JBoss Enterprise Application Platform 4.3.0 GA CP03 Security Target

Version: 2.3

Last Update: 2009-04-16

JBoss Enterprise Application Platform 4.3.0 GA CP03 Security Target

Version: 2.3

Last Update: 2009-04-16

Document History

Version	Date	Changes	Author
0.1	2008-01-25	Initial Version	Stephan Müller, atsec
0.2	2008-01-28	Move of contents of 1.4.3 to chapter 7 Enhancement of section 1.5 by a general application server description and the logical boundaries of JBoss Addition of FMT_MSA.3-JB	Stephan Müller, atsec
0.3	2008-02-04	Removal of most of the FMT SFRs, update of the Webservices and JNDI access control policy, definition of the modes of operation allowed for the TOE	Stephan Müller, atsec
0.4	2008-02-09	Minor clarifications	Stephan Müller, atsec
1.0	2008-03-18	Incorporation of evaluator comments; addition of containers part of the TOE; update of security problem definition	Stephan Müller, atsec
1.1	2008-03-18	Spelling updates Clarification of evaluated configuration Addition of JMX Access Control Policy	Stephan Müller, atsec
1.2	2008-04-02	Restore title format for FDP_ACF.1(4) Update of chapter 7 with JMX Access Control Policy	Alejandro Masino, atsec
1.3	2008-06-17	Update of FMT_MSA.3 Clarification of TOE version	Stephan Müller, atsec
1.4	2008-07-08	Removal of separate JNDI access control SFRs	Stephan Müller, atsec
1.5	2008-07-17	Update list of JVMs, editorial changes	Stephan Müller, atsec
1.6	2008-07-22	Editorial changes	Alejandro Masino, atsec
1.7	2008-07-28	BSI decision on FPT_STM.1: addition of another objective for the environment	Stephan Müller, atsec
1.8	2008-08-06	Addressing BSI comments	Stephan Müller, atsec
1.9	2008-12-23	Adding list of abbreviations Adding editorial changes Updating list of underlying JREs Updating list of allowed databases A.CLUSTER updated	Stephan Müller, atsec
2.0	2009-01-08	Removal of requirement to disable ports	Stephan Müller, atsec
2.1	2009-01-23	Update of dependency mappings, wording	Stephan Müller, atsec
2.2	2009-02-04	Update of typo in FDP_ACF.1.4(2)	Stephan Müller, atsec
2.3	2009-04-16	Inclusion of fix for CVE-2009-0027	Stephan Müller, atsec

© Red Hat, atsec 2009

atsec is a trademark of atsec GmbH

Red Hat and the Red Hat logo are trademarks or registered trademarks of Red Hat, Inc. in the United States, other countries, or both.

Intel, Xeon, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based products are trademarks of Sun Microsystems, Inc., in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group in the United States and other countries.

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Copyright © 2008 by atsec information security corporation, and Red Hat or its wholly owned subsidiaries.

Table of Content

© Red Hat, atsec 2009

© Red Hat, atsec 2009

JBoss Enterprise Application Platform 4.3.0 GA CP03 Security Target

Version: 2.3

Last Update: 2009-04-16

atsec is a trademark of atsec GmbH

Red Hat and the Red Hat logo are trademarks or registered trademarks of Red Hat, Inc. in the United States, other countries, or both.

Intel, Xeon, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based products are trademarks of Sun Microsystems, Inc., in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group in the United States and other countries.

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Copyright © 2008 by atsec information security corporation, and Red Hat or its wholly owned subsidiaries.

Table of Content

© Red Hat, atsec 2009

© Red Hat, atsec 2009

References

[CC] Common Criteria for Information Technology Security Evaluation, CCIMB-2007-09-001 to CCIMB-2007-09-003, Version 3.1 Revision 2, September 2007, Part 1 to 3

[CEM] Common Methodology for Information Technology Security Evaluation, CCIMB-2007-09-004, Evaluation Methodology, Version 3.1 Revision 2, September 2007

[GUIDE] ISO/IEC PDTR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, ISO/IEC JTC 1/SC 27 N 2449, 2000-01-04

© Red Hat, atsec 2009

Subject:	See Principal (similar information found in a Principal object for a user can be kept in a Subject object).
Target of Evaluation (TOE):	The TOE is defined as the JBoss application server, running and tested on the hardware, operating systems and Java virtual machines specified in this Security Target.
User:	Any individual/person who has a unique user identifier and who interacts with the JBoss product. Unauthorized users do not possess a valid user identifier.
User Security Attributes:	Defined by functional requirement FIA_ATD.1, every user is associated with a number of security attributes which allow the TOE to enforce its security functions on this user.

1.3 ST Reference and TOE Reference

Title: JBoss Enterprise Application Platform 4.3.0 GA CP03 with the additional patch jbeap-4.3.0.GA_CP03_CVE-2009-0027.

Version: 2.3

Authors: Stephan Müller

Publication Date: 2009-04-16

Keywords: JBoss, Application Server.

1.4 TOE Overview

The TOE is the JBoss Enterprise Application Platform which implements an application server. JBoss is based on Java Enterprise Edition (J2EE) and therefore supports a large variety of operating systems. As an application server, JBoss allows client computers or devices to access applications. Access to these applications is possible through different network protocols, such as HTTP, EJBs, and others. JBoss handles the business logic of the application, including accessing and providing the user data required by the application.

The TOE is defined as a stand-alone JBoss instance. If a cluster of JBoss nodes is defined, then the entire cluster is considered to be one TOE.

1.4.1 TOE Type

JBoss is a Java-based application server which provides many advanced product features, including clustering, failover, load balancing, and Enterprise JavaBeans version 3.

1.4.2 Intended Method of Use

The TOE is intended to operate in a networked environment with other instantiations of the TOE as well as other well-behaved client systems operating within the same management domain. All those systems need to be configured in accordance with a defined common security policy. Communication links between individual instances of the TOE can be protected against loss of confidentiality and integrity using separate physical networks or by cryptographic protection mechanisms supported by the TOE.

The data under the control of the TOE is stored in named objects, and the TOE can associate with each named object a description of the access rights to that object.

1.4.3 Major Security Features

The primary security features of the TOE are:

- Access Control covering the objects of URLs, EJB methods, message queues and topics
- Audit covering the access control decisions
- Clustering ensuring the consistency of user and TSF data between cluster nodes
- Identification and Authentication ensuring the proper identification and authentication of users to facilitate the various access control mechanisms
- Transaction Rollback ensuring data consistency for user and TSF data

© Red Hat, atsec 2009

References

[CC] Common Criteria for Information Technology Security Evaluation, CCIMB-2007-09-001 to CCIMB-2007-09-003, Version 3.1 Revision 2, September 2007, Part 1 to 3

[CEM] Common Methodology for Information Technology Security Evaluation, CCIMB-2007-09-004, Evaluation Methodology, Version 3.1 Revision 2, September 2007

[GUIDE] ISO/IEC PDTR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, ISO/IEC JTC 1/SC 27 N 2449, 2000-01-04

© Red Hat, atsec 2009

1 ST Introduction

This Security Target documents the security characteristics of the JBoss Enterprise Application Platform 4.3.0 GA CP03 (in the rest of this document the term "JBoss" is used as a synonym for this TOE).

The TOE does not include the hardware, firmware, operating system or Java virtual machine used to run the software components.

1.1 ST Structure

The structure of this document is as defined by [CC] Part 1 Annex A.

- Section 1 is the TOE Overview Description.
- Section 2 provides the conformance claims.
- Section 3 provides the Security Problem Definition
- Section 4 provides the security objectives
- Section 5 provides the extended components definition
- Section 6 provides the security requirements
- Section 7 provides the TOE summary specifications

1.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Administrative User:	This term refers to a user in one of the defined administrative roles of a JBoss system. The TOE defines a set of administrative roles where each role has specific administrative authorities. Splitting the administrative authorities among different roles allows for a more controlled operational environment without the need for a single user to have all administrative authorities.
Authentication data:	This includes the password and X.509 certificates for each user of the product. Authentication mechanisms using other authentication data are not supported in the evaluated configuration.
JBoss Container	A JBoss container, or in short container, is a part of JBoss that provides services to user-written programs. For example, the EJB functionality is implemented by the EJB container, the JSP/servlet functionality is implemented by the Tomcat container. The JBoss architecture implements various functional aspects as self-contained containers which can be selectively enabled.
Data:	Arbitrary bit sequences in computer memory or on storage media.
Group:	After a user is successfully identified and authenticated, JBoss instantiates a "Group" Java object containing the groups the authenticated subject is associated with.
Information:	Any data held within a JBoss instance, including data in transit between systems.
Named Object:	In JBoss, those objects that are subject to access control. This includes all objects except memory objects. Please note, named objects are not to be mixed with the implementation of Java objects.
Object:	For JBoss, objects are defined by the different iterations of FDP_ACC.1.
Principal:	After a user is successfully identified and authenticated, JBoss instantiates a "Principal" Java object as a token for the user. This object contains multiple information, including the users identityand the roles associated with the user. A Principal is an authenticated user requesting services from JBoss.
Product:	The term product is used to define software components that comprise the JBoss application server.
Role:	A role represents a set of actions that an authorized user, upon assuming the role, is allowed to perform.

