# National Information Assurance Partnership


TM

# Common Criteria Evaluation and Validation Scheme Validation Report

# MarkLogic Server Enterprise Edition Version 6.0-4

**Report Number:** CCEVS-VR-VID10507-2013
**Dated:** 19 December 2013
**Version:** 1.0

**National Institute of Standards and Technology**
**Information Technology Laboratory**
**100 Bureau Drive**
**Gaithersburg, MD 20899**

**National Security Agency**
**Information Assurance Directorate**
**9800 Savage Road STE 6740**
**Fort George G. Meade, MD 20755-6740**

# ACKNOWLEDGEMENTS

**Validation Team**

Patrick Mallett, Lead Validator, MITRE
Bradford O'Neil, Senior Validator, MITRE

**Common Criteria Testing Laboratory**

Katie Sykes, Lead Evaluator
Dawn Campbell, Lead Evaluator
Leidos Inc.
Columbia, Maryland

# Table of Contents

# 1 Executive Summary

This report documents the National Information Assurance Partnership (NIAP) assessment of the evaluation of the MarkLogic Server Enterprise Edition Version 6.0-4.

The Validation Report presents the evaluation results, their justifications, and the conformance results. This Validation Report is not an endorsement of the Target of Evaluation (TOE) by any agency of the U.S. Government and no warranty of the TOE is either expressed or implied.

The evaluation of MarkLogic Server Enterprise Edition Version 6.0-4 was performed by Leidos Inc. Common Criteria Testing Laboratory in the United States and was completed on 19 December 2013.

The information in this report is largely derived from the Security Target (ST), Evaluation Technical Report (ETR) and associated test report. The ST was written by LEIDOS. The ETR and Team Test Report used in developing this validation report were written by LEIDOS. The evaluation team determined the product to be Part 2 extended and Part 3 conformant, and meets the assurance requirements of EAL 2 augmented with ALC_FLR.3. All security functional requirements are derived from Part 2 of the Common Criteria.

The TOE is MarkLogic Server Enterprise Edition Version 6.0-4 provided by Mark Logic Corporation. MarkLogic Server Enterprise Edition Version 6.0-4 is an enterprise-class database or “contentbase” that provides a set of services used to build both content and search applications which query, manipulate and render XML content.

The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced.

During this validation, the Validators determined that the evaluation showed that the product satisfied all of the functional requirements and assurance requirements defined in the Security Target (ST). Therefore, the Validator concludes that the LEIDOS findings are accurate, the conclusions justified, and the conformance claims correct.

# 2 Identification

The CCEVS is a joint National Security Agency (NSA) and National Institute of Standards and Technology (NIST) effort to establish commercial facilities to perform trusted product evaluations. Under this program, commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) using the Common Evaluation Methodology (CEM) for Evaluation Assurance Level (EAL) 1 through EAL 4 in accordance with National Voluntary Laboratory Assessment Program (NVLAP) accreditation.

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products, desiring a security evaluation, contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to the Product Compliant List.

Table 1 provides information needed to completely identify the product, including:

- The Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated;
- The Security Target (ST), describing the security features, claims, and assurances of the product;
- The conformance result of the evaluation;
- The Protection Profile to which the product is conformant; and
- The organizations and individuals participating in the evaluation.

**Table 1: Evaluation Identifiers**

| Item | Identifier |
|---|---|
| **Evaluation Scheme** | United States NIAP Common Criteria Evaluation and Validation Scheme |
| **TOE**: | MarkLogic Server Enterprise Edition Version 6.0-4 |
| **Protection Profile** | None |
| **ST**: | MarkLogic Server Enterprise Edition Version 6.0 Security Target, Version 1.0, December XX, 2013 |
| **Evaluation Technical Report** | Evaluation Technical Report for MarkLogic Server Enterprise Edition Version 6.0, Part 1 (Non-Proprietary), Version 1.0, 16 December 2013, Part 2 (Proprietary), Version 1.0, 16 December 2013. |
| **CC Version** | Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012 |

| Item | Identifier |
|---|---|
| **Conformance Result** | CC Part 2 extended and Part 3 conformant, EAL 2 augmented with ALC_FLR.3 |
| **Sponsor** | Mark Logic Corporation |
| **Developer** | Mark Logic Corporation |
| **Common Criteria Testing Lab (CCTL)** | Leidos Inc., Columbia, MD |
| **CCEVS Validators** | Patrick Mallett, The MITRE Corporation<br>Bradford O'Neil, The MITRE Corporation |

# 3 TOE Overview

The MarkLogic Server TOE is built with a blend of search engine and database architecture approaches specifically designed to index and retrieve XML content. The TOE's native data format is XML and XML is accepted in an 'as is' form, while content in other formats can be converted to an XML representation or stored as is (in binary or text formats) when loaded into MarkLogic Server. As an XML database, MarkLogic Server manages its own content repository and is accessed using the W3C standard XQuery language, just as a relational database is a specialized server that manages its own repository and is accessed through Structured Query Language (SQL).

