JBoss Enterprise Application Platform 6 Version 6.2.2 Security Target

Version:	**1.1**
Status:	**Released**
Last Update:	**2014-09-26**
Classification:	**Public**

Trademarks

Legal Notice

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

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.0	2014-05-14	Stephan Mueller	First release for JBoss EAP 6.2.2
1.1	2014-05-26	Stephan Mueller	Editorial changes

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	JBoss Enterprise Application Platform 6 Version 6.2.2 Security Target
Version:	1.1
Status:	Released
Date:	2014-09-26
Sponsor:	Red Hat, Inc.
Developer:	Red Hat, Inc.
Certification Body:	BSI
Certification ID:	BSI-DSZ-CC-0909
Keywords:	Security Target, Common Criteria, JBoss, Java EE, Application Server, JBoss Enterprise Application Platform

1.2 TOE Identification

The TOE is JBoss Enterprise Application Platform (EAP) 6 Version 6.2.2.

1.3 TOE Type

The TOE type is Java EE Application Server.

1.4 TOE Overview

This Security Target documents the security characteristics of the JBoss Enterprise Application Platform (in the rest of this document the term “JBoss EAP” is used as a synonym for this TOE).

The TOE of JBoss EAP comprises the following components:

- JBoss EAP 6.2.2

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

The TOE is the JBoss Enterprise Application Platform which implements an application server. JBoss is based on Java Enterprise Edition (Java EE) 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, RMI-IIOP, 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 EAP nodes is defined, then the entire cluster is defined as one TOE.

1.4.1 TOE Type

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

1.4.2 Required Non-TOE Hardware and Software

The Operational Environment for the TOE allows the use of one of the following operating systems together with the mentioned JDKs:

- RedHat Enterprise Linux 5 (x86 and x86_64)
  - OpenJDK 1.6.x
  - OpenJDK 1.7.x
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x
  - IBM JDK 1.6.x
  - IBM JDK 1.7.x
- RedHat Enterprise Linux 6 (x86 and x86_64)
  - OpenJDK 1.6.x
  - OpenJDK 1.7.x
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x
  - IBM JDK 1.6.x
  - IBM JDK 1.7.x
- Solaris 10 (x86 and x86_64, Sparc)
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x
- Solaris 11 (x86 and x86_64, Sparc)
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x
- Microsoft Windows Server 2008 (x86 and x86_64)
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x
- Microsoft Windows Server 2008 R2 (x86_64)
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x
- Microsoft Windows Server 2012 (x86_64)
  - Oracle JDK 1.6.x
  - Oracle JDK 1.7.x

For providing the cryptographic services supporting the SSL/TLS protocol on which the certificate-based authentication relies on, the TOE uses the standard cryptographic service providers shipped with the above mentioned Java Runtime Environments.

Native code, such as OpenSSL or libAIO for Red Hat Enterprise Linux is not used in the evaluated configuration.

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.

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

- IBM DB2 9.7
- IBM DB2 10.1
- Oracle 11g R1
- Oracle 11g R1 RAC
- Oracle 11g R2
- Oracle 11g R2 RAC
- Oracle 12c
- MySQL 5.5
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012
- PostgreSQL 9.2
- Enterprise DB 9.2
- Sybase ASE 15.7

The internal database (H2 DB) is not supported in the evaluated configuration.

1.4.3 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 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.

Several TOE systems may be interlinked in a network. Individual networks may be joined by bridges and/or routers. Each of the TOE instances implements its own security policy. The TOE does not include any synchronization function for those policies between instances of the TOE. Please note that a fail-over cluster configuration of multiple instances of JBoss EAP is considered as one TOE. Therefore, the synchronization operation between the cluster members is considered internal to the TOE. As a result, a single user may have user accounts on each of those TOE instances with different user IDs. One TOE instance (either the standalone system or the cluster configuration of the multiple JBoss EAP instances) 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.

1.4.4 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
- Role-based access control to administrative operations and resources

These primary security features are supported by the appropriate use of domain separation and reference mediation. This separation functionality is provided by the Java virtual machine if the Java Security Manager is utilized. In addition, the underlying operating system supports this separation as well ensuring that the security features are always invoked and cannot be bypassed, and that the TOE can protect itself.

1.5 TOE Description

1.5.1 Introduction

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

1.5.2 Application Server Definition

JBoss is written entirely in Java with the exception of OpenSSL on RHEL and provides a Java EE-compliant environment which is consistent with the Java EE 6 specification. On RHEL, JBoss is able to use OpenSSL to provide fast cryptographic operation. Depending on the configuration of the JBoss server, components required by the Java EE 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 Java EE as provided by JBoss EAP.

The Illustration below titled JBoss components documents the structure of JBoss. The JBoss Modules framework provides the environment for the execution of different containers which allow applications to utilize services provided by these containers. The modules framework uses a different Java class loader for each module. Applications executing within JBoss containers, as well as JBoss EAP components, are started within separate modules. Based on the JVM separation mechanism using different class loaders, the different modules are isolated from each other. Using specifically configured dependencies, the JBoss Modules framework allows the establishment of links between modules.

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 Java EE framework implemented by JBoss, applications can provide their logic to remote clients through the following network protocols:

- HTTP protocol: Java servlets, EJBs, JMS queues 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 using the RMI protocol.

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. Additionally, 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 other resources.

In addition, JBoss EAP supports other protocols encapsulated in the aforementioned protocols, such as HTML or SOAP transmitted over HTTP. However, the security mechanisms defined by this ST are enforced on the above mentioned outer layer protocols.

1.5.3 JBoss Application Server Structure

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

The 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 Java EE functionality, and other containers providing mechanisms to applications which go beyond the Java EE standard into a complete, simple enterprise solution for Java applications.

The Java EE specification considers the following four layers, also called tiers. Applications utilizing the Java EE 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.

Java EE Tier	JBoss Coverage
Client Tier The client tier is the layer of the application executed 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 ● The JMS client	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. 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.

Java EE Tier	JBoss Coverage
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 supported by different frameworks provided by JBoss EAP, such as RESTEasy. The web tier is implemented by the customer-developed application.
Business Tier The business 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 business tier implements business logic, such as the management and maintenance of the sold products and the shopping cart for each user.	The business tier can be implemented using various types of EJBs executing within the JBoss framework. JBoss EAP also supports the implementation of the business logic as POJOs (Plain-Old Java Objects) which grant a greater degree of freedom to the application developer compared to EJBs. The business 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 business tier to access external data stores. This tier therefore covers database access mechanisms, such as a JDBC driver.	The TOE provides the interface to the enterprise information system's tier but does not implement the databases hosting the business data. The TOE allows the application EJBs or POJOs to access relational databases listed for JDBC. The enterprise information system's tier is not implemented by the TOE.