© Red Hat, atsec 2009

Subject:	See Principal (similar information found in a Principal object for a user can be kept in a Subject object).
Target of Evaluation (TOE):	The TOE is defined as the JBoss application server, running and tested on the hardware, operating systems and Java virtual machines specified in this Security Target.
User:	Any individual/person who has a unique user identifier and who interacts with the JBoss product. Unauthorized users do not possess a valid user identifier.
User Security Attributes:	Defined by functional requirement FIA_ATD.1, every user is associated with a number of security attributes which allow the TOE to enforce its security functions on this user.

1.3 ST Reference and TOE Reference

Title: JBoss Enterprise Application Platform 4.3.0 GA CP03 with the additional patch jbeap-4.3.0.GA_CP03_CVE-2009-0027.

Version: 2.3

Authors: Stephan Müller

Publication Date: 2009-04-16

Keywords: JBoss, Application Server.

1.4 TOE Overview

The TOE is the JBoss Enterprise Application Platform which implements an application server. JBoss is based on Java Enterprise Edition (J2EE) and therefore supports a large variety of operating systems. As an application server, JBoss allows client computers or devices to access applications. Access to these applications is possible through different network protocols, such as HTTP, EJBs, and others. JBoss handles the business logic of the application, including accessing and providing the user data required by the application.

The TOE is defined as a stand-alone JBoss instance. If a cluster of JBoss nodes is defined, then the entire cluster is considered to be one TOE.

1.4.1 TOE Type

JBoss is a Java-based application server which provides many advanced product features, including clustering, failover, load balancing, and Enterprise JavaBeans version 3.

1.4.2 Intended Method of Use

The TOE is intended to operate in a networked environment with other instantiations of the TOE as well as other well-behaved client systems operating within the same management domain. All those systems need to be configured in accordance with a defined common security policy. Communication links between individual instances of the TOE can be protected against loss of confidentiality and integrity using separate physical networks or by cryptographic protection mechanisms supported by the TOE.

The data under the control of the TOE is stored in named objects, and the TOE can associate with each named object a description of the access rights to that object.

1.4.3 Major Security Features

The primary security features of the TOE are:

- Access Control covering the objects of URLs, EJB methods, message queues and topics
- Audit covering the access control decisions
- Clustering ensuring the consistency of user and TSF data between cluster nodes
- Identification and Authentication ensuring the proper identification and authentication of users to facilitate the various access control mechanisms
- Transaction Rollback ensuring data consistency for user and TSF data

© Red Hat, atsec 2009

These primary security features are supported by the appropriate use of domain separation and reference mediation provided by the Java virtual machine if the Java Security Manager is utilized and the underlying operating system, which ensure that the security features are always invoked and cannot be bypassed, and that the TOE can protect itself.

1.4.4 Configurations

The evaluated configurations are defined as follows.

- The JMX Console (implemented in jmx-console.war) allowing users to access the JMX microkernel to perform administrative tasks must be protected against the use by all users not being trusted administrators. The protection can be achieved by either restricting access to the webfrontend using the HTTP access control facility provided by the TOE or by completely removing the console from the TOE.
- The Web Console (implemented in web-console.war) provides another web-based access into the JMX microkernel. It therefore has to be protected the same way as the JMX Console.
- The http-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 http-invoker.sar/invoker.war/WEB-INF/web.xml descriptor.
- The jmx-invoker-adaptor-server.sar 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.

The following relational databases are allowed to be used with the TOE (the listed databases are part of the IT environment and therefore not covered with security claims in this Security Target):

- Oracle 10g R2
- Oracle 9i
- Microsoft SQL Server 2005
- MySQL v5.0
- PostgreSQL v8.1
- IBM DB2 UDB 9.1
- IBM DB2 UDB 8.2

The internal database (HSQL DB) is not supported in the evaluated configuration.

1.4.4.1 Underlying software

The TOE can be executed on the following Java virtual machines which are part of the IT environment:

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

As the TOE functionality only relies on the correct operation of the Java virtual machine, the TOE can be executed 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 to execute the TOE.

1.4.4.2 TOE Environment

Several TOE systems may be interlinked in a network, and individual networks may be joined by bridges and/or routers. Each of the TOE systems implements its own security policy. The TOE does not include any synchronization function for those policies. As a result a single user may have user accounts on each of those

© Red Hat, atsec 2009

systems with different user IDs. This statement applies only to inter-TOE consistency as one TOE instance (either the standalone system or the cluster configuration of the TOE) ensures its internal data consistency.

If other systems are connected to a network they need to be configured and managed by the same authority using an appropriate security policy that does not conflict with the security policy of the TOE.

1.5 TOE Description

1.5.1 Application Server definition

The TOE representing an application server is implemented as an application, which allows users to access applications over various network protocols. JBoss executes Java applications which are registered and are executed by the application server.

JBoss is written entirely in Java and provides a J2EE-compliant environment which is consistent with the J2EE 1.4 specification as defined by SUN Microsystems. Depending on the configuration of the JBoss server, components required by the J2EE specification can be disabled. The applications developed for and served by JBoss are to be written in Java. Developers of the Java application implement the business logic and are free to utilize the supporting functionality of J2EE.

Illustration 1 documents the structure of JBoss. The JMX microcontainer provides the environment for the execution of different containers which allow applications to utilize services provided by these containers. The configuration of JBoss allows selectively enabling or disabling every container. The distribution of JBoss provides a number of containers that can be utilized, but additional containers may be implemented by third parties. The evaluated configuration defines the containers which are covered by the evaluation and therefore may be enabled in a CC-compliant configuration.

As part of the J2EE framework implemented by JBoss, applications can provide their logic to remote clients through the following network protocols:

- HTTP protocol: Java servlets provide their functionality based on URLs requested by the client.
- Enterprise Java Beans (EJB): Java classes can be made accessible to remote clients by allowing these clients to access EJB classes and their methods.

In addition to these protocols that can be used to access the business logic of an application, various other protocols may be made accessible by the application server to support the application's functionality – these protocols are provided by different JBoss containers and are unavailable if the containers are disabled. Such additional protocols might be the following:

- A message queue protocol may be provided as a reliable and possibly asynchronous communication channel. Such message queues may be used for the communication between different parts of distributed applications where different parts of an application are implemented in different instances of the application server. In addition, message queues may be used for the application to client communication.
- A JNDI name resolution service may be provided by the application server to allow different parts of an application or the client to resolve EJB classes and methods.

1.5.2 JBoss Application Server structure

JBoss Enterprise Application Platform implements a system for innovative and scalable Java applications. It includes open source technologies for building, deploying, and hosting enterprise Java applications and services.

JBoss Enterprise Application Platform balances innovation with enterprise class stability by integrating the most popular clustered Java EE application server with next generation application frameworks. Built on open standards, JBoss Enterprise Application Platform integrates various containers implementing the J2EE functionality, and other containers providing mechanisms to applications which go beyond the J2EE standard into a complete, simple enterprise solution for Java applications.

The J2EE specification considers the following four layers, also called tiers. Applications utilizing the J2EE specification may implement any combination of these tiers. In addition to listing the tiers, the following table specifies which tiers can be implemented and executed using the framework of JBoss.

Table 1 J2EE tier listing and JBoss coverage

J2EE tier	JBoss coverage
Client tier The client tier is the layer of the application executed	The applet may be stored on the JBoss server in order for the client to automatically download it when accessing a web page served by JBoss.

© Red Hat, atsec 2009

J2EE tier	JBoss coverage
on the client system in order to display the information provided by the application server. The client tier can be implemented by: • An applet executed by the client's browser • A stand-alone Java application executed by the client's Java Virtual Machine	However, neither the applet nor the application is executed by the JBoss application server, but they are executed by the Java Virtual Machine of the client system accessing the JBoss information remotely. Therefore, the client tier is considered to be not covered by JBoss.
Web tier The web tier is the presentation layer of the application server. It gathers the business information from the lower EJB tier and converts it to be presented as web pages. The web tier therefore does not implement any business logic as it can be considered an information converter from the application-internal data representation to a user-viewable and user-interpretable presentation. Considering a web-shopping application, the web tier implements the presenting layer with functionality such as the web pages showing the sold products or the display of the contents of the user's shopping cart.	The web tier can be implemented using Java servlets executing within the JBoss framework. The web tier is implemented by the customer-developed application.
Enterprise Java Beans (EJB) tier The EJB tier implements the business logic of the entire application. Business logic is considered to be the functionality implementing the information flow consistent with the purpose of the application. Considering a web-shopping application, the EJB tier implements business logic, such as the management and maintenance of the sold products, the shopping cart for each user.	The EJB tier can be implemented using various types of EJBs executing within the JBoss framework. The EJB tier is implemented by the customer-developed application.
Enterprise Information System's tier The enterprise information system's tier provides the logic to allow the EJB tier to access external data stores. This tier therefore covers database access mechanisms, such as a JDBC driver.	The enterprise information system's tier is provided by the TOE allowing the application's EJBs to access relational databases listed in section 1.4.4 via JDBC. The enterprise information system's tier is implemented by the TOE.

