Ministero dello Sviluppo Economico

Direzione generale per le tecnologie delle comunicazioni e la sicurezza informatica -
Istituto Superiore delle Comunicazioni e delle Tecnologie dell'Informazione

OCSI

Organismo di Certificazione della Sicurezza Informatica

Schema nazionale per la valutazione e certificazione della sicurezza di sistemi e prodotti ICT
(DPCM del 30 ottobre 2003 - G.U. n. 93 del 27 aprile 2004)

Certificato n. 9/19

(Certification No.)

Prodotto: JBoss Enterprise Application Platform 7 Version 7.2.3
(Product)

Sviluppato da: Red Hat, Inc.
(Developed by)

Il prodotto indicato in questo certificato è risultato conforme ai requisiti dello standard ISO/IEC 15408 (Common Criteria) v. 3.1 per il livello di garanzia:

The product identified in this certificate complies with the requirements of the standard ISO/IEC 15408 (Common Criteria) v. 3.1 for the assurance level:

EAL4+
(ALC_FLR.3)

Il Direttore
(Dott.ssa Eva Spina)

Roma, 2 dicembre 2019

Common Criteria

Fino a EAL2 *(Up to EAL2)*

This page is intentionally left blank

Ministero dello Sviluppo Economico

Direzione generale per le tecnologie delle comunicazioni e la sicurezza informatica -
Istituto Superiore delle Comunicazioni e delle Tecnologie dell'Informazione

Organismo di Certificazione della Sicurezza Informatica

Certification Report

JBoss Enterprise Application Platform 7 Version 7.2.3

OCSI/CERT/ATS/05/2018/RC

Version 1.0

2 December 2019

Courtesy translation

Disclaimer: this translation in English language is provided for informational purposes only; it is not a substitute for the official document and has no legal value. The original Italian language version of the document is the only approved and official version.

1 Document revisions

Version	Author	Information	Date
1.0	OCSI	First issue	02/12/2019

2 Table of contents

3 Acronyms

ACID	Atomicity, Consistency, Isolation and Durability
API	Application Programming Interface
CC	Common Criteria
CCRA	Common Criteria Recognition Arrangement
CEM	Common Evaluation Methodology
CM	Configuration Management
CP	Customer Portal
cPP	collaborative Protection Profile
DB	Database
DMR	Dynamic Model Representation
DN	Domain Name
DPCM	Decreto del Presidente del Consiglio dei Ministri
EAL	Evaluation Assurance Level
EJB	Enterprise Java Beans
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IT	Information Technology
JAX-RS	Java API for RESTful Web Services
JAX-WS	Java API for XML-based Web Services
JAXTX	XML Transactioning API for Java
JDBC	Java DataBase Connectivity
JDK	Java Development Kit
JMS	Java Messaging Service
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment

JSP	JavaServer Pages
JVM	Java Virtual Machine
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LGP	Linea Guida Provvisoria
LVS	Laboratorio per la Valutazione della Sicurezza
MBeans	Managed Bean
MSC	Modular Service Container
NIS	Nota Informativa dello Schema
OCSI	Organismo di Certificazione della Sicurezza Informatica
OS	Operating System
PP	Protection Profile
POJO	Plain Old Java Object
REST	Representational State Transfer
RHEL	Red Hat Enterprise Linux
RHN	Red Hat Network
RMI-IIOP	Remote Method Invocation over Internet Inter-Orb Protocol
RPC	Remote Procedure Call
RPM	Red Hat Package Manager
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security

TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
URL	Uniform Resource Locator
VFS	Virtual File System
VM	Virtual Machine

4 References

4.1 Criteria and regulations

[CC1] CCMB-2017-04-001, “Common Criteria for Information Technology Security Evaluation, Part 1 – Introduction and general model”, Version 3.1, Revision 5, April 2017

[CC2] CCMB-2017-04-002, “Common Criteria for Information Technology Security Evaluation, Part 2 – Security functional components”, Version 3.1, Revision 5, April 2017

[CC3] CCMB-2017-04-003, “Common Criteria for Information Technology Security Evaluation, Part 3 – Security assurance components”, Version 3.1, Revision 5, April 2017

[CCRA] “Arrangement on the Recognition of Common Criteria Certificates In the field of Information Technology Security”, July 2014

[CEM] CCMB-2017-04-004, “Common Methodology for Information Technology Security Evaluation – Evaluation methodology”, Version 3.1, Revision 5, April 2017

[LGP1] Schema nazionale per la valutazione e certificazione della sicurezza di sistemi e prodotti nel settore della tecnologia dell’informazione - Descrizione Generale dello Schema Nazionale - Linee Guida Provvisorie - parte 1 – LGP1 versione 1.0, Dicembre 2004

[LGP2] Schema nazionale per la valutazione e certificazione della sicurezza di sistemi e prodotti nel settore della tecnologia dell’informazione - Accreditamento degli LVS e abilitazione degli Assistenti - Linee Guida Provvisorie - parte 2 – LGP2 versione 1.0, Dicembre 2004

