home.dir}/log"
  resolveHosts="false" />
```

The access log is saved in the **log** directory of the server configuration.

2.5.3. Security Configuration

The following configuration steps must be performed to ensure security compliance with Common Criteria requirements

2.5.3.1. JBoss SX

All security domains must be created in the context of java:/jaas/ (e.g. java:/jaas/jmx-console).

Custom Login Modules are not permitted; the only login modules allowed are the following:

- **org.jboss.security.auth.spi.UsersRolesLoginModule**
- **org.jboss.security.auth.spi.LdapLoginModule**
- **org.jboss.security.auth.spi.DatabaseServerLoginModule**
- **org.jboss.security.auth.spi.BaseCertLoginModule**

This restriction on login modules is also applicable to the DynamicLoginConfig service.

Only the following security managers are allowed to be configured and used for authentication purposes:

- **org.jboss.security.plugins.JaasSecurityManager**
- **org.jboss.security.plugins.JaasSecurityDomain**

Other modules, such as SRP module are not allowed.

2.5.3.2. Securing MBean Invokers

The **httpha-invoker.sar** found in the deploy directory is a service that provides RMI/HTTP access for EJBs and the JNDI Naming service. This includes a servlet that processes posts of **marshaled org.jboss.invocation.Invocation** objects that represent invocations that should be dispatched onto the MBeanServer. Effectively this allows access to MBeans that support the detached invoker operation via HTTP when sending appropriately formatted HTTP posts. This servlet has to be protected against the use by unprivileged users. To secure this access point you would need to secure the JMXInvokerServlet servlet found in the **httpha-invoker.sar/invoker.war/WEB-INF/web.xml** descriptor.

Refer to http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/4.3.0.cp03/html-single/Server_Configuration_Guide/index.html#Security_on_JBoss-How_to_Secure_the_JBoss_Server for additional details.

The **jmx-invoker-service.xml** is a service that exposes the JMX MBeanServer interface via an RMI compatible interface using the RMI/JRMP detached invoker service. This interface has to be made unavailable to unprivileged users which can be done by using the **org.jboss.jmx.connector.invoker.AuthenticationInterceptor** interceptor for performing identification and authentication using JAAS. Additionally, access control has to be configured using the interceptors of either **org.jboss.jmx.connector.invoker.RolesAuthorization** or **org.jboss.jmx.connector.invoker.ExternalizableRolesAuthorization**.

Refer to http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/4.3.0.cp03/html-single/Server_Configuration_Guide/index.html#Security_on_JBoss-How_to_Secure_the_JBoss_Server for additional details.

2.5.3.3. JBoss Web

The JAAS based authentication and authorization realm implementation (***org.jboss.web.tomcat.security.JBossSecurityMgrRealm***) cannot be replaced. The same is true for the authenticator classes defined for each authentication method (BASIC, CLIENT-CERT, DIGEST, FORM, NONE) in ***${JBOSS_HOME}*/server/production/deploy/jboss-web.deployer/META-INF/jboss-service.xml**.

Additionally, the ***allRolesMode*** within ***${JBOSS_HOME}*/server/production/deploy/jboss-web.deployer/server.xml** must be set to **strict**. This requires the authenticated user to be assigned to one of the **web-app/security-role/role-name** in order to be authorized.

```
<Realm className="org.jboss.web.tomcat.security.JBossSecurityMgrRealm"
  certificatePrincipal="org.jboss.security.auth.certs.SubjectDNMapping"
allRolesMode="strict" />
```

2.5.4. Database Configuration

The default database HSQLDB that the Enterprise Application Platform ships with must be disabled as it is not supported. This section will outline how this can be done and then refer you to information on how to configure supported databases. This must be done in the **production** server profile.

1. Create a default DS file for the desired database. Examples of this file are located in **${JBOSS_HOME}/docs/examples/jca**.

   IMPORTANT

   A **DefaultDS** file must be supplied in the **${JBOSS_HOME}/server/production/deploy** directory.

2. Delete the following files as they refer to the HSQLDB database:
   - **${JBOSS_HOME}/server/production/deploy/hsqldb-ds.xml**
   - **${JBOSS_HOME}/server/production/lib/hsqldb.jar**
   - **${JBOSS_HOME}/server/production/lib/hsqldb-plugin.jar**

- **${JBOSS_HOME}/server/production/deploy/jboss-messaging.sar/clustered-hsqldb-persistence-service.xml**

3. Copy the file **oracle-persistence-service.xml** from **${JBOSS_HOME}/docs/examples/jms/oracle-persistence-service.xml** to **${JBOSS_HOME}/server/production/deploy/jboss-messaging.sar/**.

   This file contains the definition of persistence service for JBoss Messaging when using an Oracle Database as storage.

   NOTE

   The table definitions in **oracle-persistence-service.xml** are not optimized for performance.

4. Place your JDBC driver libraries in the directory **${JBOSS_HOME}/server/production/lib/**.

   If the security policy is to be used, proper permissions must be provided for access to it.

5. When using the Oracle Database, the database persistence plugin definition must be changed in **${JBOSS_HOME}/server/production/deploy/ejb-deployer.xml** from being:

```
<attribute name="DatabasePersistencePlugin">
org.jboss.ejb.txtimer.GeneralPurposeDatabasePersistencePlugin
</attribute>
```

   to being:

```
<attribute name="DatabasePersistencePlugin">
org.jboss.ejb.txtimer.OracleDatabasePersistencePlugin
</attribute>
```

6. Comment out the policy for **HsqlDbRealm** in the ***${JBOSS_HOME}*/server/production/conf/login-config.xml** file as shown.

```
<!-- Security domains for testing new jca framework
<application-policy name = "HsqlDbRealm">
  <authentication>
    <login-module
      code =
"org.jboss.resource.security.ConfiguredIdentityLoginModule"
      flag = "required">
      <module-option name = "principal">sa</module-option>
      <module-option name = "userName">cctest</module-option>
      <module-option name = "password">cc1248</module-option>
      <module-option name = "managedConnectionFactoryName">
        jboss.jca:service=LocalTxCM,name=DefaultDS
      </module-option>
    </login-module>
  </authentication>
</application-policy>
-->
```

For information on how to configure other supported databases refer to http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/4.3.0.cp03/html-single/Server_Configuration_Guide/index.html#alternative_DBs.

2.5.5. Required changes to the included JSM policy

The supplied Java Security Manager policy file that is included with JBoss EAP must be modified as specified below. The policy file that must be edited is ***${JBOSS_HOME}*/bin/security_cc.policy**. The copy of the complete modified policy file can be found in Appendix D, *Required Java Security Manager Policy File*.

1. - codeBase **"file:${jboss.server.home.dir}/tmp/-"** in section 3:
     - Added two more **javax.security.auth.PrivateCredentialPermission** as follows:

       ```
       permission javax.security.auth.PrivateCredentialPermission
       "javax.crypto.spec.SecretKeySpec * \"*\"", "read";
       permission javax.security.auth.PrivateCredentialPermission
       "org.jboss.security.srp.SRPParameters * \"*\"", "read";
       ```

     - **permission java.net.SocketPermission "*", "connect,accept,resolve";** moved from general grant in section 5 to this codeBase.
     - **permission org.jboss.naming.JndiPermission "JAXR", "bind,rebind,unbind,lookup,list,listBindings,createSubcontext";** added to this codeBase.

   For details refer to the grant for code base **"file:${jboss.server.home.dir}/tmp/-"** in Section 3 of the **security_cc.policy** file detailed in Appendix D, *Required Java Security Manager Policy File*.

2. Section 4 changes
   - Testsuite changes to make all tests pass under security manager.
   - Startup time related change
   - JNDI binding problem fixed with adding proper permission to test deploy directory
   - Minor changes in Oracle JDBC driver permissions need for IBM JRE 1.6 to pass the tests

   For details see Section 4 of **security_cc.policy** file in Appendix D, *Required Java Security Manager Policy File*.

3. Section 5 Changes

   The following 2 items have been removed from the general grant section.
   - **permission java.util.PropertyPermission "*", "read";**
   - **permission java.net.SocketPermission "*", "connect";**

   For details see Section 5 of **security_cc.policy** file in Appendix D, *Required Java Security*

Manager Policy File.

4. More detailed comments added throughout the policy file.

2.5.6. Guidance on Configuring Java Security Permissions

The system administrator for the operation of the certified system is expected to configure the security permissions for all enterprise applications that are deployed on the certified system, when the certified system runs in the security manager enabled mode.

NOTE

This configuration is only necessary when running JBoss EAP with the Java Security Manager enabled. Refer to Section 4.2, "Enabling the Java Security Manager" for more details.

Please refer to the Java documentation for information on configuring permissions in the JVM:

- Java 1.5: http://java.sun.com/j2se/1.5.0/docs/guide/security/permissions.html
- Java 1.6: http://java.sun.com/javase/6/docs/technotes/guides/security/permissions.html

A single entry in the Java Security Manager policy that is shipped with the certified system follows the standard Java Standard Edition model. More information is provided in the Java documentaion:

- Java 1.5: http://java.sun.com/j2se/1.5.0/docs/guide/security/PolicyFiles.html
- Java 1.6: http://java.sun.com/javase/6/docs/technotes/guides/security/PolicyFiles.html

An example would be the following:

```
grant codeBase "file:${jboss.server.home.dir}/deploy/jmx-console.war/-" {
  permission java.security.AllPermission;
};
```

This is defined by the certified system by default to provide all permissions to the jmx console web application shipping in the deploy directory.

So if the administrator needs to provide permissions to an enterprise application called as **`TestDeployment.ear`** in the deploy directory of the certified system, then an example entry would be the following:

```
grant codeBase "file:${jboss.server.home.dir}/deploy/jmx-console.war/-" {
 permission java.util.PropertyPermission "*", "read";
 permission javax.security.auth.AuthPermission
"createLoginContext.a_login";
 permission javax.security.auth.AuthPermission "getLoginConfiguration";
};
```

This entry provides the enterprise application called as **`TestDeployment.ear`** to read Java properties as well as the ability to create JAAS login context and obtain JAAS login configuration.

The certified system in the security manager enabled mode is a locked down system that forces the system administrator to configure the necessary security permissions for the operation of the user applications on the certified system.

Any interaction with the JBoss JMX Kernel (which is the standard Java MbeanServer) will require the appropriate **`javax.management.MBeanPermission`** as specified in the Java MbeanServer interface:

- Java 1.5: http://java.sun.com/j2se/1.5.0/docs/api/javax/management/MBeanServer.html
- Java 1.6: http://java.sun.com/javase/6/docs/api/javax/management/MBeanServer.html

We strongly recommend administrators to NOT assign a java.security.AllPermission to any of the user applications.

CHAPTER 3. DOWNLOADING AND VERIFYING THE PACKAGES

JBoss EAP is delivered on line through the Red Hat JBoss Customer Support Portal (CSP) at https://support.redhat.com/jbossnetwork/restricted/main.html and through the Red Hat Network (RHN) at https://rhn.redhat.com. JBoss EAP is available as ZIP files from CSP and as ZIP and RPM files from RHN.

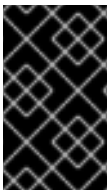
To ensure the authenticity of the downloaded software you need to verify the authenticity of the files and their source.

IMPORTANT

Unless specifically stated otherwise the screenshots and other samples shown in this section are only examples. The actual presentation of the download websites may change overtime.

3.1. VERIFY THE AUTHENTICITY OF THE DOWNLOAD SITE.

Red Hat JBoss Customer Support Portal and Red Hat Network are secure sites. This is indicated by the 'lock' icon in the browser status bar. The lock may also present itself in the address bar depending on what browser you are using.



IMPORTANT

The following images have been taken with the Firefox3 and Firefox2 web browsers. While most popular web-browsers display this information in a very similar manner it may differ slightly to these images.

Figure 3.1. Secure site 'lock' icon displayed in the Firefox3 status bar.

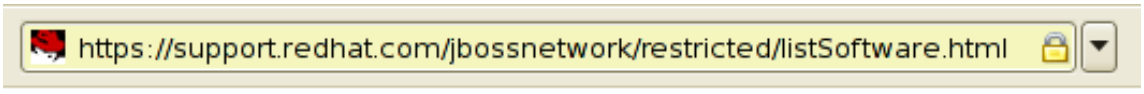


Figure 3.2. Secure site 'lock' icon displayed in the Firefox2 address bar.

If these items are not visible you may wish to check the authenticity of the site by viewing the identity certificate. To give an example of how this can be done, we will use the Firefox web browser.

Within the Firefox browser, go to Tools in the top menu bar and then click on Page Info. From here click the Security icon and then the **`View Certificate`** button.

The certificate will display details such as who the owner of the page is, who issued the certificate, when it was issued and when it expires as well as SHA1 and MD5 fingerprint verification strings. An example of the certificate for https://rhn.redhat.com follows.

Figure 3.3. The RHN certification certificate

If neither of the lock icons are present in your browser and a verified certificate cannot be found, this may mean that you are not at the correct site. If you are unable to reach the secure Red Hat JBoss Customer Support Portal or Red Hat Network sites you should contact Red Hat Support and report this problem.

3.2. DOWNLOADING JBOSS EAP FROM THE RED HAT JBOSS CUSTOMER SUPPORT PORTAL

JBoss EAP can be downloaded from the Red Hat JBoss Customer Support Portal, https://support.redhat.com/jbossnetwork/.

After you have logged in, navigate to **JBoss Enterprise Platforms**, **Application Platform** and select **4.3.0.GA_CP03** from the **Version** menu.

Figure 3.4. Software downloads page showing available files

The packages can be downloaded using either the download link on that page, or by using the download link on the software details page for that package. The software details page is reached by clicking on the package name rather than the download link.

The software details page for each package also contains the MD5 checksum value for that package. These values are used to verify the integrity of your downloaded files.

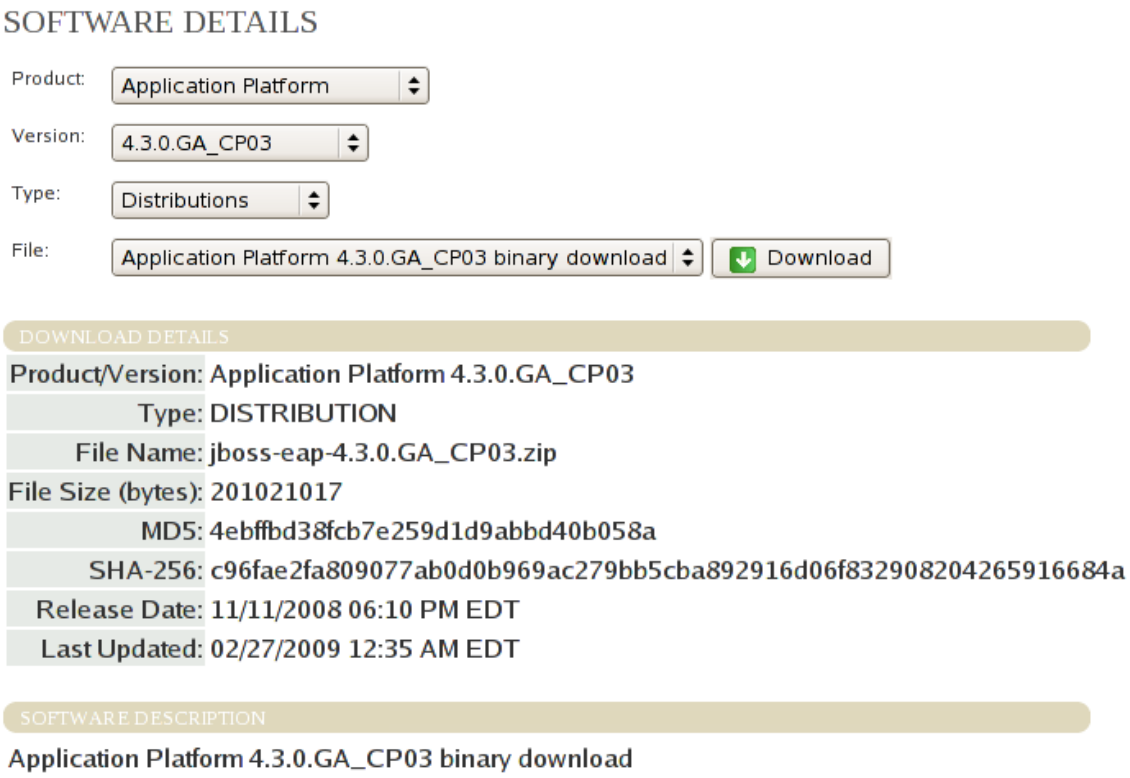


Figure 3.5. MD5 information displayed for a download at the Red Hat JBoss Customer Support Portal

You also must download the patch for Security Notice CVE-2009-0027. This is found by clicking on the **`Security Advisories`** link. Installation instructions for this patch are found in Section 3.5, “Installing the Security Notice CVE-2009-0027 patch”.

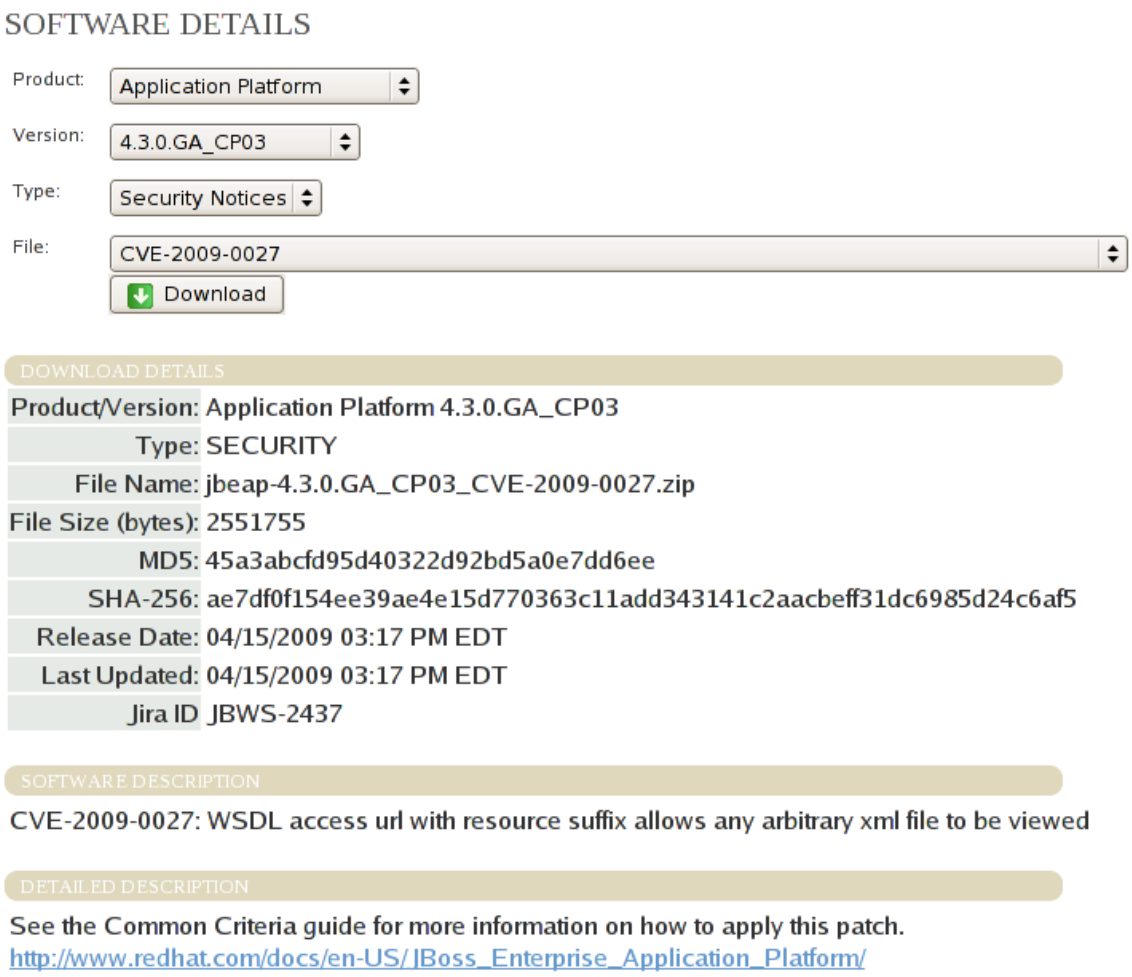


Figure 3.6. Security Notice for CVE-2009-0027

After downloading these files you need to validate their authenticity according to the directions in Section 3.4, “Verifying the Downloaded Files”.

3.3. DOWNLOADING JBOSS EAP FROM THE RED HAT NETWORK

JBoss EAP can be downloaded from the Red Hat Network, https://rhn.redhat.com.

After you have logged in, you can locate the appropriate download channel for your system by clicking on the **Channels** menu item at the top of the page. From the **Filter by Product Channel** menu select the **`JBoss Application Platform`**, version **`4.3.0`**, the architecture of your server, and then click on the **`Filter`** button.

The following image is an example filter search and displays all versions of the EAP that are available. For the certified version of JBoss EAP select **`JBoss Enterprise Application Platform 4.3.0`**.

IMPORTANT

Note that the menu items here refer to the version as being **`4.3.0 CP03`** while the listed files use **`4.3.0.GA_CP03`** in their names. They are both referring to the same version.

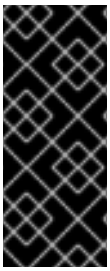
Figure 3.7. Searching for the JBoss Enterprise Application Platform

Each of the displayed channels for JBoss EAP delivers the files in the slightly different way. These are described in the following two sections.

3.3.1. JBoss Enterprise Middleware (All)

This channel provides JBoss EAP packaged in zip files as well as a Java installer. These can be installed on any supported platform.

After you have selected the JBoss Enterprise Middleware channel, you are presented with the **Download Details** page for JBoss EAP. Initially the **Details** tab will be selected. Clicking on the tab labeled **Downloads** and the list of the downloads for JBoss EAP will be displayed.



IMPORTANT

The files listed here are those of the most recent JBoss Enterprise Application Server release. Once 4.3.CP03 is superseded by another version you will have to click on the **View ISO Images for Older Releases** link and then **JBoss Enterprise Application Platform 4.3.0 CP03** to access the files for the evaluated configuration.

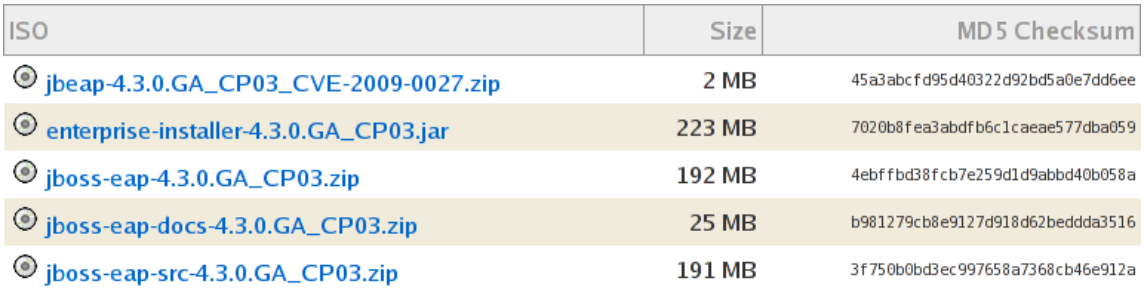


Figure 3.8. JBoss EAP download file list

The packages listed above can be explained as follows:

- **enterprise-installer-4.3.0.GA_CP03.jar**: The graphical installer for EAP 4.3.0.CP03.
- **jboss-eap-4.3.0.GA_CP03.zip**: The files that make up the EAP 4.3.0.CP03 install.
- **jboss-eap-docs-4.3.0.GA_CP03.zip**: The documentation for EAP 4.3.0.CP03.

- **`jboss-eap-src-4.3.0.GA_CP03.zip`**: The source code distribution for EAP 4.3.0.CP03.
- **`jbeap-4.3.0.GA_CP03_CVE-2009-0027.zip`**: The patch for Security Notice CVE-2009-0027.

As well as the installation package of your choice you must also download the patch for Security Notice CVE-2009-0027. Installation instructions for this patch are found in Section 3.5, "Installing the Security Notice CVE-2009-0027 patch".

After downloading these files you need to validate their authenticity according to the directions in Section 3.4, "Verifying the Downloaded Files".

3.3.2. Red Hat Enterprise Linux AS 4, ES 5, Server 5

This channel provides JBoss EAP as an ISO disk image that contains the RPM packages for installation on Red Hat Enterprise Linux Systems.

After you have selected the appropriate Red Hat Enterprise Linux channel, you are presented with the **`Download Details`** page. Initially the **Details** tab will be selected. Clicking on the tab labeled **Downloads** and the list of the downloads for JBoss EAP will be displayed.

Only a single file will be listed here, the the ISO image that contains all the RPM packages for JBoss EAP. This ISO image also includes the RPM packages for the CVE-2009-0027 Security Advisory patch.

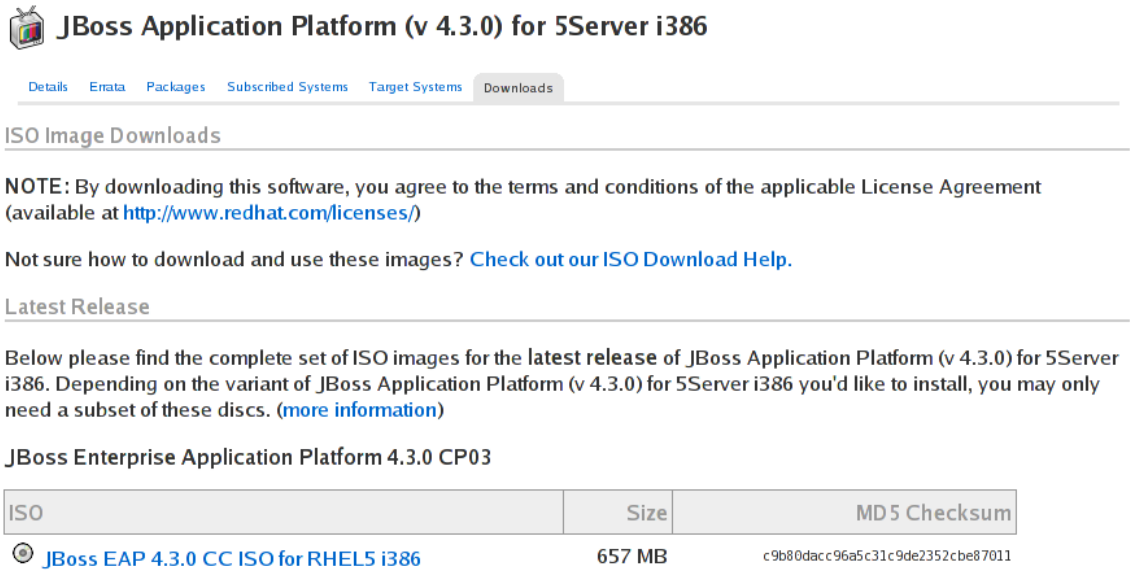


Figure 3.9. ISO Image downloads for Red Hat Enterprise Linux Server 5

After downloading these files you need to validate their authenticity according to the directions in Section 3.4, "Verifying the Downloaded Files".

3.4. VERIFYING THE DOWNLOADED FILES

The software details page for each download also contains the MD5 checksum values for that download. These values are used to verify the integrity of your downloaded files. You can use the **`md5sum`** utility as detailed below to calculate the checksum values of the files to compare to the supplied values on the website. The checksum values are also documented in Table 3.1, "JBoss EAP MD5 checksum values" for completeness.

NOTE

The command line examples given are accurate for most Linux and Unix operating systems. Mac OS X includes the equivalent command **md5**.

If you are using Microsoft Windows you will have to download a third party utility to perform these steps as it does not include a MD5SUM tool.

After you have downloaded the file, run the **md5sum** utility and specify the file you downloaded as the first argument as demonstrated here:

Example 3.1. Using the md5sum tool on Linux or Unix