Fundamentally in the JBoss architecture, the JMX MBean server acts as the microkernel or microcontainer managing set of pluggable component services – the MBeans or services. This allows assembling different configurations and provides the flexibility to tailor the configurations to meet specific requirements.

The administrator does not have to run a large, monolithic server all the time; as the components not needed (which can also reduce the server startup time considerably) can be removed. Also additional services can be integrated into JBoss by writing new own MBeans (the evaluated configuration however prohibits the use of additional MBeans).

The following illustration depicts the interoperation of the different components of JBoss. The above mentioned components or services that can be enabled or disabled individually for the JBoss runtime are the J2EE 1.4 services and the services beyond J2EE. The following description applies to the illustration:

- The Hardware together with the operating system executes the Java Virtual machine which in turn executes the JMX microkernel. This microkernel provides the foundation on which all JBoss containers perform their tasks.
- Each container implements either a service as specified in J2EE 1.4 or a service providing additional functionality beyond J2EE 1.4.
- Applications execute as part of containers (such as the EJB container) and utilize services from other containers.

© Red Hat, atsec 2009

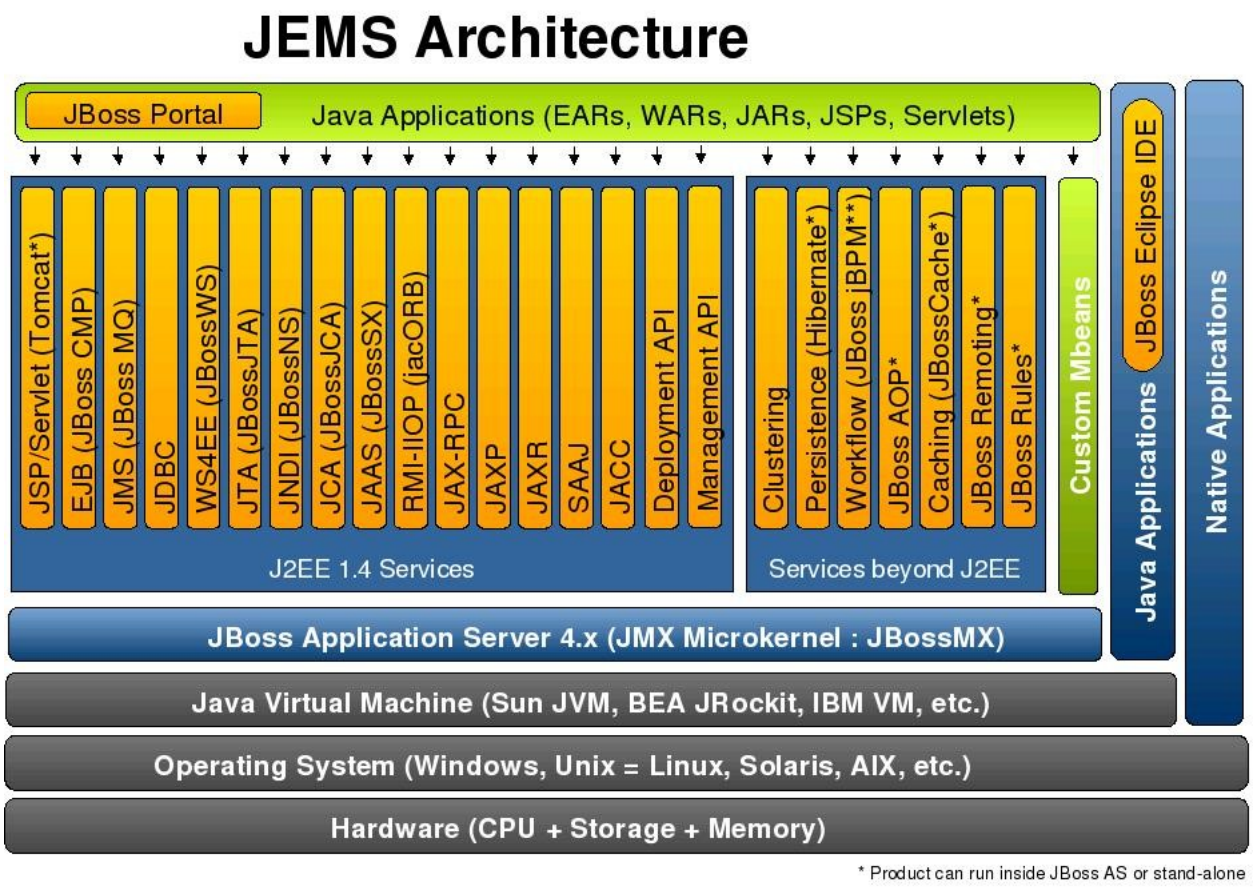

Illustration 1 JBoss components

The TOE allows the interaction with users through the following services:

- HTTP web network protocol
- Webservices
- Enterprise Java Beans (EJB)
- Java Messaging Service (JMS)
- Java Naming and Directory Interface (JNDI)

Applications utilize the services provided by the different containers by accessing the API exported by each container. These applications are loaded and executed by either the JSP/Servlet container, EJB container or the WS4EE container. If the administrator wants to achieve a technical separation between the applications and the TOE they run on, the Java Security Manager must be enabled as documented in section 1.5.3.3.

1.5.3 Definition of the TOE boundaries

1.5.3.1 Logical boundary

Please see the description of the security functionality in chapter 7.

1.5.3.2 Physical boundary

The TOE is the JBoss Enterprise Application Platform Version 4.3.0 GA CP03 with the additional patch jbeap-4.3.0.GA_CP03_CVE-2009-0027. Based on the above shown illustration, the TOE consists of the JMX microkernel and the containers specified below.

The TOE of JBoss allows the use of the following containers in the evaluated configuration (the containers are orange boxes shown in the picture above – the illustration above is provided for a better understanding only and the following list of components is authorative, regardless what the illustration shows):

- JBoss Application Server

© Red Hat, atsec 2009

- Hibernate
- Hibernate Entity Manager
- Hibernate Annotations
- JBoss Seam
- JBoss Web (embedded Tomcat 6.0)
- JBoss Cache
- JGroups (for Caching and Clustering)
- JBoss Messaging
- JBoss Transactions
- JBoss Web Services (JBossWS)
- JBossXB
- JBoss AOP
- JBoss Remoting
- JBoss Serialization
- JacORB

The TOE and its documentation (especially the CC configuration guide acting as the central guidance document covering the different aspects of the evaluated configuration of the TOE) are supplied via the Red Hat Network web site allowing a download of electronic copies of the TOE. Updates are also delivered through the Red Hat Network. The integrity and authenticity of the electronic copies are ensured by using cryptographic signatures.

1.5.3.3 Modes of operation

The evaluated configuration of the TOE allows the following two modes of operation which have an impact on how the TOE can protect itself against the behavior of applications. These modes utilize the Java Security Manager provided by the Java Virtual Machine as part of the TOE environment.

- Java Security Manager enabled: The Java Security Manager is utilized with a policy that completely protects the JBoss execution from any application utilizing the JBoss framework. The Security Manager together with its policy prohibits that any application can accidentally or intentionally interferes with the operation of JBoss.
- Java Security Manager disabled: The Java Security Manager is not used. This allows the application executed by JBoss to interfere with the execution of JBoss as the application runs in the same namespace as JBoss. Due to the assumption A.DEVEL, a developer of an application is considered to be trustworthy, not intentionally bypassing JBoss functionality. However, the nature of software implies that there is no proof that software is free of any bugs. This means that even a trustworthy developer has to assume that his application contains bugs. These bugs may provide a gate for attackers to bypass JBoss functionality. A user of an application containing bugs may find them and utilize these bugs to make the application behaving to violate JBoss functionality, including JBoss security functionality claimed in this ST.

  The evaluation of JBoss defined with this Security Target is a risk analysis on behalf of the administrator employing JBoss in its environment. The evaluation will show that when using JBoss as documented in the guidance documentation, the security functionality of JBoss is provided as defined in this ST. However, in the mode of operation without the Java Security Manager enabled, the administrator of JBoss has to perform a risk analysis on his own to ensure that the application does not contain bugs that may be abused by users of the application to circumvent the security functionality of JBoss.