Table 1: Java EE tier listing and JBoss coverage

Fundamentally in the JBoss architecture, the JBoss Module framework manages the set of pluggable component services which are either implemented as POJOs or as MBeans. 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 that are 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 MBeans. In addition, POJOs configured as services can be created for either extending the JBoss functionality or implementing business logic.

The Illustration below titled JBoss components shows the interoperation of the different components of JBoss. JBoss consists of a modular framework where the administrator can selectively enable components. JBoss EAP offers compliance with the Java EE 6 specification and offers services beyond Java EE. The following description applies to the illustration:

- The hardware together with the operating system executes the Java virtual machine which in turn executes the JBoss Modules framework. This framework provides the foundation on which all JBoss containers perform their tasks.
- Each container implements either a service as specified in Java EE 6 or a service providing additional functionality beyond Java EE 6.
- Applications execute as part of containers (such as the JAX-RS Web Services container or the EJB container) and may utilize services from other containers.

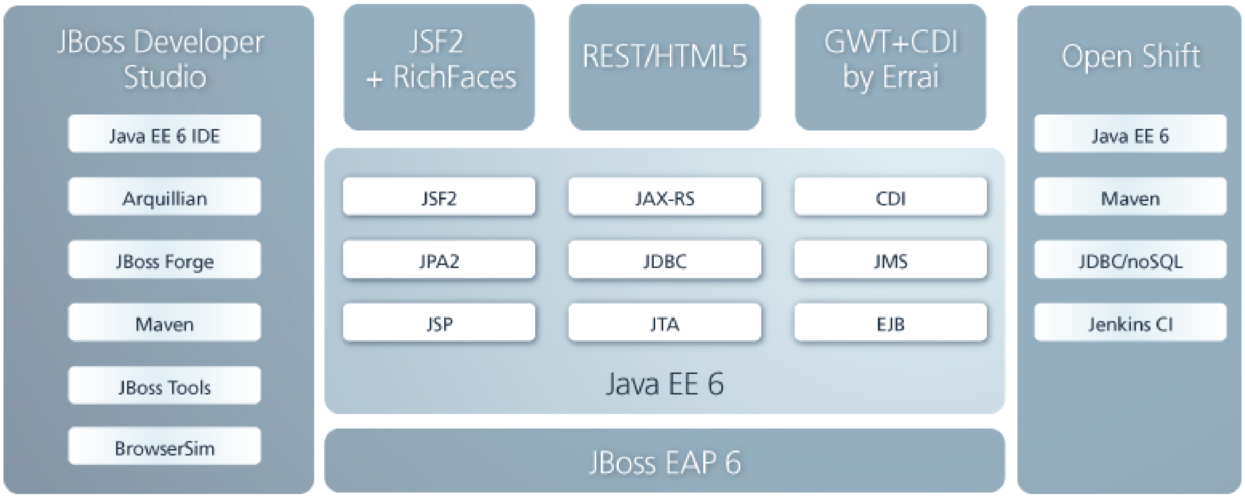

Figure 1: JBoss components

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

- HTTP web network protocol
- Webservices
- Enterprise Java Beans (EJB)
- HornetQ 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 other containers. The technical separation of the untrusted applications and the TOE is achieved using the Java Security Manager with an appropriate policy configuration.

1.5.3.1 Java Security Manager

The evaluated configuration of the TOE only allows the following mode of operation which has an impact on how the TOE can protect itself against the behavior of applications or other untrusted code. This mode utilizes the Java Security Manager provided by the Java Virtual Machine as part of the TOE environment.

The Java Security Manager is utilized with a policy that completely protects the JBoss execution from any application or other untrusted code (such as the JDBC driver or preventing Java reflections) utilizing the JBoss framework. The Security Manager together with its policy prohibits any application from accidentally or intentionally interfering with the operation of JBoss.

It is not allowed to disable the Java Security Manager or to weaken the security policy delivered with the TOE which ensures the protection of the TOE. Together with the TOE, the Security Manager policy that protects the TOE from any application or other untrusted code is provided.

1.5.4 TOE boundaries

1.5.4.1 Physical

The TOE is the JBoss Enterprise Application Platform. Based on the above shown illustration, the TOE consists of the JBoss Modules framework that instantiates the containers/services.

The TOE of JBoss allows the use of all containers and supporting libraries distributed with JBoss EAP in the evaluated configuration (the containers shown in the picture above are present for a better understanding of the TOE only and do not exhaustively list all components).

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.

Relevant guidance documents for the secure operation of the TOE are:

- JBoss Enterprise Application Platform 6.2 Administration and Configuration Guide
- JBoss Enterprise Application Platform 6.2 Security Guide
- JBoss Enterprise Application Platform 6.2 Common Criteria Guide
- JBoss Enterprise Application Platform 6.2 Installation Guide
- JBoss Enterprise Application Platform 6.2 Development Guide

1.5.4.2 Logical

Please see the description of the security functionality in the chapter covering the TOE summary specification.

Evaluated configuration

The evaluated configurations are defined as follows.

- The web management interface provides a web-based access to the management API for administering the TOE. That interface provides access to the configuration aspects that can be set via the XML-based central configuration file for JBoss. The configuration of the TOE restricts access to administrative users.

- The console management interface provides a command line interface access to the management API for administering the TOE. That interface provides access to the configuration aspects that can be set via the XML-based central configuration file for JBoss. By default configuration of the TOE restricts access to administrative users. The management interface offers a role-based access control mechanism to access various management aspects. This way, roles can be defined and mapped to users which limit users to certain administrative operations.
- JBoss remoting allows users to access the JBoss Mbean server component to perform administrative tasks. It must be protected against all users who are not trusted administrators. By default configuration of the TOE restricts access to administrative users.
- The JDBC drivers connecting to any of the database servers must be separated from the TOE using the provided Java Security Manager policy. The Security Manager policy is distributed with the TOE.
- The Security Manager must not allow applications to load native code and restrict the Java reflection API.

1.5.4.3 Security Policy Model

The security policy for JBoss is defined by the security functional requirements in chapter 6. The following is a list of the subjects and objects participating in the policy.