[LGP3] Schema nazionale per la valutazione e certificazione della sicurezza di sistemi e prodotti nel settore della tecnologia dell’informazione - Procedure di valutazione - Linee Guida Provvisorie - parte 3 – LGP3, versione 1.0, Dicembre 2004

[NIS1] Organismo di certificazione della sicurezza informatica, Nota Informativa dello Schema N. 1/13 – Modifiche alla LGP1, versione 1.0, Novembre 2013

[NIS2] Organismo di certificazione della sicurezza informatica, Nota Informativa dello Schema N. 2/13 – Modifiche alla LGP2, versione 1.0, Novembre 2013

[NIS3] Organismo di certificazione della sicurezza informatica, Nota Informativa dello Schema N. 3/13 – Modifiche alla LGP3, versione 1.0, Novembre 2013

4.2 Technical documents

[CCGUIDE] "Red Hat JBoss Enterprise Application Platform 7.2.3 Common Criteria Configuration Guide", 9 October 2019

[ETR] Final Evaluation Technical Report "JBoss Enterprise Application Platform 7.2", OCSI_CERT_ATS_05_2018_ETR_191017_v1.1, Version 1.1, atsec information security GmbH, 17 October 2019

[ST] JBoss Enterprise Application Platform 7 Version 7.2.3 Security Target, Version 1.9, Red Hat, Inc., 5 November 2019

5 Recognition of the certificate

5.1 International Recognition of CC Certificates (CCRA)

The current version of the international arrangement on the mutual recognition of certificates based on the CC (Common Criteria Recognition Arrangement, [CCRA]) was ratified on 08 September 2014. It covers CC certificates compliant with collaborative Protection Profiles (cPP), up to and including EAL4, or certificates based on assurance components up to and including EAL2, with the possible augmentation of Flaw Remediation family (ALC_FLR).

The current list of signatory nations and of collaborative Protection Profiles (cPP) and other details can be found on http://www.commoncriteriaportal.org.

The CCRA logo printed on the certificate indicates that it is recognised under the terms of this agreement by signatory nations.

This certificate is recognised under CCRA up to EAL2.

6 Statement of Certification

The Target of Evaluation (TOE) is the product “JBoss Enterprise Application Platform 7 Version 7.2.3” (JBoss EAP for short), developed by Red Hat, Inc.

JBoss EAP is an application server based on Java Enterprise Edition (Java EE) and therefore supports a large variety of operating systems. JBoss EAP allows client computers or devices to access Java applications through different network protocols. JBoss EAP handles the business logic of the application, including accessing and providing the user data required by the application.

The evaluation has been conducted in accordance with the requirements established by the Italian Scheme for the evaluation and certification of security systems and products in the field of information technology and expressed in the Provisional Guidelines [LGP1, LGP2, LGP3] and Scheme Information Notes [NIS1, NIS2, NIS3]. The Scheme is operated by the Italian Certification Body “Organismo di Certificazione della Sicurezza Informatica (OCSI)”, established by the Prime Minister Decree (DPCM) of 30 October 2003 (O.J. n.98 of 27 April 2004).

The objective of the evaluation is to provide assurance that the product complies with the security requirements specified in the associated Security Target [ST]; the potential consumers of the product should review also the Security Target, in addition to the present Certification Report, in order to gain a complete understanding of the security problem addressed. The evaluation activities have been carried out in accordance with the Common Criteria Part 3 [CC3] and the Common Evaluation Methodology [CEM].

The TOE resulted compliant with the requirements of Part 3 of the CC v 3.1 for the assurance level EAL4, augmented with ALC_FLR.3, according to the information provided in the Security Target [ST] and in the configuration shown in Annex B – Evaluated configuration of this Certification Report.

The publication of the Certification Report is the confirmation that the evaluation process has been conducted in accordance with the requirements of the evaluation criteria Common Criteria - ISO/IEC 15408 ([CC1], [CC2], [CC3]) and the procedures indicated by the Common Criteria Recognition Arrangement [CCRA] and that no exploitable vulnerability was found. However, the Certification Body with such a document does not express any kind of support or promotion of the TOE.

7 Summary of the evaluation

7.1 Introduction

This Certification Report states the outcome of the Common Criteria evaluation of the product "JBoss Enterprise Application Platform 7 Version 7.2.3" to provide assurance to the potential consumers that TOE security features comply with its security requirements.

In addition to the present Certification Report, the potential consumers of the product should review also the Security Target [ST], specifying the functional and assurance requirements and the intended operational environment.

7.2 Executive summary

TOE name	JBoss Enterprise Application Platform 7 Version 7.2.3
Security Target	JBoss Enterprise Application Platform 7 Version 7.2.3 Security Target, Version 1.9 [ST]
Evaluation Assurance Level	EAL4 augmented with ALC_FLR.3
Developer	Red Hat, Inc.
Sponsor	Red Hat, Inc.
LVS	atsec information security GmbH
CC version	3.1 Rev. 5
PP conformance claim	No compliance declared
Evaluation starting date	11 June 2018
Evaluation ending date	17 October 2019

The certification results apply only to the version of the product shown in this Certification Report and only if the operational environment assumptions described in the Security Target [ST] are met.