The administrator of the TOE has to decide which mode of operation the TOE enforces. This selection of the mode of operation is to be done during the startup of the TOE.

Together with the TOE, the Security Manager policy that protects the TOE from any application is provided.

© Red Hat, atsec 2009

2 Conformance Claims

2.1 Common Criteria

The ST is [CC] Part 2 extended and Part 3 conformant.

2.2 Packages

The ST claims an Evaluation Assurance Level of EAL2 augmented by ALC_FLR.3.

2.3 Protection Profiles

This Security Target does not claim compliance to any protection profile.

- atsec confidential -
© Red Hat, atsec 2009

3 Security Problem Definition

3.1 Introduction

The Security Problem Definition describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be deployed.

To this end, the SPD identifies the list of assumptions made on the operational environment (including physical and procedural measures) and the intended method of use of the product, defines the threats that the product is designed to counter, and the organizational security policies with which the product is designed to comply.

3.2 Threats

The assumed security threats are listed below.

The **IT assets** to be protected comprise the information stored, processed or transmitted by the TOE. The term "information" is used here to refer to all data held within the application server, including data in transit between instances of the application server.

The TOE counters the general threat of unauthorized access to information, where "access" includes disclosure, modification and destruction.

The **threat agents** can be categorized as either:

- unauthorized users of the TOE, i.e. individuals who have not been granted the right to access the system; or
- authorized users of the TOE, i.e. individuals who have been granted the right to access the system.

The threat agents are assumed to originate from a well managed user community in a non-hostile working environment, and hence the product protects against threats of security vulnerabilities that might be exploited in the intended environment for the TOE with medium level of expertise and effort. The TOE protects against straightforward or intentional breach of TOE security by attackers possessing a basic attack potential.

3.2.1 Threats countered by the TOE

T.UAUSER An attacker (possibly, but not necessarily, an unauthorized user of the TOE) may impersonate an authorized user of the TOE. This includes the threat of an authorized user that tries to impersonate as another authorized user without knowing the authentication information.

T.ACCESS An authorized user may gain access to resources or perform operations for which no access rights have been granted.

T.DIFFER An authorized user may cause user data or TSF data that is stored in multiple places to become inconsistent and cause either user data loss or circumvention of TSF.

3.3 Organizational Security Policies

The TOE complies with the following organizational security policies:

P.ACCOUNTABILITY
The users of the system shall be held accountable for their actions within the system.

3.4 Assumptions

This section indicates the minimum physical and procedural measures required to maintain security of the JBoss.

3.4.1 Physical Aspects

A.PROTECT The hardware and software executing the TOE as well as the TOE software critical to security policy enforcement will be protected from unauthorized modification including unauthorized modifications by potentially hostile outsiders.

© Red Hat, atsec 2009

3.4.2 Personnel Aspects

A.ADMIN It is assumed that there are one or more competent individuals who are assigned to manage the TOE and the TOE environment and the security of the information it contains. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.DEVEL The developer of user applications executed by the TOE, including web server applications and enterprise beans, is trustworthy and will comply with all instructions set forth by the user guidance and evaluated configuration guidance of the TOE.

3.4.3 Connectivity Aspects

A.SYSTEM The operating system and the Java virtual machine operate according to their specification. These external systems are configured in accordance with the installation guidance and the evaluated configuration guidance of the TOE.

A.CLUSTER One or more TOE instances 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 communication to pass through originated outside the TOE network segment if the network protocol is TCP and has the following destination ports: 8080, 8443. All communication originating from one of the TOE instances is to be allowed.

A.PEER Any other system with which the TOE communicates is assumed to be under the same management control and operate under the same security policy constraints as the TOE itself.

© Red Hat, atsec 2009

4 Security Objectives

4.1 Security Objectives for the TOE

O.AUTHORIZATION The TOE must ensure that only identified and authorized users gain access to the TOE and its resources.

O.ACCESS The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.

O.AUDITING The TSF must record security relevant actions of users of the TOE and security relevant events. The information recorded with security relevant events must be in sufficient detail to help an administrator of the TOE detect attempted security violations or potential misconfiguration of the TOE security features that would leave the IT assets open to compromise.

O.CONSISTENCY The TSF must ensure the consistency of user data as well as TSF data while it is being processed. Consistency needs to be ensured when data is processed that may be located in multiple places.

4.2 Security Objectives for the TOE Environment

All security objectives listed in this section are targeted at the non-IT environment of the TOE.

OE.ADMIN Those responsible for the administration of the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

OE.SYSTEM Those responsible for the TOE must ensure that the operating system and the Java virtual machine are installed and configured in accordance with the guidance of the TOE and that these mechanisms operate as specified. This also covers that only the Java virtual machines enumerated in this ST are used as underlying platform to ensure that proper date and time information is available to the audit facility.

OE.INSTALL Those responsible for the TOE must establish and implement procedures to ensure that the software components that comprise the TOE are distributed, installed, configured and administered in a secure manner.

OE.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE critical to security policy as well as the underlying hardware and software are protected from physical attack which might compromise IT security objectives.

OE.RECOVER Those responsible for the TOE must ensure that procedures and/or mechanisms are provided to assure that, after system failure or other discontinuity, recovery without a protection (i.e., security) compromise is obtained.

OE.DEVEL Those responsible for the TOE shall ensure that the developers of the applications executed by the TOE are trustworthy and implement the applications in accordance with the guidance provided with the TOE. If the Java Security Manager is disabled, the developer together with the administrator has to perform a risk analysis of his application to ensure that there are no vulnerabilities in the application that allow attackers with an attack potential consistent with the EAL chosen for this Security Target to misuse the application to circumvent TOE functionality.

4.3 Security Objective Rationale

The following tables provide a mapping of security objectives to the environment defined by the threats, policies and assumptions, illustrating that each security objective covers at least one threat, assumption or policy and that each threat, assumption or policy is covered by at least one security objective.

4.3.1 Security Objectives Coverage

Table 2 Mapping Objectives to threats, assumptions and policies

Objective	Threat / Policy
O.AUTHORIZATION	T.UAUSER
O.ACCESS	T.ACCESS

© Red Hat, atsec 2009

O.AUDITING	P.ACCOUNTABILITY
O.CONSISTENCY	T.DIFFER

Table 4-2: Mapping objectives for the environment to threats, assumptions and policies

Env. Objective	Threat / Assumption / Policy
OE.ADMIN	A.ADMIN
OE.SYSTEM	A.SYSTEM
OE.INSTALL	A.ADMIN, A.CLUSTER, A.PEER
OE.PHYSICAL	A.PROTECT
OE.RECOVER	A.ADMIN
OE.DEVEL	A.DEVEL

4.3.2 Security Objectives Sufficiency

T.UAUSER: The threat of impersonization of an authorized user by an attacker is sufficiently diminished by O.AUTHORIZATION requiring proper authorization of users gaining access to the TOE. The access control attributes are protected by the environment to be accessible to the administrator only.

T.ACCESS: The threat of an authorized user of the TOE accessing information resources without the permission from the user responsible for the resource is removed by O.ACCESS requiring access control for resources and the ability for authorized users to specify the access to their resources. This ensures that a user can access a resource only if the requested type of access has been granted by the user responsible for the management of access rights to the resource.

T.DIFFER: The threat of user data and TSF data being inconsistent among different parts of the TOE is diminished by the functionality provided by O.CONSISTENCY requiring that a mechanism is enforced that ensures the consistency of the data.

P.ACCOUNTABILITY: The policy to provide a means to hold users accountable for their activities is implemented by O.AUDITING providing the TOE with such functionality.

A.PROTECT: The assumption on physical protection of all hard- and software as well as the network and peripheral cabling is covered by the objectives OE.PHYSICAL requiring physical protection.

A.ADMIN: The assumption on competent administrators is covered by OE.ADMIN requiring competent and trustworthy administrators and OE.INSTALL requiring procedures for secure distribution, installation and configuration of systems as well as OE.RECOVER requiring the administrator to perform all the required actions to bring the TOE into a secure state after a system failure or discontinuity..

A.DEVEL: The assumption on developers of applications executed by the TOE to be trustworthy and to comply with the instructions set forth in the guidance is covered by OE.DEVEL requiring the administrator to ensure that these developers are indeed trustworthy.

A.SYSTEM: The assumption that the environment the TOE relies on to enforce its functionality (the OS and the Java virtual machine) is configured according to the guidance provided by the TOE is covered by OE.SYSTEM requiring the administrator to comply with that guidance.

A.CLUSTER: The assumption that the cluster network is physically protected is covered by OE.INSTALL requiring the administrator to install the TOE in a secure manner.

A.PEER: The assumption on the same management control and security policy constraints for systems with which the TOE communicates is covered by OE.INSTALL requiring procedures for secure distribution, installation and configuration of the networked system.