The TOE is fully transactional, runs in a distributed environment and can scale to terabytes of indexed content. It is schema independent and all loaded documents can be immediately queried without normalizing the data in advance. MarkLogic Server provides developers with the functionality and programmability, using XQuery as its query language, to build content-centric applications. Developers build applications using XQuery both to search the content and as a programming language in which to develop applications. It is possible to create entire applications using only MarkLogic Server, and programmed entirely in XQuery. Application can also be created using Java or other programming languages that access MarkLogic Server.

The security management functions of the TOE are performed via the Admin Interface, which is a web based browser GUI implemented as a MarkLogic Server web application. This interface allows authorized administrators to manage audit events, user accounts, access control and TOE sessions.

Authorized administrators can also perform security management functions programmatically using the XQuery functions included in XQuery library modules that are included with MarkLogic Server. The programmatic libraries that support

security management are the Admin API, the Security API, and the PKI API. The Admin API enables the scripting of administrative tasks that would otherwise need the Admin Interface to perform, including TOE security management tasks (for example, management of TOE sessions, configuration of auditing, and so on). For example, you can write a program using the Admin API to create and configure App Servers, including setting the type of authentication that the App Servers use. Most functions in this library perform administrative tasks and therefore require the user who runs an XQuery program executing these functions to be an authorized administrator. The Security API provide functions for managing objects stored in the security database (users, roles, amps, and privileges). For example, you can use the Security API to create and modify users (including passwords), roles, amps, and privileges. The PKI API provides functions that manage private keys and other cryptographic management functions used with SSL/TLS (HTTPS) in FIPS mode.

Security management functions include the ability to control the creation, management, and configuration of databases, forests, servers, and hosts. Documents are stored in forests. The name forests comes from the fact that XML documents are tree structures and a collection of trees is a forest. One or more forests are gathered together to form a database. Databases are logical units against which you can assign HTTP and XDBC servers and set various runtime configuration options. A host is a single instance of MarkLogic Server running on a single machine. Databases exist as a logical abstraction because in a distributed environment it can be useful to have the same logical database spread across different hosts, perhaps one host with two forests and another with three.

# 4 Assumptions, Threats, and Clarification of Scope

The statement of TOE security environment describes the security aspects of the environment in which it is intended that the TOE will be used and the manner in which it is expected to be employed. The statement of TOE security environment therefore identifies the assumptions made on the operational environment and the intended method for the product and defines the threats that the product is designed to counter.

The security environment is defined in terms of assumptions made by the TOE and threats to the TOE. There are no defined organizational policies.

## 4.1 Assumptions

Following are the assumptions identified in the Security Target:

- TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
- The OS in the environment shall be able to provide reliable time stamps for use by the TOE.
- The underlying OS is trusted to provide protection of the DBMS processes and stored data from other processes running on the underlying OS.

- It is assumed there are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
- Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
- Passwords are encrypted during the authentication process.
- The web browsers used to access the Admin Interface perform correctly such that when the browser is closed, the active Admin session is terminated. Client applications used to access the Admin API, Security API, and PKI API will perform correctly and when the application is closed, the active Admin session will be terminated.

### 4.2 Threats

Following are the threats levied against the TOE and its environment as identified in the Security Target. The threats that are identified are mitigated by the TOE and its environment. All of the threats identified in the ST are addressed.

- Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
- A user may gain unauthorized access to the TSF data and TSF executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to TSF data or TSF resources. A malicious user, process, or external IT entity may misrepresent itself as the TSF to obtain identification and authentication data.
- A user may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).