```
$ md5sum jboss-eap-4.3.0.GA_CP03.zip
4ebffbd38fcb7e259d1d9abbd40b058a jboss-eap-4.3.0.GA_CP03.zip
```

The values that are generated by the **md5sum** utility must be the same as both the values that are displayed on the Downloads page for the file and those documented in Table 3.1, "JBoss EAP MD5 checksum values". If they are not the same then then your download is either incomplete or corrupted. You will need to download it again. If after several attempts you are unable to download a copy of the file that produces a valid checksum values you should open a support case to report the problem.

Below is the complete list of the MD5 checksums for all the JBoss EAP packages available for download.

Table 3.1. JBoss EAP MD5 checksum values

File	MD5 Checksum
JBEAP4.3.0-re20090408.0-i386-disc1-ftp.iso	5561e56f493049b6cd147cdf481d6b57
JBEAP4.3.0-re20090408.0-x86_64-disc1-ftp.iso	07ef18ab2b14858be0efd9a7e1af1e5d
RHEL5.2-JBEAP-4.3.0-20090408.0-i386-disc1-ftp.iso	c9b80dacc96a5c31c9de2352cbe87011
RHEL5.2-JBEAP-4.3.0-20090408.0-x86_64-disc1-ftp.iso	077f2fd28ce3a05c769e2963f29c97cf
enterprise-installer-4.3.0.GA_CP03.jar	7020b8fea3abdfb6c1caeae577dba059
jboss-eap-4.3.0.GA_CP03.zip	4ebffbd38fcb7e259d1d9abbd40b058a
jboss-eap-docs-4.3.0.GA_CP03.zip	b981279cb8e9127d918d62beddda3516
jboss-eap-src-4.3.0.GA_CP03.zip	3f750b0bd3ec997658a7368cb46e912a
jbeap-4.3.0.GA_CP03_CVE-2009-0027.zip	45a3abcfd95d40322d92bd5a0e7dd6ee

3.5. INSTALLING THE SECURITY NOTICE CVE-2009-0027 PATCH

After you have installed JBoss EAP you must also install the Security Notice CVE-2009-0027 patch. This patch resolves an issue where a remote attacker could read arbitrary XML files with the permissions of the EAP process. You can refer to http://rhn.redhat.com/errata/RHSA-2009-0349.html for additional information regarding this exploit.

The exact files you will need to download will vary according to whether you have installed the RPM version of JBoss EAP or the zip version.

The files for the RPM install are included in the ISO image.

The files for the zip package install must be downloaded separately as described in Section 3.2, “Downloading JBoss EAP from the Red Hat JBoss Customer Support Portal”.

You can verify the authenticity of the downloaded files by using md5sum and the checksum values listed here.

Table 3.2. MD5 checksums for patch files

File	MD5 Checksum
jbossws-2.0.1-3.SP2_CP04.4.1.ep1.el5.noarch.rpm	2b94cc1b052280f2a8cf5856c64972c5
jbossws-2.0.1-3.SP2_CP04.4.1.ep1.el5.src.rpm	ccb6c9bd951b3d4df4a4004973533980
jbossws-2.0.1-3.SP2_CP04.4.ep1.el4.noarch.rpm	bf61c04a503d914186d0bd68f47dea9b
jbossws-2.0.1-3.SP2_CP04.4.ep1.el4.src.rpm	31a4fd98ce9eb02a3b98d7fa7306e8ba
jbeap-4.3.0.GA_CP03_CVE-2009-0027.zip	45a3abcfd95d40322d92bd5a0e7dd6ee

For a Red Hat Enterprise Linux 4 or 5 RPM Installation you can install the patch RPM that you downloaded just like any other RPM package. You can do this using the command line or using the GUI tool of your choice.

Example 3.2. Installing the RPM patch on Red Hat Enterprise Linux 5

```
$ rpm -ivh jbossws-2.0.1-3.SP2_CP04.3.1.ep1.el5.noarch.rpm
Preparing...                ########################################### [100%]
   1:jbossws                ########################################### [100%]
```

Installation of the patch on a JBoss EAP zip file install simply requires you to overwrite two jar files in the install with those that you have downloaded.

Procedure 3.1. Installing the patch on a zip install

1. Extract the two JAR files from **`jbeap-4.3.0.GA_CP03_CVE-2009-0027.zip`**.
2. Copy **`jbossws-client.jar`** over the existing one in ***`%JBOSS_HOME%`*`/client`**.

3. Copy **jbossws-core.jar** over the existing one in ***%JBOSS_HOME%*/server/production/deploy/jbossws.sar/**.

4. Repeat step 2 for any other server profiles that you use, such as for development and testing.

3.6. CONFIRMING THE VERSION OF YOUR JBOSS EAP INSTALLATION

There are three ways in which you can verify the version number of your JBoss EAP installation.

1. Using the **-V** with the startup script

   You can retrieve information about the version of your JBoss EAP installation by running the same script used to start the server with the **-V** switch. For Linux and Unix installations this script is **run.sh** and on Microsoft Windows installations it is **run.bat**. Regardless of platform the script is located in ***$JBOSS_HOME*/bin**. Using these scripts to actually start your server is dealt with in Chapter 4, *Launching the JBoss EAP Server*.

   Running this script with the **-V** switch will not start the JBoss EAP server nor does it require the JBoss EAP server to be running. It displays information about the JBoss EAP version and its configured Java environment. Below is an example of using this on an installation of JBoss EAP on Red Hat Linux. Note the version number (**JBoss 4.3.0.GA_CP03**) displayed as the last item before the license information.