© Red Hat, atsec 2009

5 Extended Components Definition

5.1 FDP_ROL.2-JB

The Security Target defines the extended component FDP_ROL.2-JB as part of the FDP_ROL family in CC Part 2 for usage within this ST.

5.1.1 Component leveling

Automated rollback addresses the need to roll back or undo all operations within the defined bounds.

5.1.2 Management: FDP_ROL.2-JB

The following action could be considered for the management functions in FMT:

- The boundary limit to which rollback may be performed could be configurable item within the TOE.

5.1.3 Audit: FDP_ROL.2-JB

Please see the audit information on the FDP_ROL family in CC Part 2.

5.1.4 FDP_ROL.2-JB Automated rollback

Hierarchical to: No other components

Dependencies: No dependencies

FDP_ROL.2-JB.1 The TSF shall perform an automated rollback of all the operations [assignment: *list of sub-operations belonging to one operation*] when [assignment: *list of causes for a rollback of all operations*].

5.2 FMT_MSA.3-JB

The Security Target defines the extended component FMT_MSA.3-JB as part of the FMT_MSA.3 family in CC Part 2 for usage within this ST.

The reason for specifying this SFR is that the TOE implements an access control mechanism which has defined default values. These default values can be defined but not modified at runtime of the TOE. As the access control rules are specified in a configuration file that is stored in the IT environment without the possibility to configure the default values through the TSF, the FMT_MSA.3.2 part of FMT_MSA.3 is not covered by the TOE. However, this part is fulfilled by the IT environment, specifically by the underlying OS, as only users with an account to the OS and appropriate OS access rights can modify these default values.

5.2.1 Component leveling

See FMT_MSA.3.

5.2.2 Management: FMT_MSA.3-JB

None

5.2.3 Audit: FMT_MSA.3-JB

None

5.2.4 FMT_MSA.3-JB Unmodifiable static attribute initialization

Hierarchical to: No other components

© Red Hat, atsec 2009

Dependencies: None

FMT_MSA.3-JB.1	The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.

© Red Hat, atsec 2009

6 Security Requirements

6.1 TOE Security Functional Requirements

All operations except iterations are marked in **bold** within each of the requirements. Iterations are marked by an extension of the SFR definition.

6.1.1 Security Audit (FAU)

6.1.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the **not specified** level of audit; and

c) Each access request for each access control policy;

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **no additional information.**

Application Note: The subject identity is defined by container and thread ID.

6.1.1.2 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.2 User Data Protection (FDP)

6.1.2.1 HTTP Access Control Policy (FDP_ACC.1) (1)

FDP_ACC.1.1 The TSF shall enforce the **HTTP Access Control Policy** on
Subject: a user represented by a Principal or Subject object assigned to a specific role represented by the Group object
Object: data accessible at URL
Operations: all HTTP methods of GET, POST, PUT, TRACE, DELETE, HEAD

Application Note: Access control is managed with appropriate settings in the configuration file web.xml.

6.1.2.2 EJB Access Control Policy (FDP_ACC.1) (2)

FDP_ACC.1.1 The TSF shall enforce the **EJB Access Control Policy** on
Subject: a user represented by a Principal or Subject object assigned to a specific role represented by the Group object
Objects: EJB and associated method
Operations: calling the method of the EJB

Application Note: Access control is managed with appropriate settings in the configuration file ejb-jar.xml (EJB 2.x and EJB 3) and the "@RolesAllowed", "@DenyAll", "@PermitAll" Java Annotations in the Java source code of the affected EJB (EJB 3).

6.1.2.3 JMS Access Control Policy (FDP_ACC.1) (3)

FDP_ACC.1.1 The TSF shall enforce the **JMS Access Control Policy** on
Subject: a user represented by a Principal or Subject object assigned to a specific role represented by the Group object

© Red Hat, atsec 2009

Objects: message queue, topic
Operations: read, write, create operations on a message queue or topic

Application Note: Access control is managed with appropriate settings in the configuration files matching messaging-service.xml (for the global default values applicable to destinations without specific security configurations) and destinations-service.xml (for individual message queue or topic destination configurations overriding the global default values).

Application Note: Message queues and topics are communication facilities allowing different subject to exchange information.

6.1.2.4 Webservices Access Control Policy (FDP_ACC.1) (4)

FDP_ACC.1.1 The TSF shall enforce the **Webservices Access Control Policy** on
Subject: a user represented by a Principal or Subject object assigned to a specific role represented by the Group object
Objects: Plain old Java Objects (POJOs) (deployed as Servlets) and Session Beans
Operations: all HTTP methods of GET, POST, PUT, TRACE, DELETE, HEAD; calling the method of the EJB

Application Note: Access control is managed with appropriate settings in the configuration files web.xml (servlets), ejb-jar.xml (EJB 2.x and EJB 3) and the "@RolesAllowed", "@DenyAll", "@PermitAll" Java Annotations in the Java source code of the affected EJB (EJB 3).

6.1.2.5 JMX Invokers Access Control Policy (FDP_ACC.1) (6)

FDP_ACC.1.1 The TSF shall enforce the **JMX** Invokers **Access Control Policy** on
Subjects: a user represented by a Principal or Subject object assigned to a specific role represented by the Group object
Objects: JMX Invokers
Operations: calling any method of the MBeanServer

Application Note: Access control is managed with appropriate settings in the configuration file jmx-invoker-service.xml.

6.1.2.6 HTTP Access Control Functions (FDP_ACF.1) (1)

FDP_ACF.1.1 The TSF shall enforce the **HTTP Access Control Policy** to objects based on the following:

a) Subject attributes: Roles

b) **Object attributes: URL, roles**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Access to the URL with the requested HTTP method is permitted if:

1. **the requesting user is associated with a role specified for the URL and HTTP method in the "security-constraint" element defined in the file web.xml;**
2. **the transport layer security used when accessing the URL must cover at least that security mechanism defined by the "user-data-constraint" element for the accessed URL, requiring either no protection, integrity protection or confidentiality protection.**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **permission rules marked with the "unchecked" element instead of the "role-name" element define that any authenticated user can access the URL.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **following rules: none.**

© Red Hat, atsec 2009

6.1.2.7 EJB Access Control Functions (FDP_ACF.1) (2)

FDP_ACF.1.1 The TSF shall enforce the **EJB Access Control Policy** to objects based on the following:

a) Subject attributes: Roles

b) **Object attributes: EJB name and associated method name, roles**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Access to the EJB method is permitted if the requesting user is associated with a role specified for the EJB method in the "method-permission" element defined in the file ejb-jar.xml.

For EJB3, the permission may also be specified with the "@Permissions" Java annotation.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **permission rules marked with the "unchecked" element instead of the "role-name" element define that any authenticated user can access the EJB method. For EJB3, classes, methods and constants marked with the "@Unchecked" Java annotation can be accessed by any authenticated users.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **following rules: methods marked with the "exclude-list" element are always denied access to. For EJB3, methods marked with the "@Exclude" Java annotation are always denied access to.**

6.1.2.8 JMS Access Control Functions (FDP_ACF.1) (3)

FDP_ACF.1.1 The TSF shall enforce the **JMS Access Control Policy** to objects based on the following:

a) Subject attributes: Roles

b) **Object attributes: message queue name, topic name, roles**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Access to the message queue or topic is permitted if the requesting user is associated with a role specified for the respective communication facility based on the following rules:

- **If the read attribute is true then that role will be able to read (create consumers, receive messages or browse) this destination.**
- **If the write attribute is true then that role will be able to write (create producers or send messages) to this destination.**
- **If the create attribute is true then that role will be able to create durable subscriptions on this destination.**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following rules: none.