# 5 Clarification of Scope

All evaluations (and all products) have limitations, as well as potential misconceptions that need clarifying. This text covers some of the more important limitations and clarifications of this evaluation. Note that:

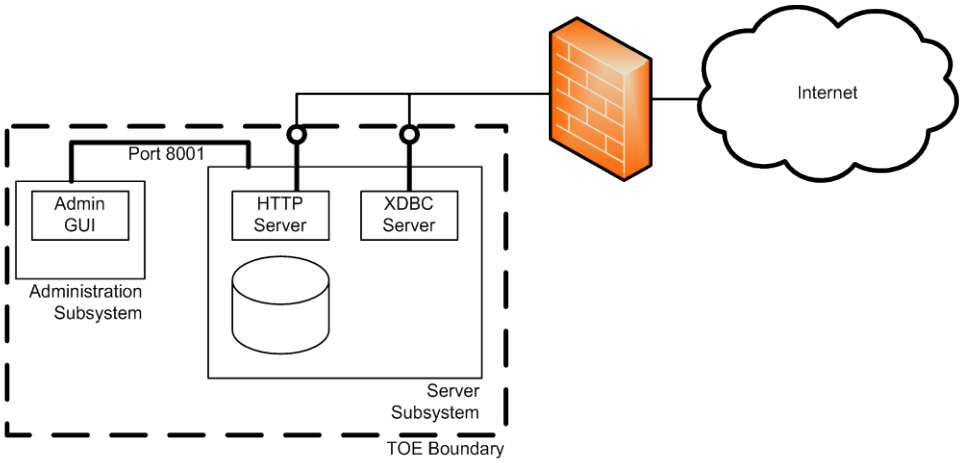
- As with any evaluation, this evaluation only shows that the evaluated configuration meets the security claims made, with a certain level of assurance (EAL 2 extended in this case).
- As with all EAL 2 evaluations, this evaluation did not specifically search for, nor seriously attempt to counter, vulnerabilities that were not “obvious” or vulnerabilities to objectives not claimed in the ST. The CEM defines an “obvious” vulnerability as one that is easily exploited with a minimum of understanding of the TOE, technical sophistication and resources.

- Cryptographic protection of passwords is used by the TOE; however, the password cryptography used in this product was not analyzed or tested to conform to cryptographic standards during this evaluation.

# 6 Architectural Information

This section provides a high level description of the TOE and its components as described in the Security Target.

The TOE consists of two subsystems, the Administration subsystem and the Server subsystem, as shown in the TOE Architecture diagram below.




TOE Architecture

## 6.1 Administration Subsystem

The Administration subsystem provides the Admin Interface to the Server subsystem. The Admin Interface application manages all features of the Server subsystem. It is composed of XQuery programs which are evaluated inside of an HTTP server. The HTTP server evaluates each request and sends a response back as a web page to the requester. The Admin Interface is accessed through HTTPS only (i.e., HTTP over SSL/TLS in FIPS mode).

## 6.2 Server Subsystem

The Server subsystem provides the software applications, network/application programming interfaces (APIs) and a database or contentbase.

The TOE supports three interfaces that are available through a network. An HTTP Server offers connectivity for the administrative interface and for customer applications with the Server subsystem. The communication pathways to and from the Server subsystem are depicted in Figure 2-1 by the lines labeled as “SSL/TLS in FIPS mode”. Two additional programmatic interfaces are provided by XDBC and ODBC protocols that can also use SSL/TLS in FIPS mode to protect the session. Developers write client applications to use

these interfaces in a system that requires access to a backend XML database. In particular, the HTTP and XDBC servers each provide the Admin API, Security API, and PKI API, which are collections of XQuery functions. The API functions are evaluated inside the HTTP and XDBC servers. Consequently, the servers enforce TOE security policy (for example, authentication, security management restrictions, access control, and auditing).

MarkLogic Server includes REST APIs, a Java Client API, and XCC libraries. These libraries are for application development. They do not provide any security functionality. The REST APIs are implemented as XQuery programs that run on a MarkLogic HTTP App Server. The Java Client API is implemented in Java, and calls the REST APIs, which in turn run on a MarkLogic HTTP App Server. The HTTP App Server is an interface to the TOE that honors DAC policy. The XCC libraries run against a MarkLogic XDBC App Server, which is an interface to the TOE that honors DAC policy.