```
$ ./run.sh -V
=========================================================================
===

  JBoss Bootstrap Environment

  JBOSS_HOME: /opt/JBoss/4.3.CP03/jboss-eap-4.3/jboss-as

  JAVA: java

  JAVA_OPTS: -Dprogram.name=run.sh -server -Xms1503m -Xmx1503m -
Dsun.rm
i.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.lang.ClassLoader.allowArraySyntax=true -
Djava.net.preferIPv4Stack
=true

  CLASSPATH: /opt/JBoss/4.3.CP03/jboss-eap-4.3/jboss-as/bin/run.jar

=========================================================================
===

JBoss 4.3.0.GA_CP03 (build: SVNTag=JBPAPP_4_3_0_GA_CP03
date=200810241616)

Distributable under LGPL license.
See terms of license at gnu.org.

$
```

2. Using the JMX Console

When the JBoss EAP server is running you can retreive many details about it using the JMX Console at http://localhost:8080/jmx-console

The MBean which contains the version information has the Domain Name of **`jboss.system`** and type of **`server`**. It is directly accessible at http://localhost:8080/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.system%3Atype%3DServer. The attributes that contain the version information are: VersionNumber, Version and VersionName.

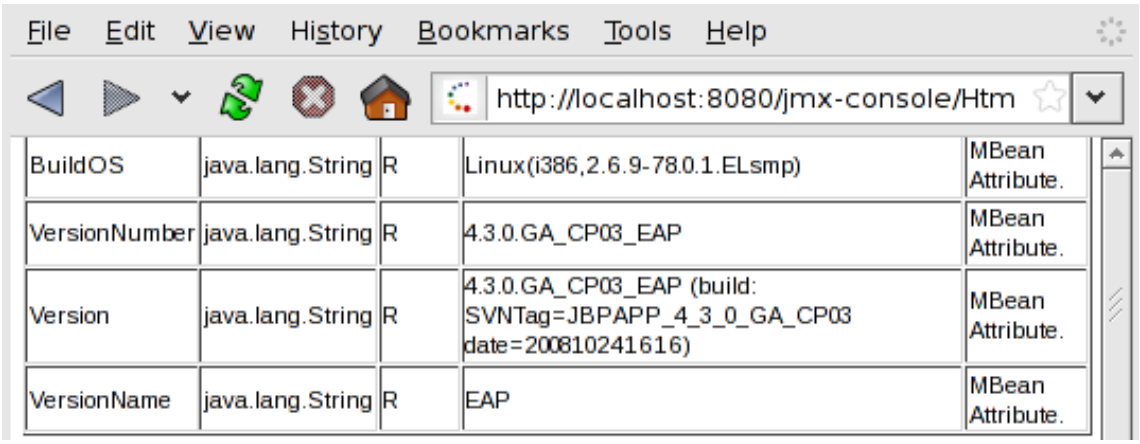


Figure 3.10. Version details displayed in JMX Console

3. Using the Web Console

When the JBoss EAP server is running you can can retrieve its version information from the first page of the Web Console as well. This is located at http://localhost:8080/web-console/.

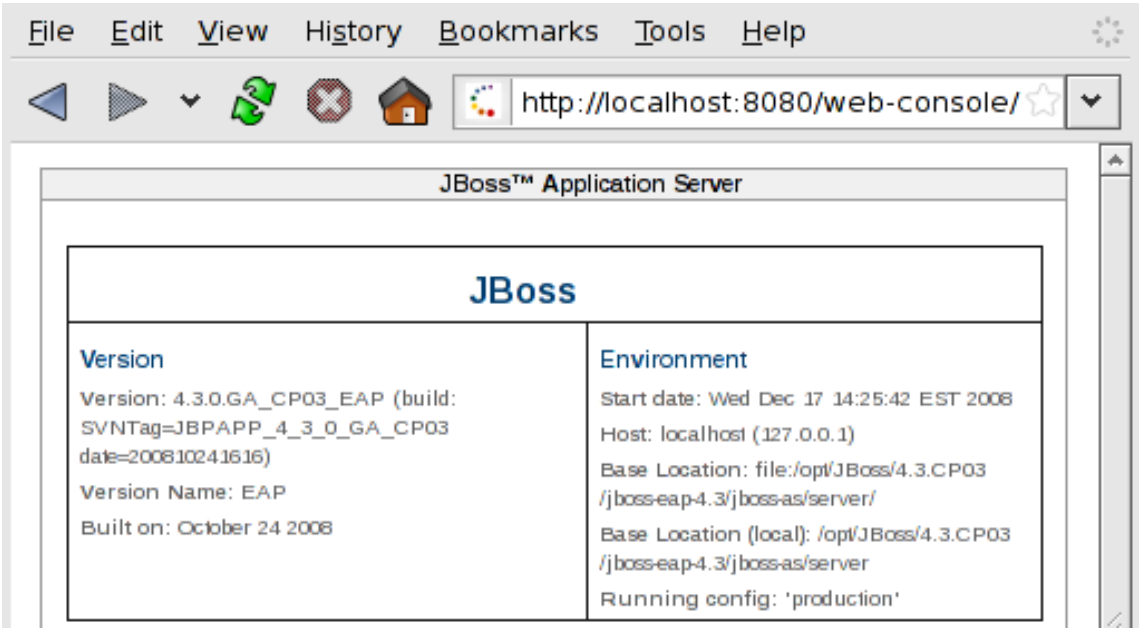


Figure 3.11. Version details displayed in Web Console

Additionally, when the server is started the version is both echoed to the console and written to ***`$JBOSS_HOME`*`/server/production/log/boot.log`**:

```
12:33:33,798 INFO  [Server] Starting JBoss (MX MicroKernel)...
12:33:33,798 INFO  [Server] Release ID: JBoss [EAP] 4.3.0.GA_CP03 (build:
SVNTag=JBPAPP_4_3_0_GA_CP03 date=200810241616)
```

CHAPTER 4. LAUNCHING THE JBOSS EAP SERVER

JBoss EAP includes a startup script for both Linux/Unix platforms & Microsoft Windows as well a configuration file , **`run.conf`**, which determines the startup environment of the server. For Linux and Unix installations the startup script is **`run.sh`** and on Microsoft Windows installations it is **`run.bat`**. Regardless of platform the script is located in ***`$JBOSS_HOME`*`/bin`**.

JBoss EAP has been certified both with and without the use of the Java Security Manger. If you use the Java Security Manager, you must also use the policy settings as defined in Appendix D, *Required Java Security Manager Policy File*. Operating JBoss EAP using the Java Security Manager and different policy settings is not considered to be a certified configuration.

This allows two modes of operation which affect how JBoss EAP can protect itself against the behavior of applications. As the administrator of your JBoss EAP server, you must decide which mode of operation is most appropriate. These modes are discussed fully below.

4.1. STARTING THE JBOSS EAP SERVER

To start the JBoss EAP server simply use the supplied start up script which is appropriate for your platform. However you must use the **`-c`** command parameter to specify the **`production`** server configuration.

Example 4.1. Starting the JBoss EAP server on Unix or Linux

```
$ cd $JBOSS_HOME/bin
$ ./run.sh -c production
```

Example 4.2. Starting the JBoss EAP server on Windows

```
cd %JBOSS_HOME%/bin
$ run.bat -c production
```

JBoss EAP's default behavior is to run without the use of the Java Security Manager. This means that any application deployed on JBoss EAP will be running in the same namespace as JBoss EAP itself. In this environment it is possible that an application deployed on JBoss EAP may interfere with the execution of JBoss EAP itself either accidentally or intentionally.

If you choose to run without using the Java Security Manger & specified policy settings then you are responsible for performing your own risk analysis to ensure that deployed applications do not contain bugs that may be abused by users of the application to circumvent the security functionality of JBoss EAP.

4.2. ENABLING THE JAVA SECURITY MANAGER

By enabling the Java Security Manager with the specified policy JBoss EAP is protected from any application deployed on it accidentally or intentionally interfering with its operation.

This policy limits the granting of full permissions to those jar files included with the evaluated configuration.

WARNING

If you use the Java Security Manager, you must configure the policy settings as explained in Section 2.5.5, "Required changes to the included JSM policy" . Operating JBoss EAP using the Java Security Manager with different policy settings is not considered to be a certified configuration.

You must edit the file **run.conf** located in the Enterprise Platform home directory at **/jboss-as/server/production/** and uncomment the lines indicated below to enable the Java Security Manager. Once those items are uncommented from **run.conf**, simply start the server using the supplied startup script (**run.sh** or **run.bat**) as normal.

Example 4.3. run.conf with Java Security Manager enabled

```
# Uncomment the following to run with Common Criteria configuration
## Specify the Security Manager Policy
POLICY="security_cc.policy"
#
## Specify the Security Manager options
JAVA_OPTS="$JAVA_OPTS -Djava.security.manager -
Djava.security.policy==$POLICY"
echo "================================================================="
echo "
"
echo "    Common Criteria Configuration (Security Manager Enabled)"
echo "
"
echo "================================================================="
## End of Common Criteria configuration
```

IMPORTANT

run.conf is part of the production configuration of the EAP. Only the production configuration with the additional configuration information specified in this guide is allowed in the Common Criteria Configuration.

IBM JRE 1.6 and the Java Security Manager

IBM JRE 1.6 uses a default policy provider which does not work correctly with the JBossEAP security policy. You must change the JRE configuration to use the standard policy provider if you want to use IBM JRE 1.6 to host JBossEAP with the Java Security Manager enabled.

You do this by editing the file ***${JAVA_HOME}*/jre/lib/security/java.security** and setting the value of policy.provider to **sun.security.PolicyFile** instead of **org.apache.harmony.security.fortress.DefaultPolicy**:

```
policy.provider=sun.security.provider.PolicyFile
```

Additional Policy file configuration

Users and administrators are free to add their own permission blocks to the policy file, however the permissions that are specified for JBoss EAP cannot be changed; doing so will invalidate the certification. Indeed any modifications of the security policy except what has been specified within this guide, will invalidate the certification configuration. Refer to Section 2.5.6, “Guidance on Configuring Java Security Permissions” for additional information on this topic.

CHAPTER 5. DEVELOPMENT GUIDE FOR THE COMMON CRITERIA CERTIFIED SYSTEM

This section describes the guidelines to be followed by a trusted developer who develops programs or applications that run on the secure certified system.

5.1. ENTERPRISE APPLICATION

An enterprise application is a Java Enterprise Edition (formerly J2EE) version 1.4 compliant application software. Typically the application accepts requests from clients, does some processing and responds with results. The enterprise application that is developed by the trusted developer is hereby referred to as a *user application*.

The types of enterprise applications include the following:

1. Web Applications based on Servlets and Java Server Pages (JSP)
2. Enterprise Java Beans (EJB)
3. JavaEE 1.4 Web Service Applications which can be based on Stateless EJBs or Plain Old Java Objects (POJOs) deployed as Java Servlets.

5.2. GENERAL RESTRICTIONS

The trusted software developer needs to follow the following restrictions when developing secure software for the certified system.

1. Application Programming Interfaces (API) that is not documented in the product documentation MUST not be used. Please refer to the section on the guidance for System administrators to configure the certified system, for more information on providing security permissions to the user applications.
2. The programming restrictions mandated by the Enterprise Java Beans Specification version 2.1 (Section 25.2, pages 562-564) (http://jcp.org/aboutJava/communityprocess/final/jsr153/index.html) should be strictly followed.

Enterprise Java Beans Specification Developer Restrictions

The restrictions are:

- An enterprise bean must not use read/write static fields. Using read-only static fields is allowed. Therefore, it is recommended that all static fields in the enterprise bean class be declared as final.
- An enterprise bean must not use thread synchronization primitives to synchronize execution of multiple instances.
- An enterprise bean must not use the AWT functionality to attempt to output information to a display or to input information from a keyboard.
- An enterprise bean must not use the `java.io` package to attempt to access files and directories in the file system.

- An enterprise bean must not attempt to listen on a socket, accept connections on a socket, or use a socket for multicast.
- The enterprise bean must not attempt to query a class to obtain information about the declared members that are not otherwise accessible to the enterprise bean because of the security rules of the Java language. The enterprise bean must not attempt to use the Reflection API to access information that the security rules of the Java programming language make unavailable.
- The enterprise bean must not attempt to
  - create a class loader
  - obtain the current class loader
  - set the context class loader
  - set security manager
  - create a new security manager
  - stop the JVM
  - or change the input, output, and error streams
- The enterprise bean must not attempt to set the socket factory used by ServerSocket, Socket, or the stream handler factory used by URL.
- The enterprise bean must not attempt to manage threads. The enterprise bean must not attempt to start, stop, suspend, or resume a thread, or to change a thread's priority or name. The enterprise bean must not attempt to manage thread groups.
- The enterprise bean must not attempt to obtain the security policy information for a particular code source.
- The enterprise bean must not attempt to load a native library.
- The enterprise bean must not attempt to gain access to packages and classes that the usual rules of the Java programming language make unavailable to the enterprise bean.
- The enterprise bean must not attempt to define a class in a package.
- The enterprise bean must not attempt to access or modify the security configuration objects (Policy, Security, Provider, Signer, and Identity).
- The enterprise bean must not attempt to use the subclass and object substitution features of the Java Serialization Protocol.
- The enterprise bean must not attempt to pass this as an argument or method result. The enterprise bean must pass the result of `SessionContext.getEJBObject`, `SessionContext.getEJBLocalObject`, `EntityContext.getEJBObject`, or `EntityContext.getEJBLocalObject` instead.

These restrictions will be enforced by the Java Security Manager when the certified system is run in the security manager enabled mode. The system administrators of the certified system have to ensure that they do not provide the user applications security permissions that relax any of the aforementioned restrictions, thereby endangering the security and stability of the certified system.

5.3. DEVELOPER ADVICE FOR USER CREDENTIALS IN REMOTE METHOD INVOCATION (RMI)

In Remote Method Invocation credentials are transmitted from client to server. These credentials populate the security context in the method invocation object. This is done through the **`setPrincipal`** and **`setCredential`** methods.

Example 5.1. Setting Principal and Credential

```
MethodInvocation mi = new MethodInvocation();
mi.setPrincipal(new SimplePrincipal("myusername"));
mi.setCredential("mypassword");
```

These additional payloads can be retrieved at the server side using similar methods on the invocation object.

Example 5.2. Retrieving Principal and Credential

```
Principal p = mi.getPrincipal();
Object cred = mi.getCredential();
// Now do authentication (and then authorization)
```

CHAPTER 6. OVERVIEW OF THE SECURITY FUNCTIONS

The following sections describe the JBoss security functions included in the product evaluation.

6.1. ACCESS CONTROL

JBoss Enterprise Application Platform has access control mechanisms to restrict access for the following request types:

HTTP

URLs and paths provided with URLs can be protected from access by subjects.

EJB

EJBs and associated method names can be protected from invocation by subjects.

JMS

Message queue destinations and topic destinations can be protected from access by subjects.

Web Services

Plain Old Java Objects (POJOs) deployed as Servlets and Session Beans can be protected from access by subjects.

JMX

The JMX invokers can be protected by validating the role of the authenticated user.

For more information refer to the JBoss EAP Server Configuration Guide:
http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/4.3.0.cp03/html/Server_Configuration_Guide/

6.2. AUDIT

JBoss Enterprise Application Platform can generate audit records for access control events. Attempts to access to web resources, invocation of EJB methods, unauthorized message destinations, and regular Web Service related access control can all be logged. As the administrator you can select the level of events to audit.

The JBoss Application server generates log events at start-up time and when it is shutdown:

Example 6.1. JBoss EAP start up log events

```
00:30:18,876 INFO [Server] Starting JBoss (MX MicroKernel)...
300:30:18,876 INFO [Server] Release ID: JBoss [EAP] 4.3.0.GA_CP03
(build: SVNTag=JBPAPP_4_3_0_GA_CP03 date=200810241616)
00:30:18,877 DEBUG [Server] Using config:
org.jboss.system.server.ServerConfigImpl@18dfef8
00:30:18,877 DEBUG [Server] Server type: class
org.jboss.system.server.ServerImpl
00:30:18,877 DEBUG [Server] Server loaded through:
org.jboss.system.server.NoAnnotationURLClassLoader
00:30:18,877 DEBUG [Server] Boot URLs:
```

Example 6.2. JBoss EAP shutdown log events

```
2008-12-12 00:32:16,460 DEBUG [org.jboss.deployment.MainDeployer]
Destroying jboss.system:service=MainDeployer
2008-12-12 00:32:16,460 DEBUG [org.jboss.deployment.MainDeployer]
Destroyed jboss.system:service=MainDeployer
2008-12-12 00:32:16,460 DEBUG [org.jboss.system.ServiceController]
removing service: jboss.system:service=MainDeployer
2008-12-12 00:32:16,460 DEBUG [org.jboss.system.ServiceController]
removing jboss.system:service=MainDeployer from server
2008-12-12 00:32:16,460 DEBUG [org.jboss.system.ServiceController]
Stopped 3 services
2008-12-12 00:32:16,460 DEBUG [org.jboss.system.server.Server] Deleting
server tmp/deploy directory
2008-12-12 00:32:16,463 INFO  [org.jboss.system.server.Server] Shutdown
complete
```

The audit facility is based on the integrated log4j mechanism. Log4j has three main components: loggers, appenders and layouts. These three types of components work together to enable developers to log messages according to message type and level, and to control at run-time how these messages are formatted and where they are reported.

The audit information is recorded in text files which can be reviewed using tools from the underlying operating system, such as pagers or editors.

User information (principal name) appears *only* in the first log that records the authentication request, and also in the ERROR log generated if the authentication is unsuccessful. Subsequent log events do not record explicitly the user executing the methods.

User information can be obtained by using the container and thread ids that are recorded in each audit log and remain during the life of the user session.

In the example below (Example 6.3, “Log output”) the first log entry informs that authentication for container 753, thread id 826541 has been requested by principal name “scott”. The second log records the execution of a method, and, although the principal name does not appear, it can be inferred by looking all logs with the same container and thread id.

Example 6.3. Log output