Subjects:

- Users represented by a Principal or Subject object assigned to a management role represented by the Group object

Objects:

- Data accessible at an URL
- EJBs and associated methods
- Message queue and topic
- Target configuration resource and target configuration attribute for a resource

TSF data:

- Deployment descriptors
- Security annotations as part of the Java source code
- User accounts, including the security attributes defined by FIA_ATD.1
- Audit records
- JBoss EAP configuration data

User data:

- Applications deployed with the TOE and all data controlled by them

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.3.

This Security Target does not claim conformance to any Protection Profile.

Common Criteria [CC] version 3.1 revision 4 is the basis for this conformance claim.

3 Security Problem Definition

3.1 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The **assets** to be protected by the TOE comprise of 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** having an interest in manipulating the data model can be categorized as either:

- Unauthorized users of the TOE, i.e., individuals who have not been granted the right to access the system.
- 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 an enhanced-basic attack potential.

3.1.1 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.USER

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

T.ACCESS.MGT

An administrator may gain access to administrative resources or perform administrative 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.

3.2 Assumptions

3.2.1 Environment of use of the TOE

3.2.1.1 Physical

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.

3.2.1.2 Personnel

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.

3.2.1.3 Connectivity

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

The TOE is operated on a system that provides three separate network interfaces following 3-tier configuration of the TOE where the underlying operating system separates the network traffic to these interfaces:

- The network ports of 8080 and 8443 are bound to the public-facing network interface accessible by users interacting with the TOE.
- If the TOE is configured as a cluster, all cluster related ports are bound to a cluster network interface.
- The remaining network ports are bound to an internal network interface where the network is at most only accessible by administrators.

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.

A.TLS

The TOE underlying JDK performs a correct and complete certificate validation of any offered X.509 client certificate.

3.3 Organizational Security Policies

P.ACCOUNTABILITY

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

4 Security Objectives

4.1 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.ROLE

The TSF must control access to administrative resources as well as administrative operations based on the role a user is assigned to. 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. 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 Objectives for the Operational Environment

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.

4.3 Security Objectives Rationale

4.3.1 Security objectives coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.AUTHORIZATION	T.UAUSER
O.ACCESS	T.ACCESS.USER
O.ROLE	T.ACCESS.MGT
O.AUDITING	P.ACCOUNTABILITY
O.CONSISTENCY	T.DIFFER

Table 2: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.ADMIN	A.ADMIN
OE.SYSTEM	A.SYSTEM P.ACCOUNTABILITY

Objective	Assumptions / Threats / OSPs
OE.INSTALL	A.ADMIN A.CLUSTER A.PEER A.TLS
OE.PHYSICAL	A.PROTECT
OE.RECOVER	A.ADMIN
OE.DEVEL	A.DEVEL

Table 3: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Security objectives sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Rationale for security objectives
T.UAUSER	The threat of impersonation 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.USER	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.ACCESS.MGT	The threat of an administrator defined by the different administrative roles listed in the table titled Pre-Configured Management Roles, accessing information resources without authorization from the administrator role responsible for the TOE is removed by O.ROLE requiring a role-based access control for administrative resources and administrative operations. This ensures that an administrator can access such resources and operations only if the requested type of access has been granted by the master administrator responsible for the management of the assignment of roles to users and capabilities to roles.

Threat	Rationale for security objectives
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.

Table 4: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported:

Assumption	Rationale for security objectives
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 TOE is configured with the 3-tier configuration 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.

Assumption	Rationale for security objectives
A.TLS	The assumption the correct and complete certificate validation is covered by OE.INSTALL requiring procedures installation and configuration of the TOE on a list of allowed JDKs. These JDKs are assumed to correctly implement the certificate validation.

Table 5: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
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. To generate appropriate audit entries, OE.SYSTEM ensures that the underlying system provides the time stamp for any action to be audited.

Table 6: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

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 Class FDP: User data protection

5.1.1 (ROL)

Component levelling

The SFR is not hierarchical to any other SFR out of the family of FDP_ROL.

Management: FDP_ROL.2-jb

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

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

Audit: FDP_ROL.2-jb

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Please see the audit information on the FDP_ROL family in CC Part 2.
b) Basic: Please see the audit information on the FDP_ROL family in CC Part 2.
c) Detailed: Please see the audit information on the FDP_ROL family in CC Part 2.

5.1.1.1 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**] .

Rationale

The SFR of FDP_ROL.2-jb is intended to specify an automated rollback of operations by the TOE. Automated rollback addresses the need to roll back or undo all operations within the defined bounds.

6 Security Requirements

6.1 TOE Security Functional Requirements

The following table shows the security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
Java EE Security Mechanisms	FAU_GEN.1 Audit data generation		CC Part 2	No	No	Yes	Yes
	FAU_GEN.2 User identity association		CC Part 2	No	No	No	No
	FDP_ACC.1(HTTP) HTTP Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1(EJB) EJB Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1(HQ) HornetQ Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(HTTP) HTTP Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(EJB) EJB Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(HQ) HornetQ Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ROL.2-jb Automated rollback		ECD	No	No	Yes	No
	FIA_ATD.1 User attribute definition		CC Part 2	No	No	Yes	No
	FIA_UAU.1 Timing of authentication		CC Part 2	No	Yes	Yes	No
	FIA_UID.1 Timing of identification		CC Part 2	No	Yes	Yes	No
	FIA_USB.1 User-subject binding		CC Part 2	No	No	Yes	No
	FMT_MSA.1(EJB) Management of object security attributes	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1(HQ) Management of object security attributes	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3(WEB) Static attribute initialisation	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.3(EJB) Static attribute initialisation	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.3(HQ) Static attribute initialization	FMT_MSA.3	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MTD.1(ACC) Management of TSF data	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(HQ) Management of TSF data	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1(AUTH) Management of TSF data	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FPT_TRC.1 Internal TSF consistency		CC Part 2	No	No	Yes	No
JBoss EAP Management security	FDP_ACC.1(RBAC) Role-Based Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1(RBAC) Role-Based Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FMT_MSA.1(RBAC) Management of object security attributes	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3(RBAC) Static attribute initialization	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_SMF.1 Specification of management functions		CC Part 2	No	No	Yes	No
	FMT_SMR.2 Security roles		CC Part 2	No	No	Yes	No

Table 7: Security functional requirements for the TOE

6.1.1 Java EE Security Mechanisms

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 (if applicable), 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.