7.3 Evaluated product

This paragraph summarizes the main functional and security features of the TOE; for a detailed description, refer to the Security Target [ST].

The TOE is the JBoss Enterprise Application Platform (EAP) comprising the following components:

- JBoss EAP 7.2.3
- JBoss Core Services OpenSSL version 1.0.2n

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

The TOE implements an application server based on Java Enterprise Edition (Java EE) and therefore supports a large variety of operating systems. As an application server, JBoss EAP 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 EAP 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 EAP instance. If a cluster of JBoss EAP nodes is defined, then the entire cluster is defined as one TOE.

The TOE provides identification and authentication of users, access control for various types of objects, audit functionality, clustering, transaction rollback, and role-based access control to administrative operations and resources.

The TOE security functions are described more in detail in section 7.3.2.3.

7.3.1 TOE Architecture

7.3.1.1 TOE general overview

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

JBoss EAP is written entirely in Java (with the exception of JBoss Core Services OpenSSL) and provides a Java EE-compliant environment, which is consistent with the Java EE 7 specification. Depending on the configuration of the JBoss EAP server, components required by the Java EE specification can be disabled. The applications developed for and served by JBoss EAP are to be written in Java. Developers of such Java applications implement the business logic and are free to utilize the supporting functionality of Java EE as provided by JBoss EAP.

The configuration of JBoss EAP allows selectively enabling or disabling every container, known as extensions in JBoss EAP. The distribution of JBoss EAP provides a number of extensions that can be utilized, but additional ones may be implemented by third parties. The evaluated configuration defines the extensions, which are covered by the evaluation and therefore may be enabled in a CC-compliant configuration.

The JBoss EAP architecture, shown in Figure 1, provides the environment for the execution of different containers, which allow applications to utilize services provided by these containers. The JBoss EAP framework uses a different Java class loader for each module. Applications executing within JBoss EAP 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 EAP framework allows the establishment of links between modules.

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 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 JBoss Remoting.

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 – those protocols are provided by different JBoss EAP 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 in the Security Target [ST] are enforced on the above-mentioned outer layer protocols.

7.3.1.2 JBoss EAP structure

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

JBoss EAP 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 EAP 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 four layers, also called tiers, listed in Table 1. Applications utilizing the Java EE specification may implement any combination of these tiers. In addition to listing the tiers, Table 1 specifies which tiers can be implemented and executed using the framework of JBoss EAP.

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 Web browser • Javascript code executed by the client's Web 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 EAP application server, but they are executed by the Java Virtual Machine of the client system accessing the JBoss EAP information remotely. Therefore, the client tier is considered not to be covered by JBoss.
Web tier The web tier is the presentation layer of the application server. It gathers the business information from the lower EJB tier and converts it to be presented as web pages. The web tier therefore does not implement any business logic as it can be considered an information converter from the application-internal data representation to a user-viewable and user-interpretable presentation.	The web tier can be implemented using Java servlets executing within the JBoss framework. 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.	The business tier can be implemented using various types of EJBs executing within the JBoss EAP framework. JBoss EAP also supports the implementation of the business logic as POJOs, 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 EJB 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 implemented by the TOE.

Table 1 - Java EE tier listing and JBoss coverage

Fundamentally in the JBoss EAP 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 EAP by writing new MBeans. In addition, POJOs configured as services can be created for either extending the JBoss EAP functionality or implementing business logic.

Figure 1 shows the interoperation of the different components of JBoss EAP. JBoss EAP consists of a modular framework where the administrator can selectively enable components. JBoss EAP offers compliance with the Java EE 7 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 EAP containers perform their tasks.
- Each container implements either a service as specified in Java EE 7 or a service providing additional functionality beyond Java EE 7.
- 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.