The TOE can be set up as a single instance of MarkLogic Server on a single machine or it can support large scale high-performance architectures through multi-host distributed architectures. The following terminology has been defined for consideration in a TOE distributed environment:

- Cluster – A cluster is a set of one or more instances (see hosts, below) of MarkLogic Server (i.e., the TOE's Server subsystem) that will work together as a unified whole to provide content services. Security management functions of the TOE are performed from the Administration subsystem by connecting to any cluster host.
- Host – A host is a single instance of MarkLogic Server running on a single machine. Even though each host in a cluster can be configured to perform a different task, the full MarkLogic Server software (Server subsystem) runs on each host. MarkLogic Server Enterprise Edition enables multi-host configurations.
- Cluster Management Group – A cluster management group is a set of hosts with uniform HTTP, XDBC and ODBC configurations (but not necessarily uniform forest configurations). Cluster Management Groups are used to simplify cluster management.
- Forest – A forest is a repository for documents. Each forest is managed by a single host. The mapping of which forest is managed by which host is transparent to queries, as queries are processed against databases, not forests.
- Database – A database is a set of one or more forests that appears as a single contiguous set of content for query purposes. Each forest in a database must be configured consistently. HTTP and XDBC servers evaluate queries against a single database. In addition to databases created by the administrator for user content, MarkLogic Server maintains databases for administrative purposes: security databases, which contain user authentication and permissions information; schema databases, which are used to store schemas used by the system; modules databases, which are used to store executable XQuery code; last-login databases, which are used to store session history and data and triggers databases, used to store trigger definitions.

# 7 Security Policy

The TOE logically supports the following security functions:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access

## 7.1 Security audit

The TOE audit records that include date and time of the event, subject identity and outcome for security events. The TOE provides authorized administrators with the ability to include and exclude auditable events based on user identity, role, event type, object identity and success and failure of auditable security events. When appropriate, the TOE also associates audit events with the identity of the user that caused the event. The environment stores the audit records and also provides the system clock information that is used by the TOE to timestamp each audit record.

## 7.2 Cryptographic Support

Secure Sockets Layer protocol or Transport Layer Security (TLS 1.0) protocol (referred to in this document as SSL/TLS in FIPS mode) is used to provide protection of the communications surrounding the remote administrative sessions from disclosure and from modification (referred to as SSL/TLS in FIPS mode in this security target). For communication between a customer application on a network and the HTTP server, XDBC server, or ODBC server of the TOE, the TOE offers the use of a SSL/TLS session in FIPS mode to protect these communications. Finally, the TOE uses an SSL/TLS in FIPS mode protected channel to distribute TSF data when it is transmitted between distributed parts of the TOE (that is, hosts within a cluster).

The TOE uses OpenSSL object module version 2.0 which has undergone a FIPS 140-2 certification (certificate #1747). The TOE includes an OpenSSL object module built without modification from the source code of the OpenSSL FIPS certification. All references to “the TOE” performing cryptographic operations in this security target are indicating that the TOE is performing the operation through its use of the OpenSSL object module

## 7.3 User data protection

The TOE enforces a Discretionary Access Control (DAC) policy which restricts access to TOE-controlled object(s). Users of the TOE are identified and authenticated by the TOE before any access to the system is granted. Once access to the system is granted, authorization provides the mechanism to control what functions a user is allowed to perform based on the user’s role membership. Access to all TOE-controlled objects is denied unless access, based on role membership, is explicitly allowed. The authorized

administrator role shall be able to access any object regardless of the object's permissions. The TOE also provides amplifications or "amps" which temporarily grant roles to a user only for the execution of a specific function. Therefore, the DAC policy can also be extended by a user who is temporarily granted the privileged role in order to perform a specific "amped" function. The TOE also ensures that any previous information content of a resource is made unavailable upon the allocation of the resource to an object. Memory or disk space is only allocated when the size of the new data is first known, so that all previous data is overwritten by the new data.

### 7.4 Identification and authentication

The TOE requires users to provide unique identification and authentication data before any access to the system is granted and further restricts access to DBMS-controlled objects based on role membership. The TOE maintains the following security attributes belonging to individual users: role membership, and password. The TOE uses these attributes to determine access.

The TOE provides a password plug-in functionality that allows administrators to write custom code to require passwords to conform to specific rules (e.g., the number of characters, special characters, last change date).

### 7.5 Security management

The security functions of the TOE are managed by authorized administrators via the web-based Admin Interface, or application written using the Admin API, Security API, PKI API, and built-in admin functions. The ST defines the security role of 'authorized administrator'. Authorized administrators perform all security functions of the TOE including managing audit events, user accounts, access control and TOE sessions.