```
2008-12-12 16:04:33,753 826541 TRACE
[org.jboss.ejb.plugins.SecurityInterceptor]
(WorkerThread#0[127.0.0.1:33182]:) Authenticated  principal=scott
2008-12-12 16:04:33,753 826541 TRACE
[org.jboss.ejb.plugins.SecurityInterceptor]
(WorkerThread#0[127.0.0.1:33182]:) method=public abstract
org.jboss.test.jca.securedejb.CallerIdentity
org.jboss.test.jca.securedejb.CallerIdentityHome.create() throws
javax.ejb.CreateException,java.rmi.RemoteException, interface=HOME,
requiredRoles=[CallerIdentityUser]
```

6.2.1. Enabling Additional Logging

Additional logging for EJB application requests has been configured during the setup process of this guide when audit logging was configured. For more information regarding audit logging configuration refer to Section 2.5.1, “Setup Configuration”

6.3. CLUSTERING

A cluster is a group of linked systems (nodes) working closely together to increase efficiency. Clustering enables the execution of applications on several parallel servers. In a JBoss EAP cluster each node is a JBoss server instance. Several JBoss server instances are grouped together to form a cluster, also known as a "partition".

JBoss EAP implements two different cluster configurations: a failover cluster and a load-distribution cluster.

In a failover cluster scenario a single node services requests from clients. In the event that the node fails another node in the cluster continues to service requests.

In a load-distribution cluster scenario multiple nodes service requests from clients. In this way a single address is serviced with the power of multiple systems.

In both cases, the server state is distributed across different servers. If any of the servers fails the application is still accessible via other non-failed cluster nodes.

Communication between the different cluster nodes ensures the data consistency of the following information:

- Applications - an application deployed on one node is replicated to the other nodes of the cluster (farming deployment)
- State of HTTP sessions, EJB 3.0 session beans, EJB 3.0 entity beans, as well as Hibernate persistence objects (distributed state replication service using JBoss Cache)
- State of HTTP sessions and EJB 2.x session beans (distributed state replication service using HASessionState MBean)
- JNDI state (JBoss HA-JNDI)
- JMS queues

6.4. IDENTIFICATION AND AUTHENTICATION

Each user is assigned a unique user identifier. Access control decisions and auditing use this identifier. JBoss EAP authenticates the user's claimed identity before allowing the user to perform any actions. After successful authentication JBoss EAP associates the identifier with the thread spawned for the user.

JBoss EAP provides different identification and authentication mechanisms for various request types.

HTTP and Web Services

HTTP-basic authentication, HTTP-digest authentication, form-based authentication, client certificate based authentication.

EJB

username and password based authentication, client certificate based authentication.

JMS

username and password based authentication.

JBoss EAP uses JBoss SX framework to implement identification and authentication. The JBossSX framework utilizes the Java Authentication and Authorization Service (JAAS) provided by the Java Virtual Machine. The authentication capabilities of JAAS are used to implement the declarative role-based J2EE security model.

The following authentication back-ends are configurable with the JAAS modules.

- File-based storage
- BaseCertLoginModule
- LDAP
- Databases accessible through JDBC

Password quality can be enforced with configuration options for the JAAS modules provided by JBoss EAP.

For information on how to configure the JAAS modules, refer to the Using JBoss Login Modules section of the Server Configuration Guide.

6.5. TRANSACTION ROLLBACK

JBoss EAP supports the aggregation of operations into transactions, which can be applied and rolled back consistently.

A transaction is a unit of work containing one or more operations involving one or more shared resources having ACID properties. ACID is an acronym for atomicity, consistency, isolation and durability - the four important properties of transactions.

Atomicity

A transaction must be atomic. This means that either all the work done in the transaction must be performed, or none of it must be performed. Doing only part of a transaction is not allowed.

Consistency

When a transaction is completed, the system must be in a stable and consistent condition.

Isolation

Different transactions must be isolated from each other. This means that the partial work done in one transaction is not visible to other transactions until the transaction is committed, and that each process in a multi-user system can be programmed as if it was the only process accessing the system.

Durability

The changes made during a transaction are made persistent when it is committed. When a transaction is committed, its changes will not be lost, even if the server crashes afterward.

The default transaction manager for JBoss EAP is JBoss Transactions, a fast in-VM transaction manager implementation.

Traditionally ACID transaction systems have shared three characteristics:

1. Transactions are short lived
2. Resources (such as databases) are locked for the duration of the transaction
3. Participants have a high degree of trust with each other.

The advent of the Internet and Web services has given rise to distributed transactions between participants unknown to each other. JBoss Transactions adds native support for Web services transactions by providing the components necessary to build interoperable, reliable, multi-party, Web services-based applications with minimum effort. The programming interfaces are based on the Java API for XML Transactions (JAXTX) and include protocol support for the WS-AtomicTransaction and WS-BusinessActivity specifications. JBoss is designed to support multiple coordination protocols.

JBoss supports both local and distributed transactions. A transaction is considered to be distributed if it spans multiple process instances, i.e. virtual machines (VMs). Typically a distributed transaction will contain participant that are located within multiple VMs but the transaction is coordinated in a separate VM (or co-located with one of the participants). If the deployment requires distributed transactions then the Web Services transactions component can be utilized, which uses SOAP/HTTP.

APPENDIX A. RPM LISTINGS FOR A RED HAT ENTERPRISE LINUX 4 INSTALLATION

JBoss EAP for Red Hat Enterprise Linux 4 is made up of the following list of specific RPMs available from the Red Hat Network. Although the Red Hat Network lists other RPM in addition to these only those RPMs listed here should be downloaded and used to install JBoss EAP.

- antlr-2.7.6-3jpp.ep1.2.noarch.rpm
- asm-1.5.3-1jpp.ep1.2.el4.noarch.rpm
- avalon-framework-4.1.5-1jpp.ep1.2.noarch.rpm
- avalon-logkit-1.2-2jpp.ep1.2.noarch.rpm
- bcel-5.1-1jpp.ep1.2.noarch.rpm
- bea-stax-1.2.0-0.rc1.2jpp_1rh.noarch.rpm
- bea-stax-api-1.2.0-0.rc1.2jpp_1rh.noarch.rpm
- berkeleydb-2.0.90-1jpp.ep1.1.noarch.rpm
- bsf-2.3.0-6jpp.ep1.2.noarch.rpm
- bsh-1.3.0-5jpp.ep1.2.noarch.rpm
- cglib-2.1.3-2jpp.ep1.6.el4.noarch.rpm
- concurrent-1.3.4-7jpp.ep1.6.el4.noarch.rpm
- dom4j-1.6.1-2jpp.ep1.2.noarch.rpm
- dtdparser-1.21-2jpp.ep1.2.noarch.rpm
- geronimo-j2ee-1.4-apis-1.0-3jpp.ep1.2.noarch.rpm
- glassfish-jaf-1.1.0-0jpp.ep1.12.el4.noarch.rpm
- glassfish-javamail-1.4.0-0jpp.ep1.10.el4.noarch.rpm
- glassfish-jaxb-2.1.4-1jpp.ep1.2.el4.noarch.rpm
- glassfish-jaxws-2.1.1-1jpp.ep1.3.el4.noarch.rpm
- glassfish-jsf-1.2_09-0jpp.ep1.2.el4.noarch.rpm
- glassfish-jstl-1.2.0-0jpp.ep1.10.el4.noarch.rpm
- gnu-getopt-1.0.12-1jpp.ep1.2.noarch.rpm
- gnu-trove-1.0.2-5jpp.ep1.2.noarch.rpm
- hibernate3-3.2.4-1.SP1_CP06.0jpp.ep1.3.el4.noarch.rpm
- hibernate3-annotations-3.2.1-5.GA_CP03.1jpp.ep1.1.el4.noarch.rpm

- hibernate3-annotations-javadoc-3.2.1-5.GA_CP03.1jpp.ep1.1.el4.noarch.rpm
- hibernate3-commons-annotations-0.0.0-3.1jpp.ep1.1.el4.noarch.rpm
- hibernate3-ejb-persistence-3.0-api-3.2.1-1jpp.ep1.1.noarch.rpm
- hibernate3-ejb-persistence-3.0-api-javadoc-3.2.1-1jpp.ep1.1.noarch.rpm
- hibernate3-entitymanager-3.2.1-2.GA_CP04.1jpp.ep1.2.el4.noarch.rpm
- hibernate3-entitymanager-javadoc-3.2.1-2.GA_CP04.1jpp.ep1.2.el4.noarch.rpm
- hibernate3-javadoc-3.2.4-1.SP1_CP06.0jpp.ep1.3.el4.noarch.rpm
- hibernate3-validator-0.0.0-2.1jpp.ep1.1.el4.noarch.rpm
- hsqldb-1.8.0.8-2.patch01.1jpp.ep1.1.noarch.rpm
- icu4j-3.4.5-2jpp.ep1.1.noarch.rpm
- isorelax-0.1-0.20041111.2jpp.ep1.1.noarch.rpm
- jacorb-2.3.0-1jpp.ep1.4.noarch.rpm
- jakarta-commons-beanutils-1.7.0-2jpp.ep1.5.el4.noarch.rpm
- jakarta-commons-codec-1.3-2jpp.ep1.2.noarch.rpm
- jakarta-commons-collections-3.1-1jpp.ep1.1.noarch.rpm
- jakarta-commons-dbcp-1.2.1-7jpp.ep1.1.noarch.rpm
- jakarta-commons-digester-1.7-6jpp.ep1.1.noarch.rpm
- jakarta-commons-discovery-0.4-1jpp.ep1.1.noarch.rpm
- jakarta-commons-fileupload-1.1.1-3jpp.ep1.1.noarch.rpm
- jakarta-commons-httpclient-3.0.1-1jpp.ep1.1.noarch.rpm
- jakarta-commons-logging-1.0.4-6jpp.ep1.1.noarch.rpm