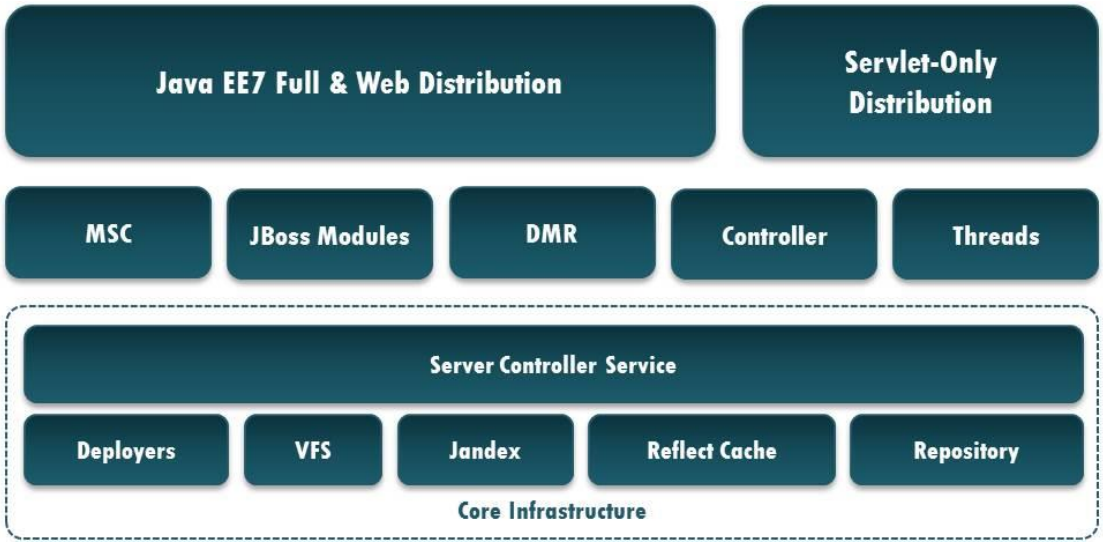

Figure 1 - JBoss EAP 7 Architecture

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

- HTTP web network protocol
- RESTful (JAX-RS) and XML-Based (JAX-WS) Web Services
- Enterprise Java Beans (EJB)
- Java Messaging Service (JMS)
- Java Naming and Directory Interface (JNDI)

Applications utilize the services provided by the different containers by accessing the API exported by each container. These applications are loaded and executed by either the JSP/Servlet container, EJB container or other containers. The technical separation of the untrusted applications and the TOE is achieved by using the Java Security Manager with an appropriate policy configuration.

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

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.

7.3.2 TOE security features

7.3.2.1 Security policy

The security policy enforced is defined by the selected set of Security Functional Requirements (SFRs) and implemented by the TOE. It covers the following security aspects:

- **Auditing**: based on the Audit Policy, the TOE monitors access of users and administrators to the system. The extent and detail of the auditing is configurable.
- **Identification & Authentication**: all users of the TOE are identified and authenticated, based on the user databases maintained by the TOE. The authentication considers user names, authentication credentials, and groups membership.
- **Access control**: access to TOE objects is protected by requiring identification and authentication of users. Authorized users are allowed to specify which resources may be accessed by which users. The TOE supports different types of access control policies.
- **Role-based management**: the TOE allows access to administrative resources as well as administrative operations based on the role a user is assigned to. Authorized users are allowed to specify which resources may be accessed by which users.

- **Consistency of State**: the TSF ensures the consistency of user data as well as TSF data while it is being processed. Consistency is ensured when data is processed that may be located in instances of the TOE.

7.3.2.2 Operational environment security objectives

The assumptions for the correct operation of the TOE defined in the Security Target [ST] and some aspects of Threats and Organisational Security Policies are not covered by the TOE. These aspects lead to specific security objectives to be fulfilled by the TOE operational environment. The following objectives for the operational environment have to be assured:

- 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.

- 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 the ST are used as underlying platform to ensure that proper date and time information is available to the audit facility.

- 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.

- 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.

- 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.

- 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.

For a complete description of the security objectives for the TOE operational environment, please refer to section 4.2 of the Security Target [ST].

7.3.2.3 Security functions

The TOE security functionality is described in detail in sect. 7.1 (TOE Security Functionality) of the Security Target [ST]. The most significant aspects are summarized in the following:

- **Access Control**: using access control, the TOE is able to restrict access for the following request types with the following access control mechanisms:
    - EJB: EJBs and associated method names can be protected from being called by subjects.

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

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 deployment descriptors and annotations can be used by authorized administrators and application developers.

- **Role-based access control for management interfaces**: the management interfaces of JBoss EAP, the command line interface as well as the web-based administrative interface, allow access to the JBoss EAP 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.
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 administrative LAN to prevent untrusted users from technically accessing the software 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 categorized into:

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

  The objects are categorized based on the following:

  - a resource;
  - an attribute residing in a resource.

  A set of object-action capabilities is mapped to a management role. This mapping defines the allowed access for this management role. A set of predefined roles is shipped with the TOE and is available after installation.
  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).

- **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. Those 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.

- **Clustering**: a cluster is a set of nodes. In a JBoss EAP cluster, a node is a JBoss EAP server instance. Thus, to build a cluster, several JBoss EAP 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 EAP: 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).
    - Replication of the state of a node covering the replication of HTTP sessions, and EJB 2.x session beans.
    - Replication of JMS queues.

- **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 Web Services: BASIC, FORM, DIGEST and CLIENT_CERT authentication.
    - EJB: username and password-based authentication, client-certificate-based identification.
    - JMS: username and password-based authentication.

  For identification and authentication using a client certificate, the TOE uses the underlying TLS channel established by the operational environment (either JDK SSL or OpenSSL). The underlying TLS protocol performs the certificate validation of the client certificate. The EJB component of the TOE queries the TLS session for the DN part of the certificate to identify the user. That DN information is used to set up the role mapping and to create a principal in the TOE. Therefore, the TOE relies on the TLS implementation in the operational environment to perform authentication by enforcing the validation of the client certificate.
  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.

- **Transaction Rollback**: JBoss EAP 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 to be 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 using 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 add 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 EAP is designed to support multiple coordination protocols. JBoss EAP 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 service transactions component can be utilized, which uses SOAP/HTTP.