### 7.6 Protection of the TSF

The TOE provides protection mechanisms for its security functions. One of the protection mechanisms is that users must authenticate and have the appropriate permissions before any administrative operations or access to TOE data and resources can be performed on the system. The TOE also maintains a security domain that protects it from interference and tampering by untrusted subjects within the TOE scope of control.

Communication with remote administrators is protected by SSL/TLS in FIPS mode, protecting against the disclosure and undetected modification of data exchanged between the TOE and the administrator. Communication with remote customer applications can also utilize SSL/TLS in FIPS mode to protect against the disclosure and undetected modification of data exchanged between the TOE and the customer application. Customer applications must determine whether the use of SSL/TLS in FIPS mode is necessary for that specific customer application's data.

The TOE ensures that TSF data is encrypted and remains consistent when transmitted between parts of the TOE. The TOE provides consistency of TSF data between distributed parts of the TOE by regularly monitoring the configuration file and security database for changes and distributing the updated configuration file or security database to all parts of

the cluster. The TOE utilizes a TLS protected channel to distribute TSF data among a cluster.

### 7.7 TOE access

The TOE restricts the maximum number of concurrent sessions that belong to the same user by enforcing an administrator configurable number of sessions per user. The TOE also denies session establishment based on attributes that can be set explicitly by authorized administrators including role identity, time of day and day of week.

Upon successful session establishment, the TOE stores and retrieves the date and time of the last successful session establishment to the user. It also stores and retrieves the date and time of the last unsuccessful session establishment and the number of unsuccessful attempts since the last successful session establishment. This information is collected by the TOE Access security function, because the information pertains to user's attempts to access the TOE. The information gathered by the TOE pertains to historical session establishment actions by a user.

# 8 Documentation

Following is list of the evaluation evidence, each of which was issued by the developer (and sponsor).

### 8.1 Design documentation

| Document | Version | Date |
|---|---|---|
| MarkLogic Server Enterprise Edition 6.0 Security Architecture | ARC_0.7 | 13 June 2013 |
| MarkLogic Server Enterprise Edition 6.0 Functional Specification | FSP_04 | 13 June 2013 |
| MarkLogic Server Enterprise Edition 6.0 Technical Design Document | TDS_0.5 | 13 June 2013 |

### 8.2 Guidance documentation

| Document | Version | Date |
|---|---|---|
| MarkLogic Server Administrator's Guide | Release 6.0 | July 2013 |
| MarkLogic Server Understanding And Using Security | Release 6.0 | September 2012 |
| MarkLogic Server Installation Guide for All Platforms Release | Release 6.0 | September 2012 |

| | | |
|---|---|---|
| MarkLogic Server Developer's Guide | Release 6.0 | April 2013 |
| MarkLogic Common Criteria Evaluated Configuration Guide | Release 6.0 | July 2013 |

### 8.3 Lifecycle documentation

| Document | Version | Date |
|---|---|---|
| MarkLogic Server Enterprise Edition 6.0 Configuration Management | ALC-0.3 | June 2013 |
| Svn.out | N/A | December 2013 |
| MarkLogic Support Handbook | 13021116 | February 2013 |
| MarkLogic Enterprise Edition 6.0 Delivery Procedures | DP_0.1 | May 2013 |
| MarkLogic Enterprise Edition 6.0 Flaw Remediation Procedures | FLR_0.3 | June 2013 |

### 8.4 Test documentation

| Document | Version | Date |
|---|---|---|
| MarkLogic Server Enterprise Edition 6.0 Test Design | ATE-0.1 | August 2013 |
| MarkLogic Server Enterprise Edition 6.0 Functional Test Plan | ATE_FUN-0.5 | October 2013 |

Actual results in the form of list files, key files, XML files, text files, and JAVA files

### 8.5 Security Target

| Document | Version | Date |
|---|---|---|
| MarkLogic Server Enterprise Edition 6.0 Security Target | Version 1.0 | December 20 2013 |

# 9 IT Product Testing

This section describes the testing efforts of the developer and the Evaluation Team.

## 9.1 Developer Testing

The developer tested the interfaces identified in the functional specification and mapped each test to the security function, more specifically to the security functional requirements tested. The scope of the developer tests included all the TSFI. The testing covered the security functional requirements in the ST including: Security Audit, Identification and Authentication, Security Management, User Data Protection, Protection of the TSF, and TOE Access. All security functions were tested and the TOE behaved as expected. The evaluation team determined that the developer's actual test results matched the vendor's expected results.