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

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 user role represented by the Group object;

Object: data accessible at URL;

Operations: all HTTP standard and user-defined methods

Application Note: *Access control is managed with appropriate settings in the deployment descriptor of web.xml.*

Application Note: *The access control functionality implemented by the web subsystem as specified in this SFR also covers access to web services application.*

6.1.1.4 EJB Access Control Policy (FDP_ACC.1(EJB))

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 user 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 deployment descriptor 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).*

6.1.1.5 HornetQ Access Control Policy (FDP_ACC.1(HQ))

FDP_ACC.1.1 The TSF shall enforce the **HornetQ Access Control policy** on

Subject: a user represented by a Principal or Subject object assigned to a user role represented by the Group object;

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 TOE configuration.*

Application Note: *Message queues (one sender, one receiver) and topics (one sender, multiple receivers) are communication facilities allowing different subjects to exchange information.*

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

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

a) **Subject attributes: User roles;**

b) **Object attributes: URL, user 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:

a) **Deployment descriptor / configuration file: the requesting user is associated with a user role specified for the URL and HTTP method in the "security-constraint" element defined in the deployment descriptor web.xml;**

b) **Deployment descriptor / configuration file: the transport layer security used when accessing the URL must cover at least that security mechanism defined by the "user-data-constraint" element defined in the deployment descriptor web.xml for the accessed URL, requiring either no protection, integrity protection or confidentiality protection.**

FDP_ACF.1.3 The TSF shall explicitly authorise 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 additional rules: **providing an empty auth-constraint element**.

Application Note: *Application developers have the possibility to define additional access control rules in the deployment descriptors applicable to their applications. The TOE provides mechanisms like ACLs which can be used to provide additional access restrictions. However, these additional access control mechanisms can only add additional restrictions without violating the restrictions defined in this SFR. Therefore, these additional access control mechanisms are allowed to be used although not attributed with security claims in this ST and therefore outside the scope of the evaluation.*

6.1.1.7 EJB Access Control Functions (FDP_ACF.1(EJB))

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

a) **Subject attributes: User roles;**

b) **Object attributes: EJB name and associated method name, user 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 user role specified for the EJB method in the "method-permission" element defined in the deployment descriptor ejb-jar.xml.

FDP_ACF.1.3 The TSF shall explicitly authorise 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**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **methods marked with the "exclude-list" element are always denied access to**.

Application Note: *Application developers have the possibility to define additional access control rules in the deployment descriptors applicable to their applications. The TOE provides mechanisms like ACLs which can be used to provide additional access restrictions. However, these additional access control mechanisms can only add additional restrictions without violating the restrictions defined in this SFR. Therefore, these additional access control mechanisms are allowed to be used although not attributed with security claims in this ST and therefore outside the scope of the evaluation.*

6.1.1.8 HornetQ Access Control Functions (FDP_ACF.1(HQ))

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

a) **Subject attributes: User roles;**

b) **Object attributes: message queue name, topic name, user 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 user role specified for the respective communication facility based on the following rules:**

a) **createDurableQueue. This permission allows the user to create a durable queue under matching addresses.**

b) **deleteDurableQueue. This permission allows the user to delete a durable queue under matching addresses.**

c) **createNonDurableQueue. This permission allows the user to create a non-durable queue under matching addresses.**

d) **deleteNonDurableQueue. This permission allows the user to delete a non-durable queue under matching addresses.**

e) **send. This permission allows the user to send a message to matching addresses.**

f) **consume. This permission allows the user to consume a message from a queue bound to matching addresses.**

g) **manage. This permission allows the user to invoke management operations by sending management messages to the management address.**

FDP_ACF.1.3 The TSF shall explicitly authorise 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 additional rules: **none**.

6.1.1.9 Automated rollback (FDP_ROL.2-jb)

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**.

6.1.1.10 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) User Role;

c) Management Role;

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

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

Application Note: *The TOE user databases are configured in the TOE configuration. Various user databases can be configured and mapped to the different services offered by the TOE. In addition, Java annotations can be used to define users.*

6.1.1.11 Timing of 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 *except for the certificate based authentication implemented with JBoss Remoting:*

1. *All actions allowed by the access control mechanisms for the identity assigned to an unauthenticated principal defined by the container implementing the respective action.*
2. *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.*
3. *All URLs (i) without a “security-constraint” element defined in the web.xml deployment descriptor, or (ii) without the “@RolesAllowed” and without the “@DenyAll” Java Annotations defined for EJB 3 servlets, or (iii) without the @ServletSecurity(@HttpConstraint(rolesAllowed)) @ServletSecurity(@HttpMethodConstraint(rolesAllowed)) Java Annotation 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.

Application Note: *For the certificate based authentication implemented with JBoss Remoting, A.TLS applies instead of FIA_UAU.1.*

6.1.1.12 Timing of 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:

1. *All actions allowed by the access control mechanisms for the identity assigned to an unauthenticated principal defined by the container implementing the respective action.*
2. *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.*
3. *All URLs (i) without a “security-constraint” element defined in the web.xml deployment descriptor, or (ii) without the “@RolesAllowed” and without the “@DenyAll” Java Annotations, or (iii) without the @ServletSecurity(@HttpConstraint(rolesAllowed)) or @ServletSecurity(@HttpMethodConstraint(rolesAllowed)) Java Annotation defined for EJB 3 servlets are accessible to unauthenticated users.*

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

6.1.1.13 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) **User role the user is operating with as used by the different access control mechanisms defined in FDP_ACC.1(HTTP), FDP_ACC.1(EJB), FDP_ACC.1(HQ);**

c) **Administrative role the user is operating with as used by the management interfaces defined by FDP_ACC.1(RBAC), FMT_SMR.2.**

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 users:

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

c) **The management role associated with a subject shall be all 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 users:

a) **run-as: The security role defined with the run-as element in a deployment descriptor or annotation is used for the execution of methods invoked by the component marked by run-as.**

b) **run-as-principal: The identity (principal) defined with the run-as-principal element in a deployment descriptor or annotation is used for the execution of methods invoked by the component marked by run-as-principal.**

6.1.1.14 Management of object security attributes (FMT_MSA.1(EJB))

FMT_MSA.1.1 The TSF shall enforce the **EJB Access Control Policy** to restrict the ability to **modify** the security attributes **of the default value of the SFP** to **authorized administrators assigned to the maintainer role, deployer role, administrator role or superuser role**.