7.4 Documentation

The guidance documentation specified in Annex A - Guidelines for secure usage of the TOE is delivered to the customer together with the product. The guidance documentation contains all the information for installation, configuration and secure usage of the TOE in accordance with the requirements of the Security Target [ST].

Customers should also follow the recommendations for the secure usage of the TOE contained in sect. 8.2 of this report.

7.5 Protection Profile conformance claims

The Security Target [ST] does not claim conformance to any Protection Profile.

7.6 Functional and assurance requirements

All Security Assurance Requirements (SAR) have been selected from CC Part 3 [CC3]. Namely, the requirements of EAL4 augmented by ALC_FLR.3 have been met.

All Security Functional Requirements (SFRs) have been selected or derived by extension from CC Part 2 [CC2]. In particular, the extended component FDP_ROL.2-jb has been defined, which provides the capability for the TOE to perform an automated rollback of all the operations that form one transaction when at least one operation part of the transaction fails. For a detailed description of the extended components properties, consult section 5 of the Security Target [ST].

Users should refer to the Security Target [ST] for a complete description of all security objectives, the threats that these objectives should address, the Security Functional Requirements (SFR) and the security functions that realize the same objectives.

7.7 Evaluation conduct

The evaluation has been conducted in accordance with the requirements established by the Italian Scheme for the evaluation and certification of security systems and products in the field of information technology and expressed in the Provisional Guideline [LGP3] and the Scheme Information Note [NIS3] and in accordance with the requirements of the Common Criteria Recognition Arrangement [CCRA].

The purpose of the evaluation is to provide assurance on the effectiveness of the TOE to meet the requirements stated in the relevant Security Target [ST]. Initially the Security Target has been evaluated to ensure that it constitutes a solid basis for an evaluation in accordance with the requirements expressed by the standard CC. Then, the TOE has been evaluated on the basis of the statements contained in such a Security Target. Both phases of the evaluation have been conducted in accordance with the CC Part 3 [CC3] and the Common Evaluation Methodology [CEM].

The Certification Body (OCSI) has supervised the conduct of the evaluation performed by the evaluation facility (LVS) atsec information security GmbH.

The evaluation was completed on 17 October 2019 with the issuance by LVS of the Evaluation Technical Report [ETR], which was approved by the Certification Body on 29 October 2019. Then, the Certification Body issued this Certification Report.

7.8 General considerations on the validity of the certification

The evaluation focused on the security features declared in the Security Target [ST], with reference to the operational environment specified therein. The evaluation has been performed on the TOE configured as described in Annex B – Evaluated configuration.

Potential customers are advised to check that this corresponds to their own requirements and to pay attention to the recommendations contained in this Certification Report.

The certification is not a guarantee that no vulnerabilities exist; it remains a probability (the smaller, the higher the assurance level) that exploitable vulnerabilities can be discovered after the issuance of the certificate. This Certification Report reflects the conclusions of the certification at the time of issuance. Potential customers are invited to check regularly the arising of any new vulnerability after the issuance of this Certification Report, and if the vulnerability can be exploited in the operational environment of the TOE, check with the Developer if security updates have been developed and if those updates have been evaluated and certified.

8 Evaluation outcome

8.1 Evaluation results

Following the analysis of the Evaluation Technical Report [ETR], issued by the LVS atsec information security GmbH, and the documents required for the certification, and considering the evaluation activities which was carried out, the Certification Body (OCSI) concluded that TOE "JBoss Enterprise Application Platform 7 Version 7.2.3" meets the requirements of Part 3 of the Common Criteria [CC3] provided for the evaluation assurance level EAL4, augmented with ALC_FLR.3, with respect to the security features described in the Security Target [ST] and the evaluated configuration, shown in Annex B – Evaluated configuration.

Table 2 summarizes the final verdict of each activity carried out by the LVS in accordance with the assurance requirements established in [CC3] for the evaluation assurance level EAL4, augmented with ALC_FLR.3.

Assurance classes and components		**Verdict**
Security Target evaluation	**Class ASE**	Pass
Conformance claims	ASE_CCL.1	Pass
Extended components definition	ASE_ECD.1	Pass
ST introduction	ASE_INT.1	Pass
Security objectives	ASE_OBJ.2	Pass
Derived security requirements	ASE_REQ.2	Pass
Security problem definition	ASE_SPD.1	Pass
TOE summary specification	ASE_TSS.1	Pass
Development	**Class ADV**	Pass
Security architecture description	ADV_ARC.1	Pass
Complete functional specification	ADV_FSP.4	Pass
Implementation representation of the TSF	ADV_IMP.1	Pass
Basic modular design	ADV_TDS.3	Pass
Guidance documents	**Class AGD**	Pass
Operational user guidance	AGD_OPE.1	Pass
Preparative procedures	AGD_PRE.1	Pass
Life cycle support	**Class ALC**	Pass
Production support, acceptance procedures and automation	ALC_CMC.4	Pass
Problem tracking CM coverage	ALC_CMS.4	Pass
Delivery procedures	ALC_DEL.1	Pass