- jakarta-commons-pool-1.3-2jpp.ep1.1.noarch.rpm
- jakarta-commons-transaction-1.1-3jpp.1.ep1.1.noarch.rpm
- jakarta-slide-webdavclient-2.1-3jpp.ep1.2.noarch.rpm
- javassist-3.8.0-1.ep1.el4.noarch.rpm
- jaxen-1.1-1jpp.ep1.2.noarch.rpm
- jboss-aop-1.5.5-3.CP03.1.ep1.el4.noarch.rpm
- jbossas-4.3.0-3.GA_CP03.6.ep1.el4.noarch.rpm
- jbossas-client-4.3.0-3.GA_CP03.6.ep1.el4.noarch.rpm

- jboss-cache-1.4.1-5.SP10.1.ep1.el4.noarch.rpm
- jboss-common-1.2.1-0jpp.ep1.2.noarch.rpm
- jboss-jaxr-1.2.0-SP1.0jpp.ep1.5.el4.noarch.rpm
- jboss-messaging-1.4.0-2.SP3_CP04.3.ep1.el4.noarch.rpm
- jboss-microcontainer-1.0.2-4.1.el4.noarch.rpm
- jboss-profiler-1.0-0.1.CR5.1jpp.ep1.2.noarch.rpm
- jboss-remoting-2.2.2-3.SP10.0jpp.ep1.1.el4.noarch.rpm
- jboss-seam-1.2.1-3.JBPAPP_4_3_0_GA.ep1.11.el4.noarch.rpm
- jboss-seam-docs-1.2.1-3.JBPAPP_4_3_0_GA.ep1.11.el4.noarch.rpm
- jboss-serialization-1.0.3-1jpp.ep1.3.noarch.rpm
- jbossts-4.2.3-1.SP5_CP02.1jpp.ep1.1.el4.noarch.rpm
- jbossweb-2.0.0-6.CP08.0jpp.ep1.1.el4.noarch.rpm
- jbossws-2.0.1-3.SP2_CP04.1.ep1.el4.noarch.rpm
- jbossws-common-1.0.0-2.GA_CP02.1.ep1.el4.noarch.rpm
- jbossws-framework-2.0.1-1.GA_CP02.1.ep1.el4.noarch.rpm
- jbossws-spi-1.0.0-1.GA_CP01.1.ep1.el4.noarch.rpm
- jbossxb-1.0.0-2.SP3.0jpp.ep1.3.el4.noarch.rpm
- jcommon-1.0.12-1jpp.ep1.3.el4.noarch.rpm
- jdom-1.0-4jpp.ep1.1.noarch.rpm
- jfreechart-1.0.9-1jpp.ep1.3.el4.noarch.rpm
- jgroups-2.4.4-2.ep1.el4.noarch.rpm
- joesnmp-0.3.4-1jpp.ep1.2.noarch.rpm
- juddi-0.9-0.rc4.2jpp.ep1.8.el4.noarch.rpm
- log4j-1.2.14-1jpp.ep1.1.noarch.rpm
- msv-1.2-0.20050722.4jpp.ep1.1.noarch.rpm
- msv-xsdlib-1.2-0.20050722.4jpp.ep1.1.noarch.rpm
- odmg-3.0-3jpp.ep1.2.noarch.rpm
- qdox-1.6.1-1jpp.ep1.4.noarch.rpm
- quartz-1.5.2-1jpp.ep1.2.noarch.rpm

- regexp-1.4-3jpp.ep1.2.noarch.rpm
- relaxngDatatype-1.0-2jpp.ep1.2.noarch.rpm
- rh-eap-docs-4.3.0-4.GA_CP03.ep1.2.el4.noarch.rpm
- servletapi6-6.0.10-3jpp.ep1.1.noarch.rpm
- snmptrapappender-1.2.8-5jpp.ep1.2.noarch.rpm
- tanukiwrapper-3.2.1-2jpp.ep1.1.i386.rpm
- tomcat5-servlet-2.4-api-5.5.17-6jpp.ep1.2.noarch.rpm
- ws-commons-policy-1.0-2jpp.ep1.7.el4.noarch.rpm
- wsdl4j-1.6.2-1jpp.ep1.8.noarch.rpm
- ws-jaxme-0.5.1-2jpp.ep1.1.noarch.rpm
- wstx-3.1.1-1jpp.ep1.2.noarch.rpm
- xalan-j2-2.7.0-2jpp.ep1.3.noarch.rpm
- xerces-j2-2.7.1-9jpp.ep1.1.noarch.rpm
- xjavadoc-1.1-1jpp.ep1.1.noarch.rpm
- xml-commons-1.3.03-7jpp.ep1.3.noarch.rpm
- xml-commons-jaxp-apis-1.3.03-7jpp.ep1.3.noarch.rpm
- xml-commons-resolver-1.1-1jpp.ep1.1.noarch.rpm
- xml-im-exporter-1.1-2jpp.ep1.1.noarch.rpm
- xml-security-1.3.0-1jpp.ep1.2.noarch.rpm
- xom-1.0-2jpp.ep1.1.noarch.rpm
- xpp2-2.1.10-4jpp.ep1.1.noarch.rpm
- xpp3-1.1.3.4-1.o.2jpp.ep1.1.noarch.rpm

APPENDIX B. RPM LISTINGS FOR A RED HAT ENTERPRISE LINUX 5 INSTALLATION

JBoss EAP for Red Hat Enterprise Linux 5 is made up of the following list of specific RPMs available from the Red Hat Network. Although the Red Hat Network lists other RPM in addition to these only those RPMs listed here should be downloaded and used to install JBoss EAP.

- asm-1.5.3-1jpp.ep1.2.el5.noarch.rpm
- bea-stax-1.2.0-0.rc1.2jpp.ep1.1.el5.noarch.rpm
- bea-stax-api-1.2.0-0.rc1.2jpp.ep1.1.el5.noarch.rpm
- berkeleydb-2.0.90-1jpp.ep1.1.el5.noarch.rpm
- cglib-2.1.3-2jpp.ep1.6.el5.noarch.rpm
- concurrent-1.3.4-8jpp.ep1.6.el5.1.noarch.rpm
- dom4j-1.6.1-2jpp.ep1.5.el5.2.noarch.rpm
- dtdparser-1.21-2jpp.ep1.2.el5.2.noarch.rpm
- geronimo-j2ee-1.4-apis-1.0-3jpp.ep1.3.el5.1.noarch.rpm
- glassfish-jaf-1.1.0-0jpp.ep1.12.el5.1.noarch.rpm
- glassfish-javamail-1.4.0-0jpp.ep1.10.el5.noarch.rpm
- glassfish-jaxb-2.1.4-1jpp.ep1.4.el5.2.noarch.rpm
- glassfish-jaxws-2.1.1-1jpp.ep1.3.el5.noarch.rpm
- glassfish-jsf-1.2_09-0jpp.ep1.2.el5.noarch.rpm
- glassfish-jstl-1.2.0-0jpp.ep1.10.el5.noarch.rpm
- gnu-getopt-1.0.12-1jpp.ep1.2.el5.2.noarch.rpm
- gnu-trove-1.0.2-5jpp.ep1.2.el5.2.noarch.rpm
- hibernate3-3.2.4-1.SP1_CP06.0jpp.ep1.3.el5.noarch.rpm
- hibernate3-annotations-3.2.1-5.GA_CP03.1jpp.ep1.1.el5.1.noarch.rpm
- hibernate3-annotations-javadoc-3.2.1-5.GA_CP03.1jpp.ep1.1.el5.1.noarch.rpm
- hibernate3-commons-annotations-0.0.0-3.1jpp.ep1.1.el5.noarch.rpm
- hibernate3-ejb-persistence-3.0-api-3.2.1-1jpp.ep1.1.el5.noarch.rpm
- hibernate3-ejb-persistence-3.0-api-javadoc-3.2.1-1jpp.ep1.1.el5.noarch.rpm
- hibernate3-entitymanager-3.2.1-2.GA_CP04.1jpp.ep1.2.el5.noarch.rpm
- hibernate3-entitymanager-javadoc-3.2.1-2.GA_CP04.1jpp.ep1.2.el5.noarch.rpm

- hibernate3-javadoc-3.2.4-1.SP1_CP06.0jpp.ep1.3.el5.noarch.rpm
- hibernate3-validator-0.0.0-2.1jpp.ep1.1.el5.noarch.rpm
- icu4j-3.4.5-2jpp.ep1.2.el5.noarch.rpm
- isorelax-0.1-0.20041111.2jpp.ep1.2.el5.4.noarch.rpm
- jacorb-2.3.0-1jpp.ep1.5.el5.noarch.rpm
- jakarta-commons-transaction-1.1-3jpp.1.ep1.3.el5.1.noarch.rpm
- jakarta-slide-webdavclient-2.1-3jpp.ep1.3.el5.1.noarch.rpm
- javassist-3.8.0-1jpp.ep1.2.el5.noarch.rpm
- jaxen-1.1-1jpp.ep1.4.el5.2.noarch.rpm
- jboss-aop-1.5.5-3.CP03.1.ep1.el5.noarch.rpm
- jbossas-4.3.0-3.GA_CP03.6.2.ep1.el5.noarch.rpm
- jbossas-client-4.3.0-3.GA_CP03.6.1.ep1.el5.noarch.rpm
- jboss-cache-1.4.1-5.SP10.1.ep1.el5.noarch.rpm
- jboss-common-1.2.1-0jpp.ep1.2.el5.1.noarch.rpm
- jboss-jaxr-1.2.0-SP1.0jpp.ep1.5.el5.noarch.rpm
- jboss-messaging-1.4.0-2.SP3_CP04.3.ep1.el5.noarch.rpm
- jboss-microcontainer-1.0.2-4.1.el5.noarch.rpm
- jboss-profiler-1.0-0.1.CR5.1jpp.ep1.3.el5.1.noarch.rpm
- jboss-remoting-2.2.2-3.SP10.0jpp.ep1.1.el5.noarch.rpm
- jboss-seam-1.2.1-3.JBPAPP_4_3_0_GA.ep1.8.el5.1.noarch.rpm
- jboss-seam-docs-1.2.1-3.JBPAPP_4_3_0_GA.ep1.8.el5.1.noarch.rpm
- jboss-serialization-1.0.3-1jpp.ep1.4.el5.noarch.rpm
- jbossts-4.2.3-1.SP5_CP02.1jpp.ep1.2.el5.noarch.rpm
- jbossweb-2.0.0-6.CP08.0jpp.ep1.1.el5.noarch.rpm
- jbossws-2.0.1-3.SP2_CP04.1.1.ep1.el5.noarch.rpm
- jbossws-common-1.0.0-2.GA_CP02.1.ep1.el5.noarch.rpm
- jbossws-framework-2.0.1-1.GA_CP02.1.ep1.el5.noarch.rpm
- jbossws-spi-1.0.0-1.GA_CP01.1.ep1.el5.noarch.rpm
- jbossxb-1.0.0-2.SP3.0jpp.ep1.3.el5.1.noarch.rpm

- jcommon-1.0.12-1jpp.ep1.3.el5.noarch.rpm
- jfreechart-1.0.9-1jpp.ep1.3.el5.1.noarch.rpm
- jgroups-2.4.4-2.ep1.el5.noarch.rpm
- joesnmp-0.3.4-1jpp.ep1.2.el5.2.noarch.rpm
- juddi-0.9-0.rc4.2jpp.ep1.8.el5.noarch.rpm
- msv-1.2-0.20050722.5jpp.ep1.1.el5.2.noarch.rpm
- msv-xsdlib-1.2-0.20050722.5jpp.ep1.1.el5.2.noarch.rpm
- odmg-3.0-3jpp.ep1.2.el5.1.noarch.rpm
- qdox-1.6.1-1jpp.ep1.5.el5.noarch.rpm
- quartz-1.5.2-1jpp.ep1.5.el5.noarch.rpm
- relaxngDatatype-1.0-2jpp.ep1.2.el5.2.noarch.rpm
- rh-eap-docs-4.3.0-4.GA_CP03.ep1.2.el5.noarch.rpm
- servletapi6-6.0.10-3jpp.ep1.1.el5.noarch.rpm
- snmptrapappender-1.2.8-5jpp.ep1.2.el5.1.noarch.rpm
- tanukiwrapper-3.2.1-2jpp.ep1.1.el5.i386.rpm
- ws-commons-policy-1.0-2jpp.ep1.5.el5.noarch.rpm
- wsdl4j16-1.6.2-0jpp.ep1.2.el5.1.noarch.rpm
- ws-jaxme-0.5.1-2jpp.ep1.1.el5.1.noarch.rpm
- wstx-3.1.1-1jpp.ep1.2.el5.1.noarch.rpm
- xml-im-exporter-1.1-2jpp.ep1.1.el5.1.noarch.rpm
- xml-security-1.3.0-1jpp.ep1.3.el5.2.noarch.rpm
- xom-1.0-2jpp.ep1.3.el5.1.noarch.rpm
- xpp2-2.1.10-4jpp.ep1.2.el5.1.noarch.rpm
- xpp3-1.1.3.4.O-2jpp.ep1.1.el5.1.noarch.rpm

APPENDIX C. PORT CONFIGURATION IN JBOSS EAP

Table C.1. TCP Port Configuration

PORT	CONFIG	ENABLED	PURPOSE
1098	conf/jboss-service.xml	Enabled	RMI Naming Service
1099	conf/jboss-service.xml	Enabled	RMI bootstrap naming service
1100	deploy/cluster-service.xml	Disabled	Clustering
1101	deploy/cluster-service.xml	Disabled	Clustering
3528	conf/jacorb.properties	Disabled	IIOP Port assigned by IANA
3529	conf/jacorb.properties	Disabled	IIOP/SSL Port assigned by IANA
3873	deploy/ejb3.deployer/META-INF/jboss-service.xml	Enabled	EJB3 Remoting Connector
4444	conf/jboss-service.xml	Enabled	RMI JRMP Invoker
4445	conf/jboss-service.xml	Enabled	RMI Pooled Invoker
4446	conf/jboss-service.xml	Enabled	Remoting server connector
4447	conf/jboss-service.xml	Enabled	Remoting server connector
4448	deploy/cluster-service.xml	Disabled	PooledInvokerHA
4457	deploy/jboss-messaging.sar/remoting-bisocket-service.xml	Enabled	Messaging bi-socket connector between client and server
7900	deploy/jboss-messaging.sar/clustered-hsqldb-persistence-service.xml	Disabled	
8009	deploy/jbossweb.deployer/server.xml	Disabled	AJP Connector
8080	deploy/jboss-web.deployer/server.xml	Enabled	Http Connector
8083	conf/jboss-service.xml	Enabled	RMI - Mini web server needed for RMI Classloading

Table C.2. UDP Port Configuration

PORT	CONFIG	ENABLED	PURPOSE
1102	cluster-service.xml/HA-JNDI	Disabled	
1161	deploy/snmp-adaptor.sar	Disabled	snmp
1162	deploy/snmp-adaptor.sar	Disabled	snmp
7500	jboss-web-cluster.sar/diagnostics ejb3-entity-cache-service.xml/diagnostics cluster-service.xml/HAPartition/diagnostics jboss-messaging.sar/clustered-hsqldb-persistence-service.xml/diagnostics	Enabled	
43333	ejb3-entity-cache-service.xml	Enabled	Clustering cache service for ejb3 entity beans
45551	ejb3-clustered-sfsbcache-service.xml	Enabled	EJB3 Stateful Session Bean Clustered Cache
45566	cluster-service.xml/HAPartition	Enabled	
45567	jboss-messaging.sar/clustered-hsqldb-persistence-service.xml	Disabled	
45568	jboss-messaging.sar/clustered-hsqldb-persistence-service.xml	Disabled	
45577	jboss-web-cluster.sar	Enabled	Tomcat5 Clustering

APPENDIX D. REQUIRED JAVA SECURITY MANAGER POLICY FILE