6.1.2.9 Webservices Access Control Functions (FDP_ACF.1) (4)

FDP_ACF.1.1 The TSF shall enforce the **Webservices Access Control Policy** to objects based on the following:

a) Subject attributes: Roles

b) **Object attributes: URL, EJB name and associated method name, roles**

© Red Hat, atsec 2009

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Access to the URL with the requested HTTP method is permitted if:

1. **the requesting user is associated with a role specified for the URL and HTTP method in the "security-constraint" element defined in the file web.xml;**
2. **the role associated with the user calling the EJB method is permitted if the role is specified for the EJB method in the "method-permission" element defined in the file ejb-jar.xml;**
3. **for EJB3, the role associated with the user calling the EJB method is permitted if the role is specified for the EJB method with the "@Permissions" Java annotation.**
4. **The role associated with the user calling the EJB 3 classes is permitted if the role is specified for the EJB class with the Java Annotation "@RolesAllowed" in the Java source code of the EJB class;**
5. **The transport layer security used when accessing the URL or EJB method must cover at least that security mechanism defined by the "user-data-constraint" element for the accessed URL, requiring either no protection, integrity protection or confidentiality protection.**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **permission rules marked with the "unchecked" element instead of the "role-name" element define that:**

a) any authenticated user can access the URL

b) any authenticated user can access the EJB method (defined in ejb-jar.xml)

For EJB3, classes, methods and constants marked with the "@Unchecked" Java annotation can be accessed by any authenticated users.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **following rules: methods marked with the "exclude-list" element are always denied access to. For EJB3, methods marked with the "@Exclude" Java annotation are always denied access to.**

6.1.2.10 JMX Invokers Access Control Functions (FDP_ACF.1) (6)

FDP_ACF.1.1 The TSF shall enforce the **JMX Invokers Access Control Policy** to objects based on the following:

a) Subject attributes: Roles

b) **Object attributes: none (every MBeanServer method is unconditionally allowed if the subject is associated with the appropriate role)**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) **If the "org.jboss.jmx.connector.invoker.RolesAuthorization" class is configured for the interceptor class "org.jboss.jmx.connector.invoker.AuthorizationInterceptor":The access request to any MBeanServer method is permitted if the subject is associated with the "JBossAdmin" role.**

b) **If the "org.jboss.jmx.connector.invoker.ExternalizableRolesAuthorization" class is configured for the interceptor class "org.jboss.jmx.connector.invoker.AuthorizationInterceptor":The access request to any MBeanServer method is permitted if the subject is associated with the one of the roles specified in the "jmx-invoker-roles.properties".**

© Red Hat, atsec 2009

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **following rules: none.**

Application Note: The access control policy can only be enforced, if the JMX Invoker also utilizes JAAS to authenticate the calling user. Therefore, the administrator must configure in "jmx-invoker-service.xml" the "org.jboss.jmx.connector.invoker.AuthenticationInterceptor" interceptor class to protect the JMX console.

6.1.2.11 Automated rollback

FDP_ROL.2-JB.1 The TSF shall perform an automated rollback of all the operations **defined to form one transaction** when **at least one operation part of a transaction fails**.

6.1.3 Identification and Authentication (FIA)

6.1.3.1 User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) **Subject identity;**

b) **Role;**

c) Password, if the services of HTTP (basic, digest and form-based authentication), EJB, JMS, Webservice are available to the user;

d) X.509 Certificate if the certificate-based authentications services of HTTP, EJB, JMS, Webservice are available to the user.

6.1.3.2 Authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow **the following actions** on behalf of the user to be performed before the user is authenticated:

a) **All actions allowed by the access control mechanisms for the identity assigned to unauthenticated users with the element "DefaultUnauthenticatedPrincipal" configured for the JaasSecurityManagerService.**

b) All actions allowed by the access control mechanism to unsecured EJBs or EJB methods that are associated with the unchecked permission constraint for the identity assigned to unauthenticated users with the "unauthenticatedIdentity" element in the login module configuration.

c) All URLs (i) without a "security-constraint" element defined in the web.xml descriptor or (ii) without the "@RolesAllowed" and without the "@DenyAll" Java Annotations defined for EJB 3 servlets are accessible to unauthenticated users.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.3 Identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow **the following actions** on behalf of the user to be performed before the user is identified:

a) **All actions allowed by the access control mechanisms for the identity assigned to unauthenticated users with the element "DefaultUnauthenticatedPrincipal" configured for the JaasSecurityManagerService.**

© Red Hat, atsec 2009

b) All actions allowed by the access control mechanism to unsecured EJBs or EJB methods that are associated with the unchecked permission constraint for the identity assigned to unauthenticated users with the "unauthenticatedIdentity" element in the login module configuration.

c) **All URLs (i) without a "security-constraint" element defined in the web.xml descriptor or (ii) without the "@RolesAllowed" and without the "@DenyAll" Java Annotations defined for EJB 3 servlets are accessible to unauthenticated users.**

FIA_UID1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.4 User-Subject Binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

a) **Subject identity associated with auditable events;**

b) **Role the user is operating with.**

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of a user:

a) **Upon successful identification and authentication, the user identity shall be that specified in the user entry for the user that has authenticated.**

b) **The role associated with a subject shall be one of the authorized roles assigned to the user.**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of a user: **User security attributes are not to be changed after being assigned to a subject.**

6.1.4 Security Management (FMT)

6.1.4.1 Static Attribute Initialization (FMT_MSA.3) (1)

FMT_MSA.3.1 The TSF shall enforce the **HTTP Access Control Policy, Webservices Access Control Policy** to provide **permissive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **nobody** to specify alternative initial values to override the default values when an object or information is created.

Application Note: This SFR has been added to satisfy the dependency from FDP_ACF.1(1). The ST author is aware that the SFR does not contribute to the enhancement of security, but it also does not decrease security. The TOE does not require special handling as defined by FMT_MSA.3 in CC part 2. Therefore, the specification of this SFR only shows that the dependency can be fulfilled in line with the CC part 2 requirements.

6.1.4.2 Static Attribute Initialization (FMT_MSA.3) (2)

FMT_MSA.3.1 The TSF shall enforce the **EJB Access Control Policy, and JMX Invoker Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the **respective** SFP.

FMT_MSA.3.2 The TSF shall allow **nobody** to specify alternative initial values to override the default values when an object or information is created.

Application Note: All mentioned access control policies are configured with XML descriptor files which are edited with tools provided by the TOE environment (such as an editor). The TOE itself does not provide any facilities to change these files.

© Red Hat, atsec 2009

6.1.4.3 Unmodifiable Static Attribute Initialization (FMT_MSA.3-JB)

FMT_MSA.3-JB.1 The TSF shall enforce the **JMS Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

Application Note: The default behavior for message queue or topic destinations is defined with the JBoss Messaging service deployment descriptor in messaging-service.xml by using the element “attribute” with the name “DefaultSecurityConfig” which has the same structure as defined in FDP_ACC.1(3) and FDP_ACF.1(3).

6.1.5 Protection of the TSF (FPT)

6.1.5.1 Internal TSF consistency (FPT_TRC.1)

FPT_TRC.1.1 The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

FPT_TRC.1.2 When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for:

a) Identification and authentication

b) Access control

c) Execution of operations for:

I. HTTP requests

II. EJB requests

III. JMS requests

IV. Webservices requests

6.2 Security Requirements Rationale

This section provides the rationale for the internal consistency and completeness of the security functional requirements defined in this Security Target.

6.2.1 Internal Consistency of Requirements

This section describes the mutual support and internal consistency of the components selected for this Security Target. These properties are discussed for both functional and assurance components.

The functional requirements were selected from thosedefined in CC part 2 of the Common Criteria. Refinements of functional requirements were conducted in accordance with CC guidelines. The explicitly defined SFRs were created which complies with the constructs found in CC part 2.

Multiple instantiation of identical or hierarchically-related components was used to clearly state the required functionality that exists in this Security Target.

For internal consistency of the requirements we provide the following rationale:

Access Control policies

The different iterations of FDP_ACC.1 require the existence of a different access control for the different objects present in the TOE. The rules of these policies are described in the respective iterations of FDP_ACF.1. To be effective an access control mechanism requires users to be properly identified and authenticated (as required by FIA_UID.1 and FIA_UAU.1), proper binding of subjects to users (as required by FIA_USB.1). FMT_MSA.3(1), FMT_MSA.3(2), and FMT_MSA.3-JB define the default permissions for the different access control mechanisms.

Audit

FAU_GEN.1 defines the events that the TOE is required to be able to audit. Those events are related to other security functional requirements showing which event contributes to make users accountable for their actions with respect to the requirement. FAU_GEN.2 requires that the events are associated with the identity of the user that caused the event. Of course this can only be done if the user is known (which may not be the case for failed login attempts).

Clustering

© Red Hat, atsec 2009

FPT_TRC.1 defines the replication mechanism to keep different parts of the TOE (the different nodes of a cluster) consistent with each other. This SFR ensures that all TSF data, including that required for the other SFRs are maintained consistently between the cluster nodes.

Identification and Authentication

As stated above Identification and Authentication is required for useful access control policies based on the identity and roles of individual users. FIA_UAU.1 and FIA_UID.1 require that users are authenticated before they can perform actions on the TOE requiring the identity of the user. Since the TOE implements threads acting on behalf of the user, FIA_USB.1 ensures that those processes act within the limits defined for the user they are acting for (unless they are trusted to perform activities beyond the rights of the user). To allow the TOE to assign the proper identifiers to subjects acting on behalf of users, FIA_ATD.1 defines various security attributes for different users.

Transaction Rollback

FDP_ROL.2-JB.1 ensures that an automated rollback of failed transactions is performed by the TOE. If the TOE identifies that an operation belonging to a transaction fails, all operations already performed for the transaction are rolled back to the state as if these operations never happened.

6.2.2 Security Requirements Coverage

The following table shows that each security functional requirement addresses at least one objective.