Assurance classes and components		Verdict
Identification of security measures	ALC_DVS.1	Pass
Developer defined life-cycle model	ALC_LCD.1	Pass
Well-defined development tools	ALC_TAT.1	Pass
Systematic flaw remediation	*ALC_FLR.3*	Pass
Tests	**Class ATE**	Pass
Analysis of coverage	ATE_COV.2	Pass
Testing: basic design	ATE_DPT.1	Pass
Functional testing	ATE_FUN.1	Pass
Independent testing - sample	ATE_IND.2	Pass
Vulnerability assessment	**Class AVA**	Pass
Focused vulnerability analysis	AVA_VAN.3	Pass

Table 2 - Final verdicts for assurance requirements

8.2 Recommendations

The conclusions of the Certification Body (OCSI) are summarized in section 6 (Statement of Certification).

Potential customers of the product “JBoss Enterprise Application Platform 7 Version 7.2.3” are suggested to properly understand the specific purpose of the certification reading this Certification Report together with the Security Target [ST].

The TOE must be used according to the Security Objectives for the operational environment specified in section 4.2 of the Security Target [ST]. Potential customers are advised to check that they meet the identified requirements and to pay attention to the recommendations contained in this Report.

This Certification Report is valid for the TOE in its evaluated configuration; in particular, Annex A - Guidelines for secure usage of the TOE includes a number of recommendations relating to delivery, initialization, configuration and secure usage of the product, according to the guidance documentation provided together with the TOE ([CCGUIDE]).

It is assumed that the TOE operates securely if the assumptions about the operational environment described in sect. 3.2 of the Security Target [ST] are satisfied. In particular, it is assumed that the administrators of the TOE are adequately trained to the correct usage of the TOE and chosen among the trusted personnel of the organization. The TOE is not realized to counter threats from unexperienced, malicious or negligent administrators.

It should also be noted that TOE security is conditioned by the proper functioning of the software and hardware platforms on which the TOE is installed, and of all trusted external IT systems supporting the implementation of TOE’s security policy. Specifications for the operational environment are described in the Security Target [ST].

9 Annex A - Guidelines for secure usage of the TOE

This Annex provides considerations particularly relevant to the potential customers of the TOE.

9.1 TOE delivery

The TOE is software only and is accompanied by guidance documentation. The TOE is made up of components distributed as RPM packages, which are compiled into an ISO image for easy retrieval or as ZIP archive available via the Red Hat Customer Portal.

Table 3 contains the items that comprise the different elements of the TOE, including software and guidance.

No.	Type	Identifier	Release	Form of Delivery
1	SW	JBoss ZIP Archive	7.2.3	Electronic
2	SW	JBoss ISO Image	7.2.3	Electronic
3	DOC	JBoss Enterprise Application Platform 7.2.3 Common Criteria 7.2.3 Configuration Guide [CCGUIDE]	7.2.3	Electronic
4	DOC	Installation Guide Red Hat JBoss Enterprise Application Platform 7.2 Getting Started Guide Red Hat JBoss Enterprise Application Platform 7.2 Security Architecture Red Hat JBoss Enterprise Application Platform 7.2 How to Configure Server Security Red Hat JBoss Enterprise Application Platform 7.2 How to Configure Identity Management Red Hat JBoss Enterprise Application Platform 7.2 Configuring Messaging Red Hat JBoss Enterprise Application Platform 7.2 Development Guide Red Hat JBoss Enterprise Application Platform 7.2 Developing EJB Applications Red Hat JBoss Enterprise Application Platform 7.2 Developing Web Services Applications GA Public API JavaDocs Red Hat JBoss Enterprise Application Platform 7.2 Management CLI Guide	7.2	Electronic

Table 3 – TOE deliverables

The Developer indicated that the distinction between the two delivery methods (ISO or ZIP) is simply dependent on the chosen customer's operating system. In other words, customers who use Red Hat Enterprise Linux (i.e., JBoss EAP subscribers) can pick the

RPM method while customers using another platform (e.g., Microsoft Windows) have to select the ZIP archive method.

The Developer's release process defines responsibilities of different organizations within Red Hat Network as follows:

- Release Engineering maintains CM tools and build infrastructure for building product deliverables.
- Quality Engineering assesses and ensures product quality.
- Security Response Team reviews, identifies, and monitors "security vulnerable" product areas.
- Program Management consults with the other two organizations about the product readiness for release, and makes the decision to release.

The build process is defined as follows:

1. The Release Engineering organization builds the product distribution components in a "controlled build environment that is carefully monitored to avoid pollution by external code."
2. The Release Engineering organization signs the RPM build components with Red Hat GNU Privacy Guard (GPG) private keys. The corresponding public key is available on both redhat.com and the public key server pgp.mit.edu.
3. Substantial quality assurance activities are carried out on a release candidate version of components, which includes installing and functional testing on all supported platforms and checking that digital signatures have been generated.
4. The Quality Engineering team notifies the Product Management and Release Engineering groups (via email or at Program Management meetings) that the product is ready for distribution.
5. The Release Engineering organization then prepares the components for customer distribution, which includes generating SHA-256 checksums for all files, recording those checksums on a secure system (managed by the Release Engineering organization), and finally transferring the components via SSH to Red Hat customer distribution centers, where customers can download them via CP or RHN distribution channels.