6.1.1.15 Management of object security attributes (FMT_MSA.1(HQ))

FMT_MSA.1.1 The TSF shall enforce the **HornetQ Access Control Policy** to restrict the ability to **modify** the security attributes **of the default value of the SFP** to **authorized administrators assigned to the maintainer role, deployer role, administrator role or superuser role**.

6.1.1.16 Static attribute initialisation (FMT_MSA.3(WEB))

FMT_MSA.3.1 The TSF shall enforce the **HTTP 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 ~~the~~**nobody** to specify alternative initial values to override the default values when an object or information is created.

Application Note: *The default values are hard-coded and cannot be changed.*

6.1.1.17 Static attribute initialisation (FMT_MSA.3(EJB))

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

FMT_MSA.3.2 The TSF shall allow ~~the~~**authorized administrators assigned to the maintainer role, deployer role, administrator role or superuser role** to specify alternative initial values to override the default values when an object or information is created.

Application Note: *The default value can be changed with the default-missing-method-permission-deny-access configuration value.*

6.1.1.18 Static attribute initialization (FMT_MSA.3(HQ))

FMT_MSA.3.1 The TSF shall enforce the **HornetQ Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow ~~the~~**authorized administrators assigned to the maintainer role, deployer role, administrator role or superuser role** to specify alternative initial values to override the default values when an object or information is created.

Application Note: *The default behavior for message queue or topic destinations is defined with the JBoss system configuration.*

Application Note: *This SFR specifies the management of the default value. For protecting the default value, the TOE relies on the environment to protect the deployment descriptor file holding the default value. As only authorized administrators are able to access the system hosting the TOE as assumed with A.PROTECT, the protection of the deployment descriptor is ensured.*

6.1.1.19 Management of TSF data (FMT_MTD.1(ACC))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **access control settings for the HTTP Access Control Policy, EJB Access Control Policy** to **the user owning the application the access control restrictions apply to**.

Application Note: *The access control policies are defined in the deployment descriptor or other locations for each application as referenced in the application notes for each FDP_ACC.1 iteration.*

6.1.1.20 Management of TSF data (FMT_MTD.1(HQ))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **access control settings for the HornetQ Access Control Policy,** to **authorized administrators assigned to the maintainer role, deployer role, administrator role or superuser role**.

6.1.1.21 Management of TSF data (FMT_MTD.1(AUTH))

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **user account settings for the HTTP Access Control Policy, EJB Access Control Policy, HornetQ Access Control Policy** to

a) **Specifying configuration in configuration file: authorized administrators assigned to the maintainer role, deployer role, administrator role or superuser role**

b) **Specifying configuration in deployment descriptor: owner of the application subject to the configuration**

.

Application Note: *The access control policies are defined in the configuration file, or deployment descriptor or other locations for each application as referenced in the application notes for each FDP_ACC.1 iteration.*

6.1.1.22 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:**

1. **HTTP requests including webservices requests;**
2. **EJB requests;**
3. **HornetQ requests;**
4. **Management requests subject to RBAC.**

Application Note: *This SFR covers the cluster communication that synchronizes the runtime state of the different cluster nodes.*

6.1.2 JBoss EAP Management security

6.1.2.1 Role-Based Access Control Policy (FDP_ACC.1(RBAC))

FDP_ACC.1.1 The TSF shall enforce the **Role-Based Access Control policy** on

Subject: a user represented by a Principal or Subject object assigned to a management role represented by the Group object;

Object:

- **Target configuration resource**
- **Target configuration attribute for a resource**

Operations:

- **Model operations - read/write from the model which may imply the starting and stopping of runtime services:**
  - **Resource visibility**
  - **Resource read**
  - **Resource write**
- **RPC operations - operations affecting the runtime state only by either read or change runtime state without affecting the model:**
  - **Resource visibility**
  - **read-runtime**
  - **write-runtime**

6.1.2.2 Role-Based Access Control Functions (FDP_ACF.1(RBAC))

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

a) **Subject attributes: management roles;**

b) **Object attributes: sensitive flag, application classification, set of objects, operations and additional activation constraints allowed for the respective object referenced by a management 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:

Access to the administrative object with the requested operation is permitted if:

a) **the requesting user is associated with a management role containing the object and operation combination and is not restricted by additional activation constraints.**

FDP_ACF.1.3 The TSF shall explicitly authorise 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 additional rules: **none**.

6.1.2.3 Management of object security attributes (FMT_MSA.1(RBAC))

FMT_MSA.1.1 The TSF shall enforce the **Role-Based Access Control Policy** to restrict the ability to **modify, delete** the security attributes **of the assignment of objects and operations to management roles** to **authorized administrators assigned to the administrator role or superuser role**.

6.1.2.4 Static attribute initialization (FMT_MSA.3(RBAC))

FMT_MSA.3.1 The TSF shall enforce the **Role-Based Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **authorized administrators assigned to the administrator role or superuser role** to specify alternative initial values to override the default values when an object or information is created.

6.1.2.5 Specification of management functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

a) Management of access control policies;

b) Management of default value for HornetQ access control policy;

c) Management of default value for role-based access control policy.

6.1.2.6 Security roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

a) Administrative roles governing access according FDP_ACC.1(RBAC)

b) User roles governing access according to all other iterations of FDP_ACC.1

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

a) The set of capabilities assigned to management roles allow the modification of security attributes for all objects accessible via the administrative interfaces.

b) User roles allow access to application functions and mechanisms as configured by the application owner.

are satisfied.