Table 3 Mapping Security Functional Requirements to Objectives

SFR	Objectives
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FDP_ACC.1(1)	O.ACCESS
FDP_ACF.1(1)	O.ACCESS
FDP_ACC.1(2)	O.ACCESS
FDP_ACF.1(2)	O.ACCESS
FDP_ACC.1(3)	O.ACCESS
FDP_ACF.1(3)	O.ACCESS
FDP_ACC.1(4)	O.ACCESS
FDP_ACF.1(4)	O.ACCESS
FDP_ACC.1(6)	O.ACCESS
FDP_ACF.1(6)	O.ACCESS
FDP_ROL.2-JB	O.CONSISTENCY
FIA_ATD.1	O.AUTHORIZATION, O.ACCESS
FIA_UAU.1	O.AUTHORIZATION
FIA_UID.1	O.AUTHORIZATION
FIA_USB.1	O.AUTHORIZATION, O.ACCESS
FMT_MSA.3(1)	O.ACCESS, O.AUTHORIZATION
FMT_MSA.3(2)	O.ACCESS, O.AUTHORIZATION
FMT_MSA.3-JB	O.ACCESS, O.AUTHORIZATION
FPT_TRC.1	O.CONSISTENCY

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE have to use an identification and authentication process [FIA_UID.1, FIA_UAU.1]. To ensure authorized access to the TOE, authentication data is protected [FIA_ATD.1]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB.1]. Nobody is able to control which default values are configured for the access control mechanisms [all iterations of FMT_MSA.3].

O.ACCESS

The TSF must control access to resources based on identity of users.

The different access control must have a defined scope of control [all iterations of FDP_ACC.1]. The rules of the different access control mechanisms must be defined [all iterations of FDP_ACF.1]. The default values of the different access control mechanisms are defined [all iterations of FMT_MSA.3, FMT_MSA.3-JB]. The security attributes of subjects used to enforce the different access control mechanisms must be defined [FIA_ATD.1, FIA_USB.1].

O.AUDITING

© Red Hat, atsec 2009

The events to be audited must be defined [FAU_GEN.1], and must be associated with the identity of the user that caused the event [FAU_GEN.2].

O.CONSISTENCY

To ensure the consistency of user data, the TSF allows the definition of transactions where each operation of the transaction must succeed for the transaction to succeed or otherwise all operations already performed for the transaction are rolled back [FDP_ROL.2-JB]. In addition, to ensure the consistency of TSF data when held in multiple locations of different cluster nodes, the TSF implements a cluster communication that updates the TSF data in the appropriate cluster nodes when one node updates these TSF data [FPT_TRC.1].

6.2.3 Security Requirements Dependency Analysis

The following table shows the dependencies between the different security functional requirements and if they are resolved in this Security Target.

Table 4 Dependencies between Security Functional Requirements

Security Functional Requirement	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1 Reliable time stamps	No
FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	Yes
FDP_ACC.1(1)	FDP_ACF.1 Security attribute based access control	Yes, by FDP_ACF.1(1)
FDP_ACF.1(1)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes, by FDP_ACC.1(1), FMT_MSA.3(1)
FDP_ACC.1(2)	FDP_ACF.1 Security attribute based access control	Yes, by FDP_ACF.1(2)
FDP_ACF.1(2)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes, by FDP_ACC.1(2) , FMT_MSA.3(2)
FDP_ACC.1(3)	FDP_ACF.1 Security attribute based access control	Yes, by FDP_ACF.1(3)
FDP_ACF.1(3)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	No (covered by FDP_ACC.1(3), FMT_MSA.3-JB)
FDP_ACC.1(4)	FDP_ACF.1 Security attribute based access control	Yes, by FDP_ACF.1(4)
FDP_ACF.1(4)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes, by FDP_ACC.1(4) , FMT_MSA.3(1)
FDP_ACC.1(6)	FDP_ACF.1 Security attribute based access control	Yes, by FDP_ACF.1(6)
FDP_ACF.1(6)	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes, by FDP_ACC.1(6) , FMT_MSA.3(2)
FDP_ROL.2-JB	No dependencies	Yes
FIA_ATD.1	No dependencies	Yes
FIA_UAU.1	FIA_UID.1 Timing of identification	Yes
FIA_UID.1	No dependencies	Yes
FIA_USB.1	FIA_ATD.1 User attribute definition	Yes
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	No
FMT_MSA.3-JB	No dependencies	Yes
FPT_TRC.1	FPT_ITT.1: Basic internal TSF data transfer protection	No

© Red Hat, atsec 2009

Comment

The security functional requirement FAU_GEN.1 covering audit generation depends on FPT_STM.1 for gathering the date/time stamp for the audit records. This dependency is uncovered due to CC version 3.1 definition as this version of the CC does not support the definition of SFRs for the IT environment. The TOE relies on the operating system to provide the appropriate time stamp. Hence, due to the definitions of CC 3.1 which does not allow the specification of SFRs for the IT environment, this dependency is unresolved. The functionality of providing a time stamp is implemented by the underlying Java virtual machine as defined by OE.SYSTEM.

The security functional requirement FMT_MSA.3 covering the default values for the different access control policies has a dependency on FMT_MSA.1 and FMT_SMR.1. The TOE does not implement the management and the respective protection of the management of the access control mechanisms. The access control mechanisms are configured via XML files located in the environment. As the environment is protected against unauthorized access (A.PROTECT), only authorized administrators can access these files to manage the different access control mechanisms. Therefore, the intent of FMT_MSA.1 is covered with the setup of the TOE and its environment.

The security functional requirement FDP_ACF.1(3) covering the access control mechanism for JMS depends on FMT_MSA.3. As explained in the rationale for the definition of FMT_MSA.3-JB, the TOE does not provide the facility to modify the default values of the access control policy as they are stored in a configuration file. Administrators have to use facilties of the IT environment (such as an editor) to modify the default values in the configuration files. Therefore, the dependency of FDP_ACF.1(3) to FMT_MSA.3 is satisfied with FMT_MSA.3-JB.

The security functional requirement FPT_TRC.1 covering the cluster communication has a dependency on FPT_ITT.1. The TOE does not rely on the technical implementation of the protection of the data channels between different TOE instances as the network utilized for the cluster communication covered by FPT_TRC.1 is physically separated from any other network. In addition, the base operating system is configured to not permit any routing from any attached network into the physically separated network used for the cluster communication. Thus, the requirement of FPT_ITT.1 is covered with non-technical means.

There are no unresolved dependencies between security assurance requirements. This is because the evaluation assurance level EAL2 has been defined such that no unresolved dependencies exist. The additional assurance component ALC_FLR.3 has no dependencies and therefore there are no unresolved dependencies for assurance components.

6.3 TOE Security Assurance Requirements

The target evaluation assurance level for the product is EAL2 [CC] augmented by ALC_FLR.3, which is seen appropriate for a controlled environment where attackers only have a low attack potential.

© Red Hat, atsec 2009

7 TOE Summary Specification

The following section explains how the security functions are implemented by JBoss. The different TOE security functions cover the various SFR classes.

The primary security features of the TOE are:

- Access Control
- Audit
- Clustering
- Identification and Authentication
- Transaction Rollback

7.1 Access Control

Using access control, the TOE is able to restrict access for the following request types with the following access control mechanisms:

- HTTP: URLs and paths provided with URLs can be protected from access by subjects:
  - Obtain the names of the roles allowed to access the URL. The role names are determined by the "security-constraint" elements defined for the invoked URL and optionally the HTTP request method (one or more of the following: GET, POST, PUT, TRACE, DELETE, HEAD) as part of the HTTP descriptor file. In addition to the specification of the URL and HTTP request method, the access control mechanism can optionally require cryptographic protection of the user's connection (either none, integrity-protected, confidentiality-protected).
  - If no roles have been assigned, or the method is specified in an exclude-list element, then access to the URL is allowed. Otherwise, the doesUserHaveRole method is invoked on the JBossSX security manager by the security interceptor to see if the caller has one of the assigned role names. This method iterates through the role names and checks if the authenticated user's Subject Roles group contains a SimplePrincipal with the assigned role name. Access is allowed if any role name is a member of the Roles group. Access is denied if none of the role names are members.
- EJB: EJBs and associated method names can be protected from being called by subjects:
  - Obtain the names of the roles allowed to access the EJB method from the EJB container. The role names are determined by the "role-name" elements of all "method-permission" elements containing the invoked method as defined in the EJB descriptor file.
  - If no roles have been assigned, or the method is specified in an exclude-list element, then access to the method is denied. Otherwise, the doesUserHaveRole method is invoked on the JBossSX security manager by the security interceptor to see if the caller has one of the assigned role names. This method iterates through the role names and checks if the authenticated user's Subject Roles group contains a SimplePrincipal with the assigned role name. Access is allowed if any role name is a member of the Roles group. Access is denied if none of the role names are members.
  - If the EJB was configured with a custom security proxy, the method invocation is delegated to it. If the security proxy wants to deny access to the caller, it will throw a java.lang.SecurityException. If no SecurityException is thrown, access to the EJB method is allowed and the method invocation passes to the next container interceptor. Note that the SecurityProxyInterceptor handles this check.
- JMS: Message queue destinations and topic destinations can be protected from access by subjects:
  - Obtain the names of the roles allowed to access the message queue destination or topic destination. The role names are determined by the "SecurityConfig" elements defined for the message queue destination or topic destination descriptor.
  - The TSF permits to specify a global default access control rule which governs the access to the destinations if no access control rule is specified for the individual destination. If no roles have been assigned, or the destinations are not covered by an access control rule (including no global access control rule is specified), then access to the method is denied. Otherwise, the doesUserHaveRole method is invoked on the JBossSX security manager by the security interceptor to see if the caller has one of the assigned role names. This method iterates through the role names and checks if the authenticated user's Subject Roles group contains a SimplePrincipal

© Red Hat, atsec 2009

with the assigned role name. Access is allowed if any role name is a member of the Roles group. Access is denied if none of the role names are members.