## 9.2 Evaluation Team Independent Testing

The evaluation team exercised the entire automated test suite and a subset of the vendor's manual test suite. The following describes the testing environment of the TOE diagramed below:

- All the computing resources are behind a firewall so only authorized and authenticated machines have access to the HTTP, XDBC, ODBC, or the Admin interfaces of the TOE.
- MarkLogic Server is installed on either a Red Hat or Solaris server.
- The HTTP, XDBC, and ODBC interfaces, as well as the Admin API, Security API, PKI API, and Admin Built-in Functions, are part of the MarkLogic Server binary and are available when MarkLogic Server is started.
- The QA Harness runs on the same Red Hat or Solaris server as MarkLogic Server.
- During testing, the QA Harness creates XDBC, HTTP, and ODBC application servers on the MarkLogic Server during testing.
- The QA harness runs in FIPS mode, which runs all tests through TLS connections.
- The Admin Interface also runs on MarkLogic Server and is accessible via HTTPS through a supported web browser.

The evaluators generated keys using the putty Key generator. The public key was provided to the MarkLogic developer who installed the key on the server (TOE). The evaluators accessed the servers (TOE) remotely using putty secure connection (SSH Tunnel).

In addition to developer testing, the evaluation team conducted its own suite of tests, which were developed independently of the sponsor. These also completed successfully.

### 9.3 Vulnerability Testing

The evaluators developed vulnerability tests to address the Protection of the TSF and TOE access security functions, as well as performing a public search for vulnerabilities. These tests identified no vulnerabilities in the specific functions provided by the TOE.

# 10 Evaluated Configuration

The TOE consists of the software applications and network protocol interfaces (described and shown in the diagram above). The MarkLogic Server administration subsystem is supported on the following browsers for the evaluated configuration:

- Firefox 17 on Windows and Mac OS
- Internet Explorer 8 and 9 on Windows
- Chrome 23 on Windows and Mac OS

The Server subsystem applications and network interfaces execute either on Sun Solaris or Linux operating systems. The TOE requires the following hardware and operating system (OS) platforms in the IT environment:

**Memory, Disk Space, and Swap Space Requirements**
Before installing the software, the system must meet the following minimum requirements:

- 512 MB of system memory, minimum.
- Three times the disk space of the source content to be loaded.
- Swap space at least equal to the amount of physical memory on the machine.

**Supported Platforms – Server Subsystem**
The MarkLogic Server server subsystem is supported on the following platforms for the evaluated configuration:

- Sun Solaris 10 (x64)
- Red Hat Enterprise Linux 5.0 (x64)
- Red Hat Enterprise Linux 6 (x64)

The TOE relies on the hosting OS to protect its applications, processes, and any locally stored data. Web browsers in the IT environment are utilized to access the Admin Interface and the HTTP server. As noted previously, the TOE can be deployed on a single machine or in a distributed environment across multiple machines.

For specific configuration settings required in the evaluated configuration see MarkLogic Server Installation Guide for All Platforms, MarkLogic Server Administrator's Guide, MarkLogic Server Understanding and Using Security, and MarkLogic Common Criteria Evaluated Configuration Guide.

# 11 Results of the Evaluation

The evaluation was conducted based upon the Common Criteria (CC), Version 3.1, Revision 4, September 2012; the Common Evaluation Methodology (CEM), Version 3.1, Revision 4, September 2012. The evaluation confirmed that MarkLogic Server Enterprise Edition Version 6.0-4 product is compliant with the Common Criteria Version 3.1 Revision 4, functional requirements (Part 2), Part 2 extended, assurance requirements (Part 3) conformant for EAL2 augmented with ACL_FLR.3. The details of the evaluation are recorded in the CCTL's evaluation technical report; Final Evaluation Technical Report for the MarkLogic Server Enterprise Edition Version 6.0, Part 1 (Non-Proprietary) and Part 2 (Proprietary). The product was evaluated and tested against the claims presented in the MarkLogic Server Enterprise Edition Version 6.0 Security Target, Version 1.0, December 20, 2013.