6.2 Security Functional Requirements Rationale

6.2.1 Security requirements coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security functional requirements	Objectives
FAU_GEN.1	O.AUDITING

Security functional requirements	Objectives
FAU_GEN.2	O.AUDITING
FDP_ACC.1(HTTP)	O.ACCESS
FDP_ACC.1(EJB)	O.ACCESS
FDP_ACC.1(HQ)	O.ACCESS
FDP_ACF.1(HTTP)	O.ACCESS
FDP_ACF.1(EJB)	O.ACCESS
FDP_ACF.1(HQ)	O.ACCESS
FDP_ROL.2-jb	O.CONSISTENCY
FIA_ATD.1	O.ACCESS, O.AUTHORIZATION, O.ROLE
FIA_UAU.1	O.AUTHORIZATION
FIA_UID.1	O.AUTHORIZATION
FIA_USB.1	O.ACCESS, O.AUTHORIZATION, O.ROLE
FMT_MSA.1(EJB)	O.AUTHORIZATION
FMT_MSA.1(HQ)	O.ACCESS, O.AUTHORIZATION, O.ROLE
FMT_MSA.3(WEB)	O.ACCESS, O.AUTHORIZATION
FMT_MSA.3(EJB)	O.ACCESS, O.AUTHORIZATION
FMT_MSA.3(HQ)	O.ACCESS, O.AUTHORIZATION, O.ROLE
FMT_MTD.1(ACC)	O.ACCESS
FMT_MTD.1(HQ)	O.ACCESS, O.ROLE
FMT_MTD.1(AUTH)	O.AUTHORIZATION, O.ROLE
FPT_TRC.1	O.CONSISTENCY
FDP_ACC.1(RBAC)	O.ROLE
FDP_ACF.1(RBAC)	O.ROLE

Security functional requirements	Objectives
FMT_MSA.1(RBAC)	O.ROLE
FMT_MSA.3(RBAC)	O.ROLE
FMT_SMF.1	O.ACCESS, O.AUTHORIZATION, O.ROLE
FMT_SMR.2	O.ACCESS, O.AUTHORIZATION, O.ROLE

Table 8: Mapping of security functional requirements to security objectives

6.2.2 Security requirements sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
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]. In case of accessing the TOE using JBoss Remoting tunneled through TLS and configuring the TOE to perform certificate-based authentication, the TOE only implements the identification of the peer and relies on A.TLS for the authentication aspect. 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]. The management of the authorizations is specified in [FMT_MTD.1(AUTH), FMT_SMF.1, FMT_SMR.2]. The default value for the HornetQ access control policy is modifiable as specified with [FMT_MSA.3(HQ) and FMT_MSA.1(HQ)]. Nobody is able to control which default values are configured for the remaining access control mechanisms [FMT_MSA.3(WEB), FMT_MSA.3(EJB) together with FMT_MSA.1(EJB)]. Please note that the default values have an impact on the authorization, as a permissive default value allows access to the resource without I&A.
O.ACCESS	The different access control mechanisms 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 security attributes of subjects used to enforce the different access control mechanisms must be defined [FIA_ATD.1, FIA_USB.1]. The management of the access control settings is specified in [FMT_MTD.1(ACC), FMT_SMF.1, FMT_SMR.2].

Security objectives	Rationale
	The default value for the HornetQ access control policy is modifiable as specified with [FMT_MSA.3(HQ) and FMT_MSA.1(HQ)]. Nobody is able to control which default values are configured for the remaining access control mechanisms [FMT_MSA.3(WEB), FMT_MSA.3(EJB) together with FMT_MSA.1(EJB)].
O.ROLE	The role-based access control mechanism as defined in FDP_ACC.1(RBAC) and FDP_ACF.1(RBAC) ensures that the administrative operations and administrative resources are accessible to authorized administrators only when they belong to the role granted access. The capabilities that can be assigned to roles are specified in each management function FMT_MSA.1(HQ), FMT_MSA.3(HQ), and the iterations of FMT_MTD.1 which define operations accessible to authorized administrators assigned to management roles. The security attributes of roles mapped to subjects used to enforce the role-based access control mechanisms must be defined [FIA_ATD.1, FIA_USB.1]. The management of the role-based access control mechanism is defined in FMT_MSA.1(RBAC) and FMT_MSA.3(RBAC).
O.AUDITING	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].

Table 9: Security objectives for the TOE rationale

In addition, the following listing demonstrates the internal consistency of the SFRs:

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), supported by A.TLS, proper binding of subjects to users (as required by FIA_USB.1). FMT_MSA.3(WEB), FMT_MSA.3(EJB), and FMT_MSA.3(HQ) define the default permissions for the different access control mechanisms. The management of access control settings specified in FMT_MSA.1(EJB), FMT_MSA.1(HQ), and FMT_MTD.1(ACC) as well as account settings with FMT_MTD.1(AUTH) support the access control policies.

Role-based Access Control for Management

FDP_ACC.1(RBAC) together with FDP_ACF.1(RBAC) specify the role-based access control policy with the subject, objects and actions covered by the policy. This access control mechanism is tied with roles in FMT_SMR.2. The role-based access control mechanism is configured as defined in FMT_MSA.1(RBAC) and FMT_MSA.3(RBAC). The different iterations of FMT_MSA.1 and FMT_MSA.3 that allow management activities specify the authorization needed to perform administration. To be effective an access control mechanism requires users to be properly identified and authenticated (as required by FIA_UID.1(COMMON) and FIA_UAU.1), supported by A.TLS, proper binding of subjects to users (as required by FIA_USB.1).

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

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 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.

6.2.3 Security requirements dependency analysis

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	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 operational environment. The TOE relies on the underlying Java virtual machine 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 operational 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.
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UID.1
FDP_ACC.1(HTTP)	FDP_ACF.1	FDP_ACF.1(HTTP)
FDP_ACC.1(EJB)	FDP_ACF.1	FDP_ACF.1(EJB)
FDP_ACC.1(HQ)	FDP_ACF.1	FDP_ACF.1(HQ)
FDP_ACF.1(HTTP)	FDP_ACC.1	FDP_ACC.1(HTTP)
	FMT_MSA.3	FMT_MSA.3(WEB)
FDP_ACF.1(EJB)	FDP_ACC.1	FDP_ACC.1(EJB)
	FMT_MSA.3	FMT_MSA.3(EJB)
FDP_ACF.1(HQ)	FDP_ACC.1	FDP_ACC.1(HQ)
	FMT_MSA.3	FMT_MSA.3(HQ)
FDP_ROL.2-jb	No dependencies.	
FIA_ATD.1	No dependencies.	
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UID.1	No dependencies.	
FIA_USB.1	FIA_ATD.1	FIA_ATD.1

Security Functional Requirement	Dependencies	Resolution
FMT_MSA.1(EJB)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(EJB)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(HQ)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(HQ)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(WEB)	FMT_MSA.1	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. As specified in the SFR, the TOE does not implement an interface that allows altering of the default settings for the different access control mechanisms. Thus, the dependency on FMT_MSA.1 is not fulfilled.
	FMT_SMR.1	As explained, the management of the access control mechanism is not required by the TOE. Hence, FMT_SMR.1 is appropriately excluded.
FMT_MSA.3(EJB)	FMT_MSA.1	FMT_MSA.1(EJB)
	FMT_SMR.1	FMT_SMR.2
FMT_MSA.3(HQ)	FMT_MSA.1	FMT_MSA.1(HQ)
	FMT_SMR.1	FMT_SMR.2
FMT_MTD.1(ACC)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(HQ)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AUTH)	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1