The Red Hat distribution servers are located at multiple secure third-party facilities, which are only accessible to Red Hat personnel and authorized contractors who have proper agreements with Red Hat and who are also typically escorted by Red Hat personnel.

9.2 Identification of the TOE

The cover pages of every guidance document shows the TOE version number as 7.2. In the Red Hat Customer Portal download area, the JBoss EAP version of 7.2.3 may also be referred to as 7.2 CP03 as both reference types are equivalent.

When executing the TOE, the server log shows the following information about the TOE:

```
INFO [org.jboss.as] (MSC service thread 1-2) WFLYSRV0049: JBoss EAP
7.2.3.GA (WildFly Core 6.0.15.Final-redhat-00001)

INFO [org.wildfly.security] (ServerService Thread Pool -- 27)
ELY00001: WildFly Elytron version 1.6.3.Final-redhat-00001
```

The JBoss EAP CC Configuration Guide [CCGUIDE] in section “Confirming the Version of your JBoss Enterprise Application Platform Installation” provides three ways to verify the version number of the installed TOE.

9.3 Installation, initialization and secure usage of the TOE

TOE installation and configuration should be done following the instructions in the appropriate sections of the guidance documentation provided with the product to the customer, listed in items 3 and 4 of Table 3.

In particular, the JBoss EAP CC Configuration Guide [CCGUIDE] contains information for the secure initialization of the TOE and the preparation of its operational environment in accordance with the security objectives specified in the Security Target [ST].

10 Annex B – Evaluated configuration

The TOE is Red Hat JBoss Enterprise Application Platform (EAP) version 7.2.3. The TOE is software only and is accompanied by guidance documentation. The items listed in Table 3 represent the TOE.

The evaluated configuration is documented in the JBoss EAP CC Configuration Guide [CCGUIDE]. This document specifies a number of constraints. The description includes the following information:

- usable SQL databases and the applicable JDBC drivers;
- combination of allowed JDKs and operating systems to be used;
- restrictions on the configuration of Elytron and the reference of the allowed user credential stores;
- set up of the auditing functionality to meet the requirements specified in the Security Target [ST].

11 Annex C –Test activities

This Annex describes the effort of both Developer and LVS in testing activities. For the assurance level EAL4, augmented with ALC_FLR.3, such activities include the following three steps:

- evaluation of the tests performed by the Developer in terms of coverage and level of detail;
- execution of independent functional tests by the Evaluators;
- execution of penetration tests by the Evaluators.

11.1 Test configuration

The testing of the TOE were performed several times with different configuration constraints.

The following constraints were considered by the Developer:

- the testing was executed with the Java Security Manager and its well-defined policy enabled;
- testing was performed on all JDKs specified in the ST;
- all user account data stores allowed in the ST were covered by the tests;
- the different databases listed in the ST were used as a database backend;
- user data store of LDAP, database and properties files are tested.

Testing was performed on the TOE version specified in [CCGUIDE] and [ST]. Additionally, the test environments/platforms were configured to be compliant with requirements of the evaluated configuration as dictated in [CCGUIDE] and [ST]. Therefore, the testing configurations meet the configuration requirements for the evaluated configuration.

As part of independent test, the Evaluators installed the TOE using the [CCGUIDE] and product installation documentation. The test cases are prepared as described in the Developer test plan.

The Evaluators verified the test system used for the re-performing of the Developer tests as follows:

- JRE version: OpenJDK
- OS version: RHEL 7
- DB version: MariaDB 10.1
- LDAP version: Active Directory 2016

The Evaluators verified the system used for the execution of the independent Evaluator tests as follows:

- JRE version: Oracle JDK 1.8
- OS version: Windows Server 2016
- DB version: MariaDB 10.1
- LDAP version: Active Directory 2016

11.2 Functional tests performed by the Developer

11.2.1 Testing approach

The test mapping document provided by the Developer lists the tree of test suites which comprises of test cases which in turn comprise of the test units. This mapping document also provides the ability to trace the individual test unit back to the interfaces that the test unit covers.

The tests are written in Java and are completely automated and available from the Developer.

The Evaluators note that these test cases are developed upstream in conjunction with the JBoss TOE source code. The tests include applications which are loaded onto the TOE as well as user programs which try to access the applications by interfacing with the TOE.

The test cases contain information about the desired/expected behaviors and validates whether the TOE acts according to the expected behavior(s). If the TOE acts as expected, a pass result is returned to the test framework, otherwise, a fail is returned. The test framework records and collects the test results and present them in human-readable HTML files.

11.2.2 Test coverage

The test case mapping identifies the interfaces to which the test cases map. The following types of TSFI are covered by the tests:

- Network protocols enforcing access control and identification and authentication configurations.
- Source code annotations for configuring access control functionality.
- Configuration files and deployment descriptors for the configuration of the identification and authentication and access control functionality. In addition, transaction support is tested using deployment descriptors.
- The command line interface is indirectly covered by starting the TOE in two different modes of operation, which can only be done using appropriate command line switches.

The test depth, i.e., the coverage of all subsystems implementing SFR-enforcing functionality, is also provided by the same tests for test coverage. The test mapping document maps test cases to applicable subsystems. The test depth analysis shows that the test cases not only cover the subsystems they invoke directly but also the subsystems that can only be triggered indirectly such as Elytron.

11.2.3 Test results

The test results provided by the Developer were generated on the JDK platforms and configurations listed in sect. 11.1. As described in the testing approach, the test results for all these automated tests are recorded and collected by the framework and written to HTML files and Jenkins log files.

All test results from all tested configurations show that the expected test results are consistent with the actual results.

11.3 Functional and independent tests performed by the Evaluators

The Evaluators testing effort consists of two parts: observation of the Developer test execution and execution of the tests created by the Evaluators.

The test system was set up as stated in sect. 11.1. When rerunning the Developer tests using one specific test scenario configuration executed by the Developer, the Evaluators used the Developer test plan to set up and initiate these tests. All tests were executed successfully and test results were recorded in a test result file.

In addition to repeating all Developer tests, the Evaluators devised tests for a subset of the TOE functionality.

The tests were chosen by the Evaluators based on the following reasons:

- Audit configuration in the evaluated configuration adds an additional audit trail file.
- A large number of different interfaces are invoked by the Developer testing.
- Different access control functions are covered by the Developer testing.
- As the Developer test cases already cover the central TOE functions with a large number of tests, the Evaluators focused on minor security functionality that was covered lightly by the Developer testing.
- The HTTP HEAD access type was not covered in the TOE testing for verifying the access control enforcement for HTTP connections.

The Evaluators created his own test cases expanding the functional aspects of auditing and HTTP access control. Through examination of the Developer test cases, the Evaluators gained sufficient confidence in the Developer test effort as well as coverage. The Developer tests were shown to demonstrate a very wide coverage of the TSF, therefore, the Evaluators decided to devise only a small number of test cases.

In addition to running the Developer tests, the Evaluators devised independent tests. These tests cover the following functional areas:

- Auditing: different tests were executed covering different functional areas of the TOE to verify that appropriate audit records are created and maintained by the TOE for the access requests.

- HTTP access control: the Evaluators tested the enforcement of the HTTP access control policy on the HEAD HTTP request type.

All tests passed successfully.

11.4 Vulnerability analysis and penetration tests

11.4.1 Testing approach

First the Evaluators checked common sources for vulnerabilities of the JBoss server in general and the TOE in particular. The Evaluators determined:

- Whether the reported vulnerability would affect the evaluated configuration of the TOE in its intended environment. If yes, the Evaluators performed a vulnerability analysis.

- Whether the reported vulnerability has already been fixed in the evaluated configuration of the TOE. If the reported vulnerability does not have a fix, the Evaluators analyzed the potential impact and exploitability.

Beside those vulnerabilities reported in common sources, the Evaluators checked other evaluation reports for potential vulnerabilities mentioned within those reports. For those vulnerabilities, the Evaluators devised the way to check for the existence or absence of the hypothetical vulnerability, while taking into account that the TOE is an Open Source product and so the Evaluator had full access to the source code.

Based on the vulnerability analysis, the Evaluators conducted testing in the following areas:

- Verification of the effectiveness of access control of a typically unused and rarely known HTTP request type of HEAD.

- Verification that shared components maintaining sensitive information do not leak them.

11.4.2 Test coverage

Although the Evaluators decided to only generate a small number of penetration tests, for some of the identified potential vulnerabilities, the Evaluators performed a very extensive analysis exceeding the requirements of EAL4 claimed by the TOE. The reasons are as follows:

- The TOE as an open source product is already subject to the scrutiny of obvious vulnerabilities by the Open Source community. Yet, this consideration is not taken as a guarantee of the absence of vulnerabilities.

- The TOE as an open source product is delivered with full source code, thus, allowing the Evaluators the means to perform an extensive analysis which usually considered inconceivable for products evaluated at an EAL4 assurance level. In general, the Evaluators considered source code review as a more effective method for vulnerability analysis than testing. Due to the nature of vulnerabilities, a perceived vulnerability is usually obscure in reality and therefore can only be exploitable when meeting certain constraints. Testing may not cover all constraints (as certain constraints are not fully defined or known to testers), thus, a test yielding no vulnerability does not necessarily demonstrate that no vulnerability is present.

11.4.3 Test results

The Evaluators performed all penetration tests on a TOE that was installed and configured according to the JBoss EAP CC Configuration Guide [CCGUIDE].

The penetration testing addressed the following security functionalities:

- Non-bypassability of TOE security functions.

No vulnerability was detected that is exploitable in the intended operational environment of the TOE by attackers with an assumed attack potential of at most Enhanced-Basic.

The Evaluators also identified no residual vulnerabilities.