The Validator followed the procedures outlined in the Common Criteria Evaluation Scheme publication number 3 for Technical Oversight and Validation Procedures. The Validator has observed that the evaluation and all of its activities were in accordance with the Common Criteria, the Common Evaluation Methodology, and the CCEVS. The Validator therefore concludes that the evaluation team's results are correct and complete.

The evaluation team's assessment of the evaluation evidence demonstrates that the claims in the ST are met. Additionally, the evaluation team's performance of a sample of the suite of the vendor test, the evaluation team's independent tests and the vulnerability test, also demonstrated the accuracy of the claims in the ST.

# 12 Validator Comments/Recommendations

All Validator concerns with respect to the evaluation have been addressed. No issues are outstanding.

# 13 Security Target

The Security Target is identified MarkLogic Server Enterprise Edition 6.0 Security Target, Version 1.0, December 20, 2013. The document identifies the security functional requirements (SFRs) that are levied on the TOE, which are necessary to implement the TOE security policies. Additionally, the Security Target specifies the security assurance requirements necessary for EAL 2 augmented with ALC_FLR.3.

# 14 Glossary

The following definitions are used throughout this document:

| | |
|---|---|
| API | Application Programming Interface |
| CC | Common Criteria |
| CEM | Common Evaluation Methodology |

| | |
|---|---|
| CCEVS | Common Criteria Evaluation and Validation Scheme |
| DAC | Discretionary Access Control |
| DBMS | Database Management System |
| DoD | Department of Defense |
| DoS | Denial of Service |
| EAL | Evaluation Assurance Level |
| GUI | Graphical User Interface |
| HLD | High-level Design |
| IA | Initial Assessment |
| NIAP | National Information Assurance Partnership |
| NIST | National Institute of Standards and Technology |
| NSA | National Security Agency |
| OS | Operating System |
| SFP | Security Function Policy |
| SOF | Strength of Function |
| SQL | Structured Query Language |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSC | TOE Scope of Control |
| TSF | TOE Security Functions |
| URI | Uniform Resource Identifier |
| US | United States |
| W3C | World Wide Web Consortium |
| XML | Extensible Markup Language |

# 15 Glossary of Terms

The terminology below is described in order to clarify and distinguish the terms used in the ST and those used in the TOE product documentation.

**Amps** Amps are security objects that temporarily grant role membership to unprivileged users only for the

| | |
|---|---|
| | execution of a given function. While executing an "amped" function, the user is temporarily part of the amped role which in turn temporarily grants the user the additional privileges and permissions given by the roles configured in the amp. Amps enable the effect of the additional permissions and privileges to be limited to a particular function. |
| **Permissions** | Permissions provide a role with the ability to perform capabilities (that is, read, insert, update, execute) on documents. A permission is a combination of role and capability. Permissions are assigned to documents. Users gain the authority to perform a capability on a document if they are members of the role the permission associates with the capability. |
| **Capabilities** | Capabilities are operations on documents: Read, Update, Insert or Execute. |
| **Execute Privileges** | Execute privileges allow developers to control authorization for the execution of an XQuery function. These privileges are assigned to a user through a role. |
| **Role** | MarkLogic Server implements a role-based security model. A role contains privileges and the privileges allow access to execute code on the system (for example, security management functions). A role also allows access to a documents based on permissions defined on the document. |
| **URI Privileges** | Uniform Resource Identifier privileges are used to control the creation of documents with a given URI prefix. In order to create a document with a prefix that has a URI privilege associated with it, a user must be part of a role to which the needed URI privilege is assigned. |
| **Application Server Privileges** | Application Server Privileges are Execute Privileges that can be configured to control access to each application server (that is, HTTP or XDBC server). If such a privilege is specified, any users that access the server must possess the specified privilege. |

# 16 Bibliography

The Validation Team used the following documents to produce this Validation Report:

[1] Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 4, September 2012

[2] Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1, Revision 4, September 2012

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 4, September 2012

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 4, September 2012.

[5] MarkLogic Server Enterprise Edition 6.0, Final Proprietary ETR – Part 2, Version 0.1 dated 16 December 2013 and Supplemental Team Test Report, Version 1.0, 10 December 2013.

[6] MarkLogic Server Enterprise Edition 6.0 Security Target, Version 1.0, December 20, 2013.

[7] Common Criteria Evaluation and Validation Scheme, Publication #4, Guidance to CCEVS Approved Common Criteria Testing Laboratories, Version 2.0, 8 September 2008.