Security Functional Requirement	Dependencies	Resolution
FPT_TRC.1	FPT_ITT.1	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.
FDP_ACC.1(RBAC)	FDP_ACF.1	FDP_ACF.1(RBAC)
FDP_ACF.1(RBAC)	FDP_ACC.1	FDP_ACC.1(RBAC)
	FMT_MSA.3	FMT_MSA.3(RBAC)
FMT_MSA.1(RBAC)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(RBAC)
	FMT_SMR.1	FMT_SMR.2
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(RBAC)	FMT_MSA.1	FMT_MSA.1(RBAC)
	FMT_SMR.1	FMT_SMR.2
FMT_SMF.1	No dependencies.	
FMT_SMR.2	FIA_UID.1	FIA_UID.1

Table 10: TOE SFR dependency analysis

6.3 Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 components as specified in [CC] part 3, augmented by ALC_FLR.3.

The following table shows the Security assurance requirements, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.4 Complete functional specification	CC Part 3	No	No	No	No
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3	No	No	No	No
	ADV_TDS.3 Basic modular design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.4 Production support, acceptance proce dures and automation	CC Part 3	No	No	No	No
	ALC_CMS.4 Problem tracking CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No
	ALC_DVS.1 Identification of security measures	CC Part 3	No	No	No	No
	ALC_FLR.3 Systematic flaw remediation	CC Part 3	No	No	No	No
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3	No	No	No	No
	ALC_TAT.1 Well-defined development tools	CC Part 3	No	No	No	No
ASE Security Target evaluation	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_CCL.1 Conformance claims	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.2 Analysis of coverage	CC Part 3	No	No	No	No
	ATE_DPT.1 Testing: basic design	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3	No	No	No	No

Table 11: Security assurance requirements

6.4 Security Assurance Requirements Rationale

The evaluation assurance level commensurate with the threat environment that is experienced by typical consumers of the TOE. In addition, the evaluation assurance level augmented with ALC_FLR.3 commensurate with the augmented flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

7 TOE Summary Specification

7.1 TOE Security Functionality

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

The primary security features of the TOE are:

- Access Control
- Role-based access control for management interfaces
- Audit
- Clustering
- Identification and Authentication
- Transaction Rollback

7.1.1 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 as part of the HTTP deployment descriptor or the "@ServletSecurity" annotation. Note, JBoss EAP supports all HTTP request methods specified in the RFCs and custom methods. 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).
- 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 deployment descriptor or annotation.
  - 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 PicketBox 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 role with the given 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.
- HornetQ: 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 “security-setting” elements defined for the message queue destination or topic destination in the JBoss system configuration file.

  - ○ 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 PicketBox 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 the required 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.

The above mentioned network protocols tunnel the client requests to the TOE. After the TOE performed the I&A and access control checks, the request is forwarded to the intended application. As the TOE only uses the credential information from the network request, only the aspect of communicating the user credentials as well as the requested object and the request type is relevant for the enforcement of the access control policy.

The TOE allows independent management of the access control policy for each application and for each policy. The mentioned deployment descriptors, and annotations can be used by authorized administrators (and developers which belong to the category of authorized administrators as specified in A.DEVEL) to configure the access control. Note that the TOE provides the interfaces for managing the access control policies. However, it does not restrict the use of the interfaces as direct access to the JBoss system configuration file is technically possible if a user can log into the host system and has write access to that file - based on A.PROTECT the environment ensures that only authorized administrators are allowed to access the host system.

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

7.1.2 Role-based Access Control for Management Interfaces

The management interfaces of JBoss, the command line interface as well as the web-based administrative interface, allow access to the JBoss system configuration to manage all configurable aspects of JBoss EAP. Administrators can access general system aspects, such as network port configurations, and container configuration. In addition, configuration aspects for services offered by containers are managed as well.

The configuration aspects of applications, such as the application access control, are addressed with the deployment descriptors shipped with the application. Therefore, this configuration aspect is not accessible via the administrative interface.

The administrative interfaces can be bound to a specific network interface. This allows the maintenance of an admin LAN to prevent untrusted users from technically accessing the interfaces. In order for an administrator to interact with administrative interfaces, he must log in. The administrative accounts are maintained separately from other user accounts.

Each action on an object that an administrative user can perform is subject to a role-based access control mechanism. The actions are classified into:

- Model operations - the main function of these is to read/write from the model, although there will often be associated runtime services started/stopped as a consequence.
- RPC operations - these invoke some runtime affecting runtime state only. This may either read runtime state or change it. The model is not affected.

The objects are classified based on the following:

- A resource
- An attribute residing in a resource

A set of object-action capabilities are mapped to a management role. This mapping defines the allowed access for this management role. A set of pre-defined roles is shipped with the TOE and is available after installation. The following table specifies the pre-configured roles.

Role	Description
Monitor	The monitor role has the fewest permissions and restricts the user to viewing the configuration and the current state. The monitor role does not have permission to view sensitive data.
Configurator	The configurator role has the same permissions as the monitor role, and can change the configuration. For example, the configurator can deploy an application. The configurator role does not have permission to view sensitive data.
Operator	The operator role has monitor permissions and can also change the runtime state but not the persistent configuration. For example, the operator can start or stop servers.The operator role does not have permission to view sensitive data.
Administrator	The administrator role has the combined permissions of the operator and the configurator. This role has also permission to access sensitive data, including passwords. The administrator role is the superuser of the Application Server and can modify administrative users and roles.
Deployer	The deployer role has the combined permissions of the operator and the configurator, but with those permissions constrained to operating on deployments.
Auditor	The auditor role can view and modify the configuration settings for the security auditing system. The auditor role includes the monitor role, allowing the auditor to view but not change the rest of the security configuration.

Table 12: Role-based access control pre-configured management roles

A role is a named set of permissions. Those permissions include constraints (e.g., the read permissions for the Monitor role is constrained to non-sensitive actions and targets).

Redefinition of the permissions and constraints associated with the above mentioned standard roles is not permitted.