- Webservices: Plain old Java Objects (POJOs) (deployed as Servlets) and Session Beans can be protected from access by subjects:
  - POJOs deployed as servlets are protected the same way as specified for the HTTP access control. Session Beans are protected the same way as EJB methods.
- JMX: The JMX invokers can be protected by validating the role of the authenticated user:
  - Every user who has successfully identified and authenticated and is associated with one of the roles required to access the JMX invoker is allowed to access the entire JMX invoker.
  - The validation of the user being associated with a role can be configured in the XML descriptor for JMX. The TOE provides two classes where one needs to be selected configured by the administrator to protect the JMX invokers. One of these classes validates whether the requesting user is associated with the “JBossAdmin” role. The other class validates whether the user is associated at least with one role specified in a configuration file.

This security function covers all SFRs mapped to O.ACCESS

7.2 Audit

The TOE implements an audit mechanism that allows generating audit records for security-relevant events concerning access control. The administrative user is able to select the audited events.

The audit facility is based on the log4j mechanism which is integrated into the TOE. 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 runtime 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.

This security function covers all SFRs mapped to O.AUDITING.

7.3 Clustering

A cluster is a set of nodes. In a JBoss cluster, a node is a JBoss server instance. Thus, to build a cluster, several JBoss instances have to be grouped together (known as a "partition").

Clustering allows the execution of applications on several parallel servers (a.k.a cluster nodes). Two different cluster concepts are possible with JBoss: a failover cluster and a load-distribution cluster. In both cases, the server state is distributed across different servers, and even if any of the servers fails, the application is still accessible via other cluster nodes.

The cluster communication establishes the data consistency between the different cluster nodes of the following information:

- Replication of applications across the cluster which allows to deploy one application on one node and the cluster replicates the application to all nodes (farming deployment)
- Replication of the state of a node covering the replication 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).

  JGroups and JBoss Cache provide the underlying communication, node replication and caching services, for JBoss clusters. Those services are configured as MBeans. There is a set of JBossCache and JGroups MBeans for each type of clustering applications (e.g., the Stateful Session EJBs, the distributed entity EJBs etc.).

  The JGroups framework provides services to enable peer-to-peer communications between nodes in a cluster. It is built on top a stack of network communication protocols that provide transport, discovery, reliability and failure detection, and cluster membership management services.

  JBoss Cache provides distributed cache and state replication services for the JBoss cluster. A JBoss cluster can have multiple JBoss Cache MBeans (known as the TreeCache MBean), one for HTTP session replication, one for stateful session beans, one for cached entity beans, etc.
- Replication of the state of a node covering the replication of HTTP sessions, and EJB 2.x session beans (distributed state replication service using HASessionState MBean).

© Red Hat, atsec 2009

- Replication of the JNDI state (JBoss HA-JNDI)

  The JBoss clustered JNDI service is based on the client-side interceptor architecture. The client must obtain a JNDI stub object (via the InitialContext object) and invoke JNDI lookup services on the remote server through the stub. Furthermore, JNDI is the basis for many other interceptor-based clustering services: those services register themselves with the JNDI so that the client can lookup their stubs and make use of their services.

  The JBoss HA-JNDI (High Availability JNDI) service maintains a cluster-wide context tree. The cluster wide tree is always available as long as there is one node left in the cluster. Each JNDI node in the cluster also maintains its own local JNDI context. The server side application can bind its objects to either tree.

- Replication of JMS queues

  JBoss Messaging clusters JMS queues and topics transparently across the cluster. Messages sent to a distributed queue or topic on one node are consumable on other nodes.

This security function covers the SFR FPT_TRC.1.

7.4 Identification and Authentication

Users are assigned unique user identifies which is used as the basis for access control decisions and auditing. The TOE authenticates the claimed identity of the user before allowing the user to perform any further actions. The TOE internally maintains the identifier associated with the thread spawned for the user after a successful authentication.

The TOE provides different identification and authentication mechanisms for the different request types:

- HTTP and webservices: 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 implements identification and authentication using Java Authentication and Authorization Service (JAAS) with the JBossSX framework. The JAAS framework is provided by the TOE external Java virtual machine. The JBossSX framework uses only the authentication capabilities of JAAS to implement the declarative role-based J2EE security model.

JAAS authentication is performed in a pluggable fashion. This permits Java applications to remain independent from underlying authentication technologies and allows the JBossSX security manager to work in different security infrastructures. Integration with a security infrastructure can be achieved without changing the JBossSX security manager implementation. All that needs to change is the configuration of the authentication stack that JAAS uses. The TOE provides the JAAS modules which are called by the JAAS framework to perform the identification and authentication.

Although the JBossSX framework is heavily dependent on JAAS, the basic security interfaces required for implementation of the J2EE security model are not. The JBossSX framework is simply an implementation of the basic security plug-in interfaces that are based on JAAS. JBossSX provides an abstraction layer which is based on JAAS to other containers of JBoss. The implication of this plug-in architecture is that the administrator is free to replace the JAAS-based JBossSX implementation classes with an individual custom security manager implementation that does not make use of JAAS. The evaluated configuration, however, prohibits the replacement of JBossSX.

The following authentication backends are allowed to be configured with the JAAS modules:

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

The passwords quality used can be enforced with configuration options for the JAAS modules provided by the TOE.

If the JAAS login authenticates the user, a JAAS Subject is created that contains the following in its PrincipalsSet:

- A java.security.Principal that corresponds to the client identity as known in the deployment security environment.
- A java.security.acl.Group named Roles that contains the role names from the application domain to which the user has been assigned. org.jboss.security.SimplePrincipal objects are used to represent the role names;

© Red Hat, atsec 2009

SimplePrincipal is a simple string-based implementation of Principal. These roles are used to validate the roles assigned to methods in ejb-jar.xml and the EJBContext.isCallerInRole(String) method implementation.

- An optional java.security.acl.Group named CallerPrincipal, which contains a single org.jboss.security.SimplePrincipal that corresponds to the identity of the application domain's caller. The CallerPrincipal sole group member will be the value returned by the EJBContext.getCallerPrincipal() method. The purpose of this mapping is to allow a Principal as known in the operational security environment to map to a Principal with a name known to the application. In the absence of a CallerPrincipal mapping the deployment security environment principal is used as the getCallerPrincipal method value. That is, the operational principal is the same as the application domain principal.

This security function covers all SFRs mapped to O.AUTHORIZATION.

7.5 Transaction Rollback

JBoss includes a fast in-VM implementation of a JBoss Transactions compatible transaction manager that is used as the default transaction manager. A transaction is defined as 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. The meanings of these terms are:

- 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 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 afterwards.

In traditional ACID transaction systems, transactions are short lived, resources (such as databases) are locked for the duration of the transaction and participants have a high degree of trust with each other. With the advent of the Internet and Web services, the scenario that is now emerging requires involvement of participants unknown to each other in distributed transactions. JBoss Transactions adds native support for Web services transactions by providing all of the components necessary to build interoperable, reliable, multi-party, Web services-based applications with the minimum of effort. The programming interfaces are based on the Java API for XML Transactioning (JAXTX) and the product includes 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.

This security function covers the SFR FDP_ROL.2-JB.

© Red Hat, atsec 2009

8 Abbreviations

EJB	Enterprise JavaBeans
HA	High Availability
HTTP	Hypertext Transfer Protocol
IIOP	Internet Inter-ORB Protocol
J2EE	See Java EE
JAAS	Java Authentication and Authorization Services
JATAX	Java API for XML Transactioning
Java EE	Java Enterprise Edition
JDBC	Java Database Connectivity
JDK	Java Development Kit
JMS	Java Messaging Service
JMX	Java Management Extensions
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JRMP	Java Remote Method Protocol
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
ORB	Object Request Broker
POJO	Plain Old Java Object
SOAP	Simple Object Access Protocol
VM	Virtual Machine

© Red Hat, atsec 2009