```
//**************************************************************************
// Common Criteria Evaluated Configuration Java2 Security Manager Policy
// Author: Anil Saldhana
//**************************************************************************

//*********************************************************
//
//   Section 1: JBOSS code with codebase references in time
//              of JBOSS startup
//   (Permissions are given fully)
//   Do not modify this section.
//
//*********************************************************
grant codeBase "file:${user.dir}/run.jar" {
  permission java.security.AllPermission;
};

grant codeBase "file:${user.dir}/../lib/*" {
  permission java.security.AllPermission;
};

grant codeBase "file:${user.dir}/../server/production/lib/-" {
  permission java.security.AllPermission;
};

//******************* End of Section 1 **********************

//*********************************************************
//
//   Section 2:  Java JDK Core Code
//               Trusted core Java code
//   (Permissions are given fully)
//   Do not modify this section.
//
//*********************************************************
grant codeBase "file:${java.home}/lib/ext/-" {
   permission java.security.AllPermission;
};
grant codeBase "file:${java.home}/lib/*" {
   permission java.security.AllPermission;
};
// For java.home pointing to the JDK jre directory
grant codeBase "file:${java.home}/../lib/*" {
   permission java.security.AllPermission;
};

//******************* End of Section 2 **********************


//*********************************************************
//
//   Section 3:  Permissions assigned to JBoss Core Codebase
```

```
//               Trusted JBoss code
//
//   Do not modify this section.
//
//*********************************************************
grant codeBase "file:${jboss.home.dir}/bin/-" {
   permission java.security.AllPermission;
};

// Trust all the jars in the server lib that JBoss has shipped
grant codeBase "file:${jboss.home.dir}/lib/-" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/work/-" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/activation.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/antlr.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/asm-attrs.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/asm.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/autonumber-plugin.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/avalon-framework.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/bcel.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/bindingservice-plugin.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/bsf.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/bsh-deployer.jar" {
```

```
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/bsh.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/cglib.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/commons-codec.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/commons-collections.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/commons-httpclient.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/commons-logging.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/dom4j.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/ejb3-persistence.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/el-api.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/hibernate3.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/hibernate-annotations.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/hibernate-commons-annotations.jar" {
   permission java.security.AllPermission;
};

grant codeBase
```

```
"file:${jboss.server.home.dir}/lib/hibernate-entitymanager.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/hibernate-validator.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/hsqldb.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/hsqldb-plugin.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jacorb.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/javassist.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jaxen.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-cache-jdk50.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/jboss-common-jdbc-wrapper.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-ejb3x.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossha.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-hibernate.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-iiop.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-j2ee.jar" {
   permission java.security.AllPermission;
```

```
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-jaxrpc.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-jaxws.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-jca.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-jsr77.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-jsr88.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/jbossjta-integration.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossjta.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-management.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/jboss-messaging-client.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-messaging.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-monitoring.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-remoting-int.jar"
{
   permission java.security.AllPermission;
};
```

```
grant codeBase "file:${jboss.server.home.dir}/lib/jboss-remoting.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-saaj.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/jboss-serialization.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-srp.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbosssx.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-transaction.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossts-common.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jboss-vfs.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossws-common.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossws-framework.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossws-jboss42.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jbossws-spi.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jgroups.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jmx-adaptor-plugin.jar"
{
```

```
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jnpserver.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/joesnmp.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/jsp-api.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/log4j.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/log4j-snmp-appender.jar"
{
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/mail.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/mail-plugin.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/properties-plugin.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/quartz-all.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/lib/scheduler-plugin-example.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/scheduler-plugin.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/servlet-api.jar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/lib/xmlentitymgr.jar" {
   permission java.security.AllPermission;
};
```

```
// DEPLOY DIR

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-ha-local-jdbc.rar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-ha-xa-jdbc.rar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-local-jdbc.rar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/deploy/jboss-xa-jdbc.rar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/deploy/jms-ra.rar" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/deploy/quartz-ra.rar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/httpha-invoker.sar/-" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-web-cluster.sar/jboss-web-
cluster.aop" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jaxb-api.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jaxb-impl.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jboss-jaxb-intros.jar" {
   permission java.security.AllPermission;
};
```

```
grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jboss-jaxrpc.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jboss-jaxws.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jboss-saaj.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jbossws-core.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/jbossws-native.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/policy.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/stax-api.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/wsdl4j.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/wstx.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jbossws.sar/xmlsec.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/juddi-service.sar/juddi.jar" {
   permission java.security.AllPermission;
};

grant codeBase
```

```
"file:${jboss.server.home.dir}/deploy/juddi-service.sar/juddi-saaj.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/juddi-service.sar/juddi-service.jar"
{
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/juddi-service.sar/juddi.war" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/juddi-service.sar/scout.jar" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/uuid-key-generator.sar/*" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/deploy/ejb3.deployer/-" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-aop-jdk50.deployer/-" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-bean.deployer/-" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-web.deployer/*" {
   permission java.security.AllPermission;
};

grant codeBase
"file:${jboss.server.home.dir}/deploy/jboss-web.deployer/jsf-libs/*" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/deploy/management/-" {
   permission java.security.AllPermission;
};

grant codeBase "file:${jboss.server.home.dir}/deploy/jmx-console.war/-" {
   permission java.security.AllPermission;
};
```

```
grant codeBase "file:${jboss.server.home.dir}/tmp/-" {

   permission java.io.FilePermission
      "${jboss.server.home.dir}/-", "read,write,delete";
   permission java.io.FilePermission
      "${java.io.tmpdir}", "read,write,delete";

   permission java.io.FilePermission "<<ALL FILES>>", "read";

   // MBean permissions
   permission javax.management.MBeanTrustPermission "*";
   permission javax.management.MBeanServerPermission "findMBeanServer";
   permission javax.management.MBeanPermission "*", "*";

   permission java.lang.RuntimePermission "setContextClassLoader";
   permission java.lang.RuntimePermission "accessDeclaredMembers";
   permission java.lang.RuntimePermission "createClassLoader";
   permission java.lang.RuntimePermission
      "org.jboss.security.SecurityAssociation.setPrincipalInfo";
   permission java.lang.RuntimePermission
      "org.jboss.security.SecurityAssociation.getPrincipalInfo";
   permission java.lang.RuntimePermission
      "org.jboss.security.SecurityAssociation.setServer";
   permission java.lang.RuntimePermission
      "org.jboss.security.SecurityAssociation.setRunAsRole";
   permission java.lang.RuntimePermission "loadLibrary.tcnative-1";
   permission java.lang.RuntimePermission "loadLibrary.libtcnative-1";

   permission java.net.NetPermission "specifyStreamHandler";

   permission java.util.PropertyPermission "*", "read,write";
   permission java.security.SecurityPermission
      "getProperty.package.definition";
   permission java.security.SecurityPermission
      "setProperty.package.definition";
   permission java.security.SecurityPermission
      "getProperty.package.access";
   permission java.security.SecurityPermission
      "setProperty.package.access";
   permission java.security.SecurityPermission "setPolicy";
   permission java.security.SecurityPermission
      "putProviderProperty.JBossSX";
   permission java.security.SecurityPermission "insertProvider.JBossSX";

   permission java.lang.reflect.ReflectPermission "suppressAccessChecks";

   permission java.net.SocketPermission "*:1024-", "accept,listen";
   permission java.util.logging.LoggingPermission "control";

   permission javax.security.auth.AuthPermission "doAsPrivileged";
   permission javax.security.auth.AuthPermission "modifyPrincipals";

   permission javax.security.auth.PrivateCredentialPermission
      "javax.resource.spi.security.PasswordCredential * \"*\"", "read";
```

```
   permission javax.security.auth.PrivateCredentialPermission
      "javax.crypto.spec.SecretKeySpec * \"*\"", "read";
   permission javax.security.auth.PrivateCredentialPermission
      "org.jboss.security.srp.SRPParameters * \"*\"", "read";

   permission java.security.SecurityPermission "getPolicy";
   permission java.lang.RuntimePermission "accessClassInPackage.*";
   permission java.lang.RuntimePermission "getClassLoader";
   permission java.lang.RuntimePermission "getProtectionDomain";
   permission java.lang.RuntimePermission
      "org.jboss.security.SecurityAssociation.getSubject";

   permission javax.security.auth.AuthPermission "createLoginContext.*";
   permission javax.security.auth.AuthPermission "getLoginConfiguration";

   permission java.net.SocketPermission "*", "connect,accept,resolve";
   permission org.jboss.naming.JndiPermission "JAXR",
      "bind,rebind,unbind,lookup,list,listBindings,createSubcontext";
};

//****************** End of Section 3 *********************

//************************************************************
//
//   Section 4: JBoss EAP Testsuite Permissions
//
//   This section is just for test suite purpose and can
//   safely removed.
//   General recomendation: This section should be deleted or
//   commented out in production.
//************************************************************

// Testing configuration lib directory permissions
grant codeBase "file:${user.dir}/../server/cc/lib/-" {
  permission java.security.AllPermission;
};

// Permissions for the WarPermissionsUnitTestCase
// Permissions for crypto tests (putProvider)
grant codeBase "file:${jboss.test.deploy.dir}/-" {
   permission java.util.PropertyPermission "*", "read";
   permission java.io.FilePermission "<<ALL FILES>>", "read,write,delete";
   permission java.security.SecurityPermission
      "putProviderProperty.JBossSX";
   permission org.jboss.naming.JndiPermission "<<ALL BINDINGS>>",
      "bind,rebind,unbind,lookup,list,listBindings,createSubcontext";
};

// Following JDBC driver is included just for CC test purpose.
// When you test with different JDBC driver than Oracle DB you have to
// create your own entries.
grant codeBase "file:${jboss.server.home.dir}/lib/ojdbc14.jar" {
   // change host name and port to one where your database resides.
   permission java.net.SocketPermission
      "dev68.qa.atl2.redhat.com:1521", "connect";
```

```
   permission java.util.PropertyPermission
      "oracle.net.wallet_location", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.TcpNoDelay", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.defaultNChar", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.useFetchSizeWithLongColumn", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.convertNcharLiterals", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.V8Compatible", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.J2EE13Compliant", "read";
   permission java.util.PropertyPermission
      "oracle.jdbc.FastConnectionFailover", "read";
   permission java.util.PropertyPermission "oracle.net.tns_admin", "read";
   permission java.util.PropertyPermission "line.separator", "read";
   permission java.util.PropertyPermission "user.name", "read";
   permission java.util.PropertyPermission "java.version", "read";

   permission java.lang.RuntimePermission
      "accessClassInPackage.sun.jdbc.odbc";
   permission java.net.SocketPermission "*", "resolve";

};

//******************* End of Section 4 **********************


//************************************************************
//
// Section 5: User Applications Permissions
//
// This sections is for user application permissions.
// Can be modified with care and attention to previously
// entered permissions.
//************************************************************

//  Following lines are here as template for creating JDBC driver
//  permissions entry specific for your database. If using Oracle, one can
//  copy JDBC driver permissions from Section 4.
//grant codeBase "file:${jboss.server.home.dir}/lib/<your JDBC
driver>.jar"
//{
//   <grant necessary permissions>
//};

// Minimal permissions are allowed to everyone else
grant {
   permission java.lang.RuntimePermission "queuePrintJob";
};

//******************* End of Section 5 **********************
```

APPENDIX E. REVISION HISTORY

Revision 4.3.3-12.33.400 **2013-10-30** **Rüdiger Landmann**
Rebuild with publican 4.0.0

Revision 4.3.3-12.33 **July 24 2012** **Ruediger Landmann**
Rebuild for Publican 3.0

Revision 4.3.3-12 **Friday Dec 10 2010** **Jared Morgan**
Document was missing from docs.redhat.com. Minor changes to XML attributes, and staging attribute versions in `Book_Info.xml` to enable restaging. Note that no word content in this guide was altered in any way. This guide is unaltered under the requirements of Common Criteria.

Revision 1.0-1 **Mon Apr 20 2009** **Darrin Mison**
Published