A limited form of creation of new roles is allowed. These new roles are equivalent to the standard roles, but with an additional constraint applied to all permissions, i.e., the target must be related to a particular host or server group.

The JBoss system configuration file is either domain.xml or standalone.xml depending on the startup mode of JBoss or a configuration file specified via a command line option. Any administrative operations are stored in that configuration file. The administrative interfaces are an in-memory image of the data stored in the XML file. Once the in-memory image is modified, the modified XML file is stored.

The role-based access control mechanism can only be enforced if the administrator accesses the JBoss system configuration via the above mentioned interfaces. It is technically possible that an administrator has shell access to the host. In this case, the underlying operating system may grant direct read or write access to that JBoss system configuration file. Such access would imply that the role-based access control mechanism is not enforced. However, as assumed in A.PROTECT, the host is located in a protected environment, where direct access to the JBoss system configuration file is not allowed.

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

7.1.3 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 events which are to be audited.

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 based on message type and level, and to control how these messages are formatted and where they are reported at runtime.

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.

7.1.4 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 the state of a node covers 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 Infinispan).

  JGroups and Infinispan provide the underlying communication, node replication and caching services, for JBoss clusters. Those services are configured as MBeans. There is a set of Infinispan 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.

  Infinispan provides distributed cache and state replication services for the JBoss cluster. A JBoss cluster can have multiple Infinispan MBeans, 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.
- Replication of HornetQ queues.

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

JBoss EAP does not perform an automated replication of the JNDI state. When applications defining JNDI resources are replicated to the different cluster nodes, they are newly deployed at the nodes. With the deployment, the JNDI resources are created similarly to a regular deployment. System configuration changes that involve modifications of JNDI resources are replicated to the cluster nodes and applied similarly to a local reconfiguration. With these approaches, the JNDI registry maintaining the JNDI mappings are managed consistently between the different cluster nodes. As JNDI does not maintain a state other than the JNDI registry, the approach is sufficient to ensure cluster-wide consistency of the JNDI service.

This security function covers the SFR FPT_TRC.1.

7.1.5 Identification and authentication

Users are assigned unique user identifiers which are 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 TSF-mediated 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 identification -- For client certificate based identification, the TOE uses the SSL/TLS channel established by the underlying JVM. The JVM performs the certificate validation of the client certificate. The EJB component of the TOE queries the JVM for the DN part of the certificate to identify the user. That DN information is used to set up the role mapping and create a principal in the TOE. Therefore, the TOE relies on the JVM SSL/TLS implementation to perform the authentication by enforcing the certificate validation.
- HornetQ: username and password based authentication

JBoss implements identification and authentication using Java Authentication and Authorization Service (JAAS) with the PicketBox framework. The JAAS framework is provided by the Java virtual machine in the operational environment. The PicketBox framework uses only the authentication capabilities of JAAS to implement the declarative role-based Java EE security model.

JAAS authentication is performed in a pluggable fashion. This permits Java applications to remain independent from underlying authentication technologies and allows the PicketBox security manager to work in different security infrastructures. Integration with a security infrastructure can be achieved without changing the PicketBox 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 PicketBox framework is heavily dependent on JAAS, the basic security interfaces required for implementation of the JAVA EE security model are not. The PicketBox framework is simply an implementation of the basic security plug-in interfaces that are based on JAAS. PicketBox 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 PicketBox implementation classes with an individual custom security manager implementation that does not make use of JAAS. The evaluated configuration, however, prohibits the replacement of PicketBox.

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

- File based authentication using UsersRolesLoginModule
- File based authentication for EJB Remoting Framework using RemotingLoginModule
- Certificate based authentication using BaseCertLoginModule
- LDAP based authentication using LdapLoginModule
- Advanced LDAP based authentication using LdapExtLoginModule
- Database based authentication using DatabaseServerLoginModule

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 or custom objects registered as principalClass are used to represent the role names; 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.

The above mentioned network protocols tunnel the client requests to the TOE. After the TOE performed the I&A checks, the request is forwarded to the intended application. As the TOE only uses the credential information from the network request, only the aspect of communicating the user credentials is relevant for the enforcement of the I&A policy.

The TOE allows the management of the authorization independently for each application and service. The mentioned deployment descriptors, and annotations can be used by authorized administrators (and developers which belong to the category of authorized administrators as specified in A.DEVEL) to configure the I&A mechanism. Note that the TOE provides the interfaces for managing the I&A policy. However, it does not restrict the use of the interfaces to authorized administrators. These settings are stored in the JBoss system configuration. This configuration file could be accessed by users with write permissions when having access to the host system. Thus, the host system ensures that only authorized administrators are allowed to access the host system as assumed by A.PROTECT.

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

7.1.6 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 participants 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.

8 Abbreviations, Terminology and References

8.1 Abbreviations

ACL
Access Control List

API
Application Programming Interface

EJB
Enterprise Java Beans

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 Transationing

Java EE
Java Enterprise Edition

JAX-RS
Java API for RESTful Web Services

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

RBAC
Role-Based Access Control

SFR
Security Functional Requirement

SOAP
originally defined as Simple Object Access Protocol

SSL
Secure Sockets Layer

ST
Security Target

TCP/IP
Transmission Control Protocol / Internet Protocol

TLS
Transport Layer Security

TOE
Target of Evaluation

TSF
TOE Security Functionality

VM
Virtual Machine

VPN
Virtual Private Network

XML
Extensible Markup Language

8.2 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 management roles of a JBoss system. The TOE defines a set of management 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.

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.

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 JBossWeb container. The JBoss architecture implements various functional aspects as self-contained containers which can be selectively enabled.

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

To authorize access to resources, applications first need to authenticate the source of the request. The JAAS framework defines the term subject to represent the source of a request. A subject may be any entity, such as a person or a service. Once the subject is authenticated, a javax.security.auth.Subject is populated with associated identities, or Principals. A Subject may have many Principals. For example, a person may have a name Principal ("John Doe") and a SSN Principal ("123456789"), which distinguish it from other subjects. A Subject may also own security-related attributes, which are referred to as credentials. Sensitive credentials that require special protection, such as private cryptographic keys, are stored within a private credential Set. Credentials intended to be shared, such as public key certificates, are stored within a public credential Set.

Product

The term product is used to define software components that comprise the JBoss Enterprise Application Platform.

Role

A role represents a set of actions that an authorized user, upon assuming the role, is allowed to perform.

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.

8.3 References

CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R4
	Date	September 2012
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R4